

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA MECÁNICA

"ANÁLISIS DE LA DIRECTIVIDAD Y EL  
COMPORTAMIENTO ACÚSTICO DE DIFERENTES  
SONIDOS DE ADVERTENCIA PARA VEHÍCULOS  
ELÉCTRICOS"



**UNIVERSITAS**  
*Miguel Hernández*

TRABAJO FIN DE GRADO

Julio –2020

AUTOR: David Sánchez Tello

DIRECTOR: Ramón Peral Orts

COTUTOR: Miguel Fabra Rodríguez

## **AGRADECIMIENTOS.**

En primer lugar, me gustaría darle las gracias a la Universidad Miguel Hernández de Elche, por haberme visto crecer profesionalmente. Me gustaría agradecer también al Departamento de Ingeniería Mecánica y Energía por brindarme la oportunidad de realizar un TFG de investigación como este, sobre un tema tan puntero como los SAAV.

También tengo que reconocer la enorme labor que Ramón Peral Orts ha realizado al ayudarme a sacar este trabajo adelante. Estoy muy contento con haber dado el máximo en la consecución de este proyecto y haber tenido mi primera beca de investigación.

Debo destacar también la inestimable ayuda que me dio Miguel Fabra Rodríguez en todas aquellas dudas que fueron surgiendo.

Por último, me gustaría agradecer a mis padres y compañeros de universidad por haberme apoyado en mis momentos más duros.



## ÍNDICE

1.	INTRODUCCIÓN .....	6
1.1.	CONTEXTO GENÉRICO .....	6
1.2.	PERCEPCIÓN DEL SONIDO.....	7
1.2.1.	MOLESTIA.....	11
1.3.	DESARROLLO TEÓRICO .....	12
1.3.1.	TIPOS DE PONDERACIONES .....	13
1.3.1.1.	PONDERACIÓN FRECUENCIAL.....	13
1.3.1.2.	PONDERACIÓN TEMPORAL.....	14
1.3.2.	ARRAYS DE MICRÓFONOS Y CÁMARAS ACÚSTICAS .....	16
1.3.3.	TEORÍA ACÚSTICA .....	19
1.3.3.1.	FUNDAMENTOS DEL SONIDO.....	19
1.3.3.2.	DIRECTIVIDAD .....	23
2.	ANTECEDENTES.....	26
2.1.	PROBLEMÁTICA DEL VEHÍCULO ELÉCTRICO.....	26
2.2.	TIPOS DE SAAV.....	27
2.3.	ESTADO DEL ARTE.....	31
2.4.	NORMATIVA .....	34
2.4.1.	CARACTERÍSTICAS ACÚSTICAS DE LOS SONIDOS .....	34
2.4.2.	ENSAYO DE VELOCIDAD CONSTANTE .....	35
2.4.3.	ENSAYO DE MARCHA ATRÁS.....	35
2.4.4.	CAMBIOS DE FRECUENCIA PARA LA ACELERACIÓN Y DESACELERACIÓN.....	35
3.	OBJETIVOS .....	37
4.	METODOLOGÍA .....	39
4.1.	SONIDOS DE ADVERTENCIA EMBARCADOS .....	39
4.2.	DESARROLLO DEL ENSAYO .....	40
4.2.1.	EMPLAZAMIENTO DEL ENSAYO.....	41
4.2.2.	CONDICIONES METEOROLÓGICAS .....	42
4.2.3.	POSICIÓN DE LOS MICRÓFONOS .....	42
4.2.4.	CONDICIONES DE FUNCIONAMIENTO .....	42
4.2.5.	OBTENCIÓN DE DATOS .....	44
4.3.	DISPOSITIVOS EXPERIMENTALES.....	44
4.3.1.	DISPOSITIVO DE ADQUISICIÓN DE DATOS SONOROS.....	44
4.3.2.	CALIBRACIÓN.....	46
4.3.3.	INSTRUMENTACIÓN METEOROLÓGICA .....	46
4.3.4.	CONDICIONES DEL VEHÍCULO.....	47

4.4.	PROCESAMIENTO DE DATOS.....	48
4.4.1.	NIVEL SONORO CONTINUO EQUIVALENTE MÁXIMO.....	49
4.4.1.1.	PROGRAMA EXTRACTOR DE DATOS.....	49
4.4.1.2.	PROGRAMA OPERACIONES.....	55
4.4.1.3.	PROGRAMA ANÓMALOS.....	60
4.4.1.4.	PROGRAMA ESPFREC.....	64
4.4.1.5.	VERIFICACIÓN.....	66
4.4.1.6.	ESQUEMA DE ARCHIVOS Y PROGRAMAS.....	68
4.4.1.7.	PROGRAMA SELECTORPASADAS.....	69
4.4.1.8.	CONJUNTO DE PROGRAMAS OBTPRESSON + SONIDO DE ADVERTENCIA.....	75
4.4.1.9.	PROGRAMA REPS.....	78
4.4.1.10.	PROGRAMA INFO.....	80
4.4.1.11.	PROGRAMA SELECTOREXC.....	82
4.4.1.12.	CONJUNTO DE PROGRAMAS VELEXC + SONIDO DE ADVERTENCIA.....	86
4.4.2.	DIRECTIVIDAD VERTICAL.....	90
4.4.2.1.	OPERACIONES DE PARTIDA.....	91
4.4.2.2.	PROGRAMA PLOTTRIDIMENSIONAL.....	91
4.4.2.3.	PROGRAMAS DIRECTIVIDADIZQ Y DIRECTIVIDADDDCHA.....	98
4.4.2.4.	ESQUEMA DE ARCHIVOS Y PROGRAMAS.....	115
4.4.2.5.	PROGRAMA SELECTORPASADAS.....	116
4.4.2.6.	CONJUNTO DE PROGRAMAS DISTRIB + SONIDO DE ADVERTENCIA.....	120
4.5.	RESULTADOS.....	125
4.5.1.	NUMÉRICO.....	125
4.5.1.1.	NIVEL SONORO CONTINUO EQUIVALENTE MÁXIMO.....	125
4.5.1.1.1.	GRÁFICA GLOBAL.....	125
4.5.1.1.2.	ESPECTRO DE FRECUENCIA.....	126
4.5.1.2.	DIRECTIVIDAD VERTICAL.....	147
4.5.2.	DISCUSIÓN.....	168
4.5.2.1.	NIVEL SONORO CONTINUO EQUIVALENTE MÁXIMO.....	168
4.5.2.1.1.	GRÁFICA GLOBAL.....	168
4.5.2.1.2.	ESPECTRO DE FRECUENCIAS.....	173
4.5.2.2.	DIRECTIVIDAD VERTICAL.....	178
5.	CONCLUSIONES.....	187
5.1.	TRABAJOS FUTUROS.....	189
6.	REFERENCIAS.....	190
6.1.	BIBLIOGRAFÍA.....	190
6.2.	WEBGRAFÍA.....	193

---

ANEXO I. PROGRAMAS PARA ANÁLISIS DE NIVEL SONORO EQUIVALENTE MÁXIMO .....	197
ANEXO II. PROGRAMAS PARA ANÁLISIS DIRECTIVIDAD VERTICAL .....	216



# 1. INTRODUCCIÓN

## 1.1. CONTEXTO GENÉRICO

De acuerdo con la AEMA, el transporte es responsable de más del 30% de las emisiones de CO<sub>2</sub> en la UE, de los cuáles un 72% proviene directamente del transporte por carretera [I]. Las emisiones debidas al transporte han incrementado un 25% respecto de los valores en 1990, mientras que en el resto de sectores ha disminuido. Los coches son el medio de transporte más contaminantes, con un 60,7% del total de las emisiones del transporte por carretera.

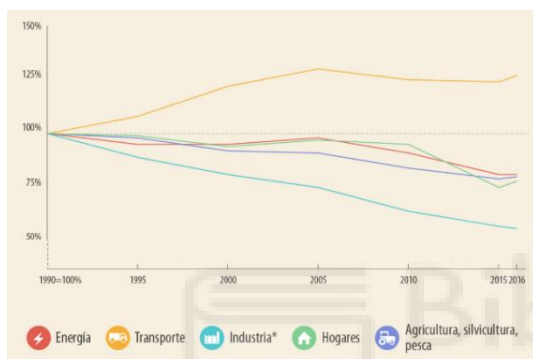


Figura 1.01. Evolución de la proporción de cada grupo respecto del total de contaminación [I].

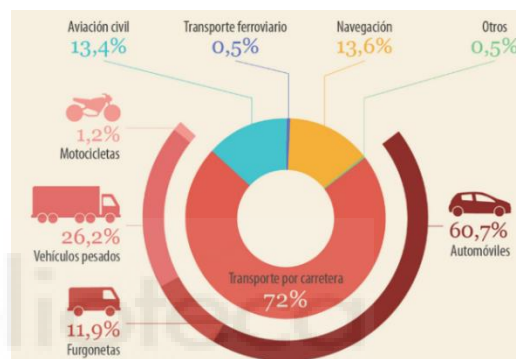


Figura 1.02. Reparto de la contaminación por vehículo [I].

En la actualidad, la preocupación por el medio ambiente y la alineación de la industria con los Objetivos de Desarrollo para el Milenio ha provocado el aumento de la inversión en tecnologías alternativas al automóvil tradicional con motor de combustión.

Una de esas alternativas con mayor futuro es el coche eléctrico. Los coches con tecnología 100% eléctrica no emiten gases contaminantes al medio ambiente. Además, por lo general, sufren menos averías mecánicas y se produce un gran ahorro en combustible.

Además, numerosas autoridades están fomentando la adquisición de estos vehículos mediante ventajas fiscales. En Cataluña, los conductores de vehículos eléctricos pueden optar a un descuento del 75% en todos los peajes de dicha comunidad [II]. Por otro lado, con alternativas como Madrid Central en Madrid y ZBE en Barcelona, se propone incentivar a los ciudadanos al empleo del transporte eléctrico, permitiéndoles acceder a zonas de bajas emisiones [III].

No obstante, los coches eléctricos presentan también una serie de desventajas, destacando en gran medida la baja sonoridad de estos vehículos. Por ello, este TFG surge para obtener una

serie de resultados que permita apoyar la búsqueda de una alternativa viable para solucionar este problema.

A su vez, este TFG también tiene otra motivación- Esta es la conclusión de una etapa dentro de mi carrera profesional. Sin embargo, otra razón por la que decidí realizar esta tarea fue aprender acerca de un tema completamente nuevo, del cual no recibí ningún tipo de conocimiento a lo largo de la carrera. Quería demostrarme a mí mismo que era capaz de afrontar nuevos retos de una forma más o menos autónoma.

## 1.2. PERCEPCIÓN DEL SONIDO

En este apartado se recoge la información acerca del modo en el que el oído humano se comporta, la teoría de la audición aceptada hasta el momento y la subjetividad en la percepción [1, 2, 3, 4, 5]. Más tarde, se tratará la molestia sonora [6, 7, 8].

El oído humano es el órgano sensorial que tiene como función percibir los sonidos que se generan en el entorno y enviar los estímulos que se generan en él al cerebro, para que puedan ser procesados.

El sistema humano de audición presenta gran sofisticación, siendo capaz de detectar frecuencias de entre 20 Hz y 20000 Hz. Además, es capaz de realizar distinciones entre varios timbres de sonido.

La audición del sonido, básicamente se basa en la captación de las ondas de presión acústica por medio de la oreja, siendo posteriormente amplificadas por el conducto auditivo. La oreja tiene un doble propósito: el primero es recoger la mayor cantidad de energía acústica posible para, a continuación, concentrarla en el conducto auditivo y la segunda función es proporcionar direccionalidad para poder localizar el foco sonoro.

El conducto auditivo, en el oído externo, trabaja como un resonador de banda ancha. Este resuena con los múltiplos impares de la frecuencia fundamental del sonido. La ganancia que se consigue por medio de esa resonancia, llega a 10 dB, para la banda de 2000 a 6000 Hz.

Las ondas amplificadas en el conducto auditivo, producen la vibración del tímpano. Este órgano se presenta como una membrana circular, de forma cónica. El tímpano trabaja como un resonador amortiguado cuyo rango de resonancia se encuentra entre 20 y 20000 Hz. Esta membrana se encarga de recoger la onda sonora y transformarla en vibración mecánica. Cuando al tímpano llegan bajas frecuencias, la membrana vibra de forma similar a como lo haría un cuerpo

rígido. No obstante, cuando el tímpano recibe frecuencias superiores a 2400 Hz, este vibra, para cada frecuencia específica, según una forma singular.

Las oscilaciones generadas en el tímpano se transmiten del oído medio hasta el interno a través de la cadena de huesecillos. Los huesecillos quedan conectados por medio de unos músculos que son capaces de disminuir la amplitud de las oscilaciones que entran, cambiando su tensión. El objetivo de lo anteriormente explicado es para que el oído pueda protegerse de sonidos que sean excesivamente intensos. Sin embargo, este mecanismo se demora medio segundo en actuar, así que, existe cierta indefensión ante sonidos intensos muy abruptos o repentinos, como explosiones.

Los huesecillos se encargan de enviar la vibración a la cóclea. Este órgano es una cavidad hermética que se encuentra llena de un líquido, la linfa. La cóclea tiene forma de espiral y se divide a lo largo de su longitud en dos partes, realizando esa división la membrana basilar. Aquí, las vibraciones son transformadas en impulsos nerviosos que serán interpretados por el cerebro.

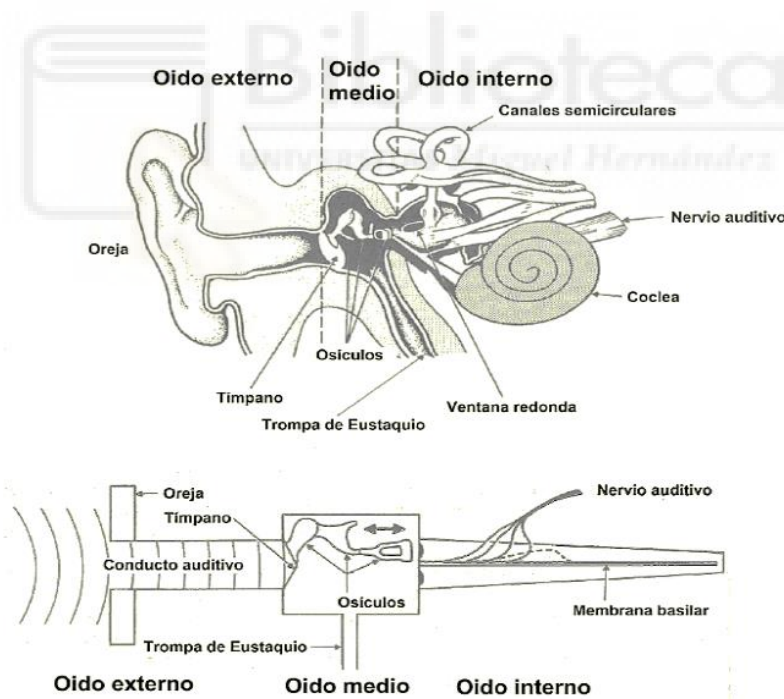


Figura 1.03. Estructura del oído [1].

La audición se puede definir como el fenómeno por el que la cóclea convierte el sonido en estímulos eléctricos que es el único tipo de señal que el cerebro sabe interpretar. La audición puede explicarse a través de la teoría que se conoce como Teoría del Lugar.



En la Teoría del Lugar (o Localización), se asegura que la cóclea, por medio de la membrana basilar, dispersa espacialmente las componentes de frecuencia del sonido. En esta teoría se afirma, a su vez, que solamente algunas células sensoriales se activan para cada componente de frecuencia que presente el sonido. Además, la teoría afirma que cuando un sonido llega al tímpano, este provoca vibraciones que origina un movimiento de la linfa en la cóclea. De este modo, se produce una onda que se desplaza por la membrana basilar. Esta teoría es la más aceptada hasta el momento.

La membrana basilar es más estrecha y rígida en el extremo basal (o base), y más ancha y flexible en el extremo apical (o ápex). Estas zonas están representadas en la Figura 1.04, donde se visualiza la cóclea desarrollada.

Mientras la onda viaja hacia el ápex, su amplitud aumenta, hasta llegar a cierto punto, donde la vibración decrece describiendo un máximo. Los sonidos de baja frecuencia poseen un máximo cerca del ápex, mientras que los de alta frecuencia lo poseen cerca de la base. Con esta afirmación, se puede determinar que cada punto de la membrana responde a una frecuencia diferente.

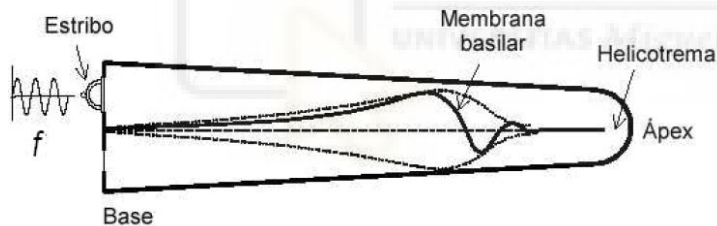


Figura 1.04. Cóclea desarrollada e identificada con todas sus partes [1].

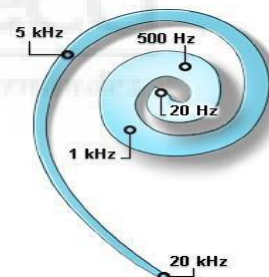


Figura 1.05. Sensibilidad de la cóclea para distintas frecuencias [2].

Los terminales neuronales están acoplados a la membrana basilar, por lo que, dependiendo de la frecuencia que se reciba, estas se excitarán de forma localizada. Las terminaciones, al excitarse generan impulsos eléctricos que son transmitidos al cerebro para su interpretación.

Para la estimulación de estas terminaciones, se necesita un determinado nivel o umbral. Sin embargo, los niveles recibidos no pueden ser excesivos, puesto que estas terminaciones pueden dañarse.

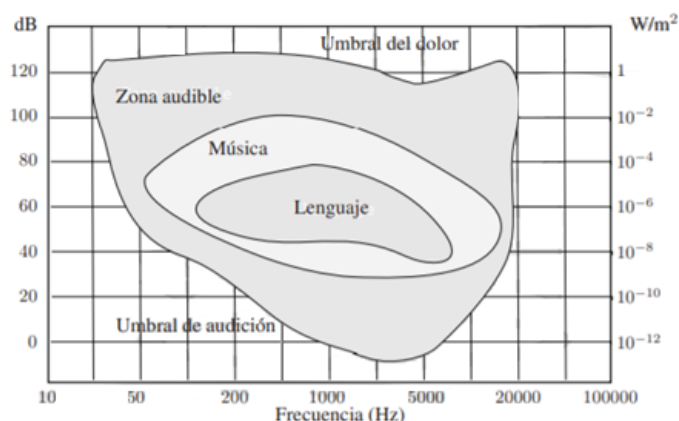


Figura 1.06. Rango audible de una persona sana [3].

No obstante, la forma en el que los seres humanos perciben los sonidos, tiene un alto grado de subjetividad. La intensidad percibida por una persona no está directamente relacionada con la intensidad física, sino que afectan otros factores como puede ser la frecuencia del sonido o la duración del mismo. La medida que indica la subjetividad en la percepción de los sonidos se conoce como sonoridad.

Para estudiar esta subjetividad, se decidió representar la sonoridad. La representación de la sonoridad se hace por medio de los que se conoce como escala de fonios. A 1000 Hz, el valor de sensación de intensidad sonora percibida en fonios y el nivel de presión sonora en dB, son iguales. Si se realiza una comparación del sonido a 1000 Hz con sonidos a otras frecuencias, se obtienen unos contornos con igual sensación de intensidad. Esto se conoce como líneas isofónicas.

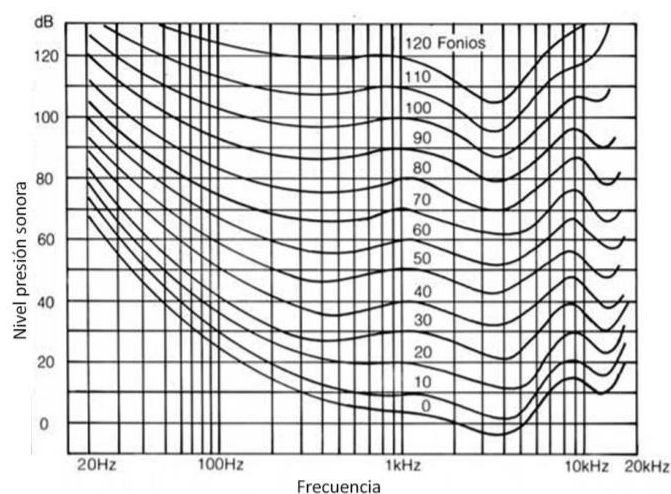


Figura 1.07. Líneas isofónicas [4].

Las líneas isofónicas, tienen forma de curva y unen niveles de presión sonora que tienen, a distinta frecuencia, la misma sonoridad. Esto implica que, todos los puntos que se encuentren en una línea isofónica, se perciben con la misma intensidad subjetiva, aunque físicamente el nivel de intensidad no lo sean. Por ello es posible que existan sensaciones sonoras parecidas, aunque los valores de frecuencia y/o intensidad sean diferentes.

### 1.2.1. MOLESTIA

El ruido, se define como todo aquel sonido no deseado y que, por lo tanto, genera molestia. El análisis del foco donde se originó y su propagación, permite establecer las características de este sonido no deseado. A pesar de que la cuantificación del ruido es posible, existe otra variable que juega un papel fundamental cuando se busca evaluar si un sonido es molesto o no. Este factor es la subjetividad del ser humano a la hora de percibir un sonido. Esta subjetividad, incrementa considerablemente la complejidad de establecer, de forma general, que parámetros del sonido generan molestia, y más aún cuantificarlos según el grado de molestia que generen.

No obstante, con el paso de los años, se ha conseguido establecer una serie de patrones en las características del sonido que indican que este va a producir molestia.

- La energía sonora. El nivel de presión sonora influye sustancialmente en que un sonido sea percibido como molesto. A mayor nivel de presión sonora, más energía presenta el mismo y, por lo tanto, se producirá más molestia.

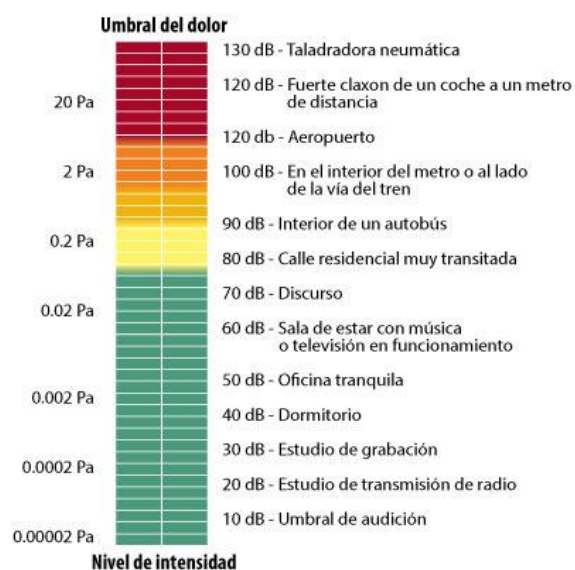


Figura 1.08. Niveles de presión sonora en función de la molestia percibida [IV]

- El tiempo de exposición. Generalmente, estar expuesto un mayor tiempo al ruido, supone percibirlo como molesto.
- Las características del receptor. Como ya se ha comentado, existe una gran subjetividad en la forma en la que los seres humanos perciben un sonido. Variables como la cultura, las facultades físicas del receptor, su sensibilidad auditiva o la actividad profesional de este, determinan si el oyente sentirá el sonido como molesto o no.
- Las características del sonido. Aun cuando un sonido se emita con un nivel de presión sonora determinado durante un tiempo específico, otras características como la frecuencia o el ritmo del mismo determinan lo molesto que es un sonido.

Con el objetivo de dar soluciones al problema de la molestia acústica, se estableció el Real Decreto 1367/2007 [6]. Aquí, se especifican una serie de componentes que pueden presentar los sonidos y que generan molestia. Estos componentes son:

1. Componentes tonales. Se produce cuando una determinada frecuencia destaca enormemente por encima del resto. Sin embargo, también se puede producir que existan dos o más tonos, presentándose en distintas frecuencias.
2. Componentes de baja frecuencia. Son todos aquellos ruidos donde la energía se presenta acumulada en el rango bajo de frecuencias (20-125 Hz).
3. Componentes impulsivas. Aquellos sonidos que presentan un cambio transitorio elevado y brusco en el nivel energético, produciéndose en un pequeño intervalo de tiempo (menor a 1 segundo).

El Real Decreto establece una serie de correcciones para adecuar el nivel de presión sonora, en función de las componentes que presente el ruido analizado. De esta forma, se puede obtener un procedimiento para cuantificar la molestia de un sonido.

### **1.3. DESARROLLO TEÓRICO**

A lo largo de este apartado, se mostrarán y explicarán todos los términos relacionados con el mundo de la Acústica que han sido empleados en este TFG.

La información que se presenta en las siguientes partes está relacionada con las ponderaciones de datos existentes, la teoría relacionada con los arrays de micrófonos y cámaras acústicas, y con los fundamentos del sonido.

### 1.3.1. TIPOS DE PONDERACIONES

Este apartado se encarga de enumerar y detallar las distintas ponderaciones que se utilizan al trabajar con sonidos. Estos apartados se basarán en los recursos [5, 7, 8].

#### 1.3.1.1. PONDERACIÓN FRECUENCIAL

A pesar de que el rango de frecuencias audibles es de 20 a 20000 Hz, el oído humano no responde de igual forma para distintas frecuencias. Es decir, para niveles bajos de presión sonora, únicamente son perceptibles aquellos sonidos emitidos con frecuencias medias. Para niveles altos de presión, la frecuencia no influye en la percepción del sonido, escuchándose todas las frecuencias de igual manera.

Partiendo de las curvas isofónicas, se llevó a cabo la definición de una serie de filtros que, al ponderar los datos obtenidos a través de dispositivos de recolección de datos acústicos, proporcionasen respuestas similares a cómo estos serían percibidos por el ser humano.

Estos filtros, actúan en los niveles de presión de cada una de las bandas de frecuencia que son corregidos a través de unos índices. Los filtros producen en gran medida una atenuación de las componentes bajas de frecuencia. Para la obtención de los índices a emplear, se debe recurrir a las curvas de ponderación que quedan representadas en la Figura 1.08. Las ponderaciones más importantes son las ponderaciones A, B y C.

- La curva de ponderación A se aplica a sonidos con niveles de presión sonora bajos.
- La curva de ponderación B se usa para filtrar sonidos con niveles de presión sonora medios.
- La curva de ponderación C se emplea con sonidos que presentan niveles de presión sonora altos.

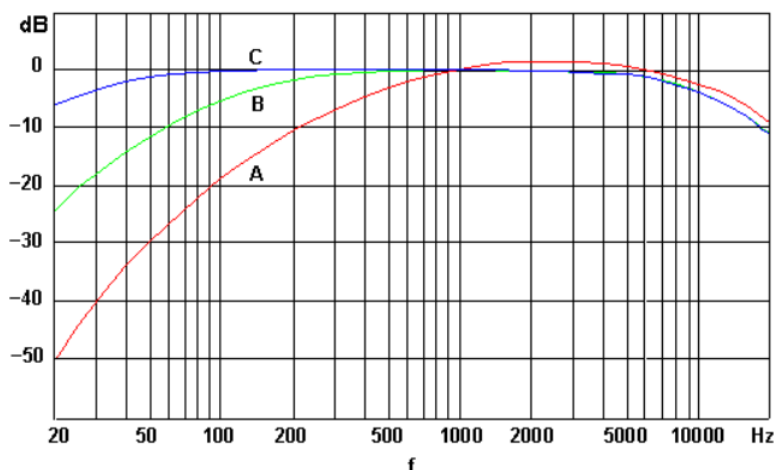


Figura 1.09. Curvas de ponderación frecuencial [V].

Para el sector del transporte y automovilístico se usa la ponderación tipo A. Por ello, en este trabajo se usará este tipo de ponderación.

### 1.3.1.2. PONDERACIÓN TEMPORAL

Este tipo de ponderación tiene su origen en los primeros sonómetros que se desarrollaron. La primera generación de estos dispositivos consistía en un marcador con forma de esfera al que se le incluía un puntero. El puntero se movía sobre una escala calibrada e indicaba el nivel sonoro en dB.

En función de las características del sonido que se deseaba medir, existían ocasiones donde la tarea de adquisición de datos se hacía difícil. Esto se debía a que el puntero fluctuaba demasiado rápido y determinar la posición media del mismo era una tarea complicada. Por ello, las medidas que se consiguieron a través de esta forma de medición se les denominó mediciones tipo Fast.

Con el objetivo de reducir la variabilidad en las mediciones, lo que se decidió fue añadir a los sonómetros una amortiguación eléctrica. De este modo, disminuyó la variabilidad en los resultados, debido a que la fluctuación se redujo considerablemente. Las medidas recogidas por este método, se las llamó mediciones tipo Slow.

Para ponderar temporalmente en Fast y Slow, lo que debe realizarse es multiplicar un término exponencial equivalente a la señal de presión sonora elevada al cuadrado con ponderación A. Dicho término exponencial contiene una constante nominal que depende de la ponderación a usar. En la ponderación Fast, la constante nominal de tiempo es de 0.125 segundos. Por otro lado, para una ponderación temporal Slow, la constante es de 1 segundo.

Cuando se opta por realizar una ponderación temporal de tipo Fast, el nivel sonoro que se da en un determinado instante está mayormente influenciado por los niveles sonoros que se producen en momentos cercanos a este y menos por los sonidos del pasado distante. En el lado opuesto se encuentra la ponderación Slow.

Por último, destacar que, para el mismo sonido no estable, el nivel sonoro rápido (Fast) máximo suele ser mayor que el correspondiente nivel sonoro lento (Slow) máximo.

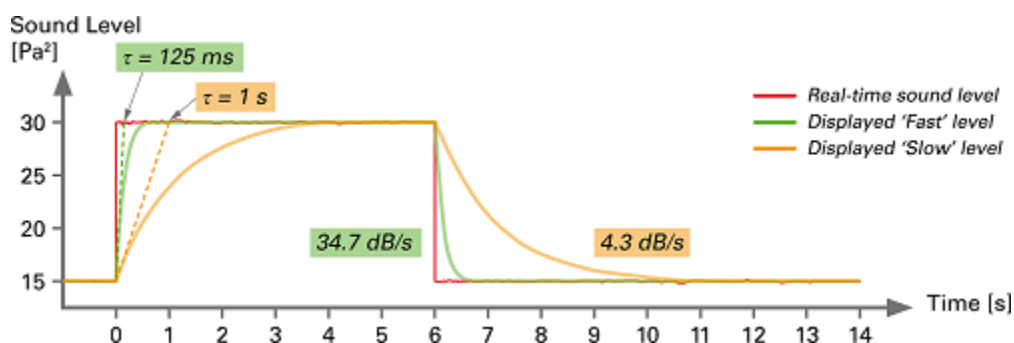


Figura 1.10. Ponderaciones temporales Fast y Slow [VI].

No obstante, las ponderaciones tipo Fast y Slow no son las únicas que existen, ya que también se puede usar una ponderación temporal de tipo impulso. Este tipo de constante presenta una característica diferenciadora y es que la constante nominal es de 0.035 segundos para zonas de las señales que aumentan con el tiempo, mientras que, si la señal decrece, la constante es de 1.5 segundos.

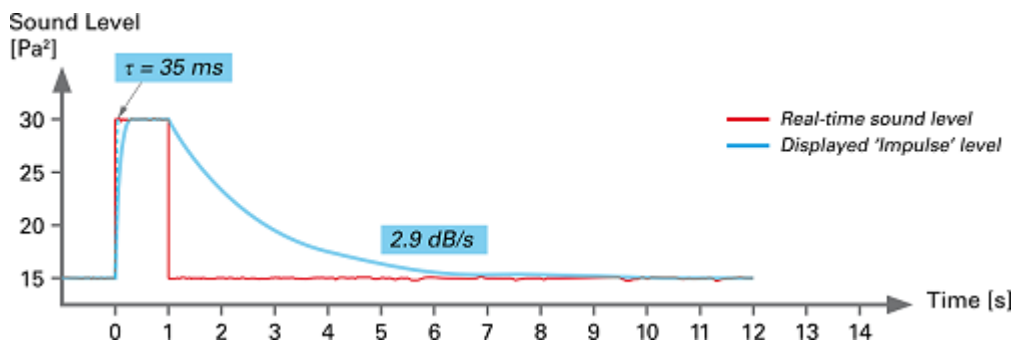


Figura 1.11. Ponderación temporal Impulse [VI].

Cada ponderación temporal tiene una función específica y se emplea en distintos casos:

- Fast. El empleo de esta ponderación se da cuando las medidas de sonido son fluctuantes.

- Slow. El uso de esta ponderación se da cuando el sonido a analizar no fluctúa rápidamente.
- Impulse. Esta ponderación se utiliza cuando se presenta un sonido de carácter impulsivo.

La aplicación de las ponderaciones temporales, se realiza cuando se desea calcular el promedio temporal de la señal de presión sonora elevada al cuadrado, generando así una señal más fácilmente interpretable para sacar conclusiones (se obtiene un valor RMS).

### 1.3.2. ARRAYS DE MICRÓFONOS Y CÁMARAS ACÚSTICAS

Para comprobar que un sistema SAAV cumple con las disposiciones y ensayos normativos mencionados en el apartado anterior, se debe hacer uso de unos equipos especializados en la grabación y tratamiento del sonido.

La unidad básica de estos equipos es el micrófono, que es el dispositivo capaz de efectuar grabaciones de sonido [VII]. Quizá no sea tan conocido que los micrófonos son sensibles de varias maneras a las ondas que llegan desde ángulos diferentes. En la Figura 1.12 se muestran los tres tipos de micrófonos principales, aunque hay más.

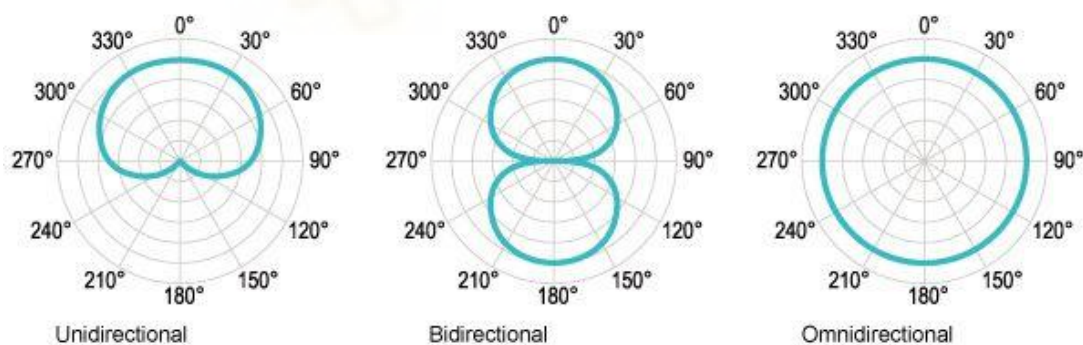


Figura 1.12. Direccionalidad según tipo de micrófonos [VII].

Los micrófonos unidireccionales son aquellos sensibles al sonido en una sola dirección. Por otro lado, los micrófonos bidireccionales cuentan con direccionalidad tanto trasera como delantera. En último lugar, los omnidireccionales, de forma teórica, tendrían la misma sensibilidad en todas las direcciones. Sin embargo, esto no ocurre en la realidad: siempre hay alguna zona de la circunferencia del diagrama que sufre un aplanamiento, que aumenta con el tamaño del micrófono.



Para las mediciones deseadas, se necesita conocer con gran precisión los sonidos emitidos desde el vehículo eléctrico. Se combinarán varios micrófonos, dando lugar al array de micrófonos que se encargarán de captar la información acústica que se da en el ambiente [9]. Existen diversos distintos tipos de array de micrófonos, de los cuales se destacan los siguientes:

- Array lineal. Los micrófonos se sitúan se disponen en una recta. Los elementos del array pueden encontrarse con espacios iguales entre elementos, denominándose así array lineal equiespaciado. En cambio, si los elementos del array se definen con la misma fase y amplitud, el array se conoce como lineal uniforme.
- Array plano. En este tipo, los micrófonos se distribuyen sobre un plano. Dentro de esta clase, están los circulares y reticulares que son los más conocidos. Este tipo presenta una ventaja en contraposición con los lineales, puesto que permiten generar diagramas de radiación en donde se puede estudiar cualquier dirección del espacio.
- Array tridimensional. Esta clase presenta los elementos situados alrededor de un volumen. Los arrays esféricos y semiesféricos son los más representativos en esta clase.

La existencia de distintas clases de arrays se debe a que, para cada tarea, es necesaria la aplicación de un tipo específico y no otro [9, 10]. Para adoptar el array adecuado para obtener los resultados de la forma deseada, parámetros como su tamaño, la disposición del array y el número de elementos que presentará el dispositivo son básicas para determinar por ejemplo el rango dinámico o la resolución de la información. Otros parámetros como la apertura máxima del array, la distancia entre los elementos que conformen el array o la geometría de este, definen la máxima frecuencia de muestreo utilizable, así como la máxima distancia que puede establecerse entre array y fuente. De este modo, si se decide emplear un array de gran tamaño, disponiéndolo a una distancia pequeña de la fuente sonora, se conseguirá una resolución espacial mayor. Si lo que se decide es emplear un array cuyos elementos se encuentran a distancias pequeñas unos de otros, el resultado será un dispositivo con un límite de alta frecuencia más grande.

Los arrays de micrófonos, entre otras funciones, pueden emplearse para la localización de fuentes emisoras de sonido, y su caracterización. Si se combina los arrays de micrófonos con pantallas y técnicas de visualización, se genera un dispositivo conocido como cámara acústica.

Básicamente, las cámaras acústicas pueden dividirse en dos clases distintas: las 2D y las 3D.

Las cámaras 2D emplean un array de micrófonos lineales o planos con elementos unidireccionales. Estos dispositivos son ideales para realizar mediciones en superficies planas.

Sin embargo, la mayoría de objetos no son planos, y el método de mapeado bidimensional introduce errores en los cálculos de sonido. En muchas aplicaciones este error es despreciable, pero en espacios cerrados sí que es significativo.

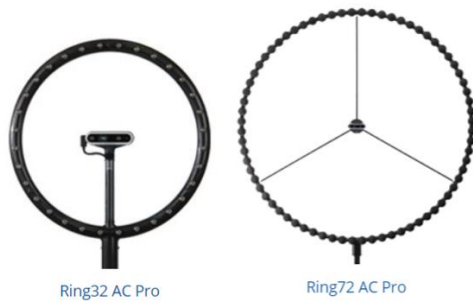


Figura 1.13. Cámaras acústicas 2D [VIII].

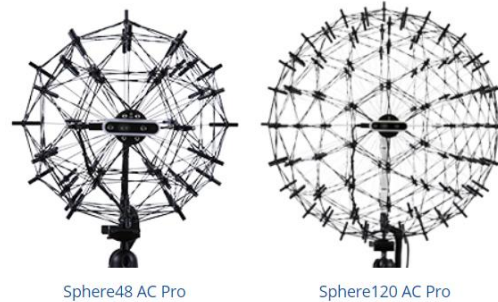


Figura 1.14. Cámaras acústicas 3D [VIII].

Las cámaras 3D tienen en cuenta las tres dimensiones en el espacio, por lo que los arrays con los que cuentan son tridimensionales. Este tipo de cámaras solucionan el error presente en las cámaras 2D. Son más precisas, pero pueden verse afectadas por sonidos que no forman parte del modelo particular estudiado. Presentan una clara diferencia con el anterior tipo y es que estas sí pueden emplearse en espacios cerrados, con la condición de que los micrófonos en el array sean omnidireccionales.

El principio de funcionamiento de las cámaras acústicas actuales se basa en lo que se conoce como Beamforming. Este método utiliza un array más un sistema de adquisición de datos multicanal que recolecta información para cada uno de los micrófonos de forma simultánea.

Tras realizar un procesamiento de los datos, se obtiene un mapa sonoro de la fuente, la cual puede incorporarse a la imagen obtenida a través de la cámara del conjunto, obteniendo así un conjunto donde se puede ver el mapa sonoro. El Beamforming, es flexible permitiendo realizar el procesado en el dominio del tiempo o de la frecuencia.

La técnica de Beamforming [11] emplea los retardos que se dan en los micrófonos, la suma de estos y la fase con la que la señal llega a cada uno de los micrófonos. La obtención de estos datos se define tomando como punto de referencia la distancia que existe entre la fuente y los micrófonos. Con todos estos datos, se consigue obtener un conjunto de valores que representan el mapa sonoro. Esto queda reflejado en Figura 1.15.

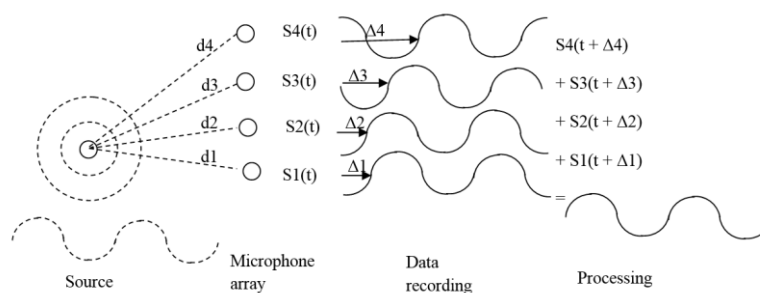


Figura 1.15. Técnica de Beamforming [11].

Para poder llevar a cabo esta técnica, se deben definir dos parámetros: el primero es la capacidad dinámica (esto se refiere a la comparación entre las fuentes virtuales y las reales) y la resolución dinámica (habilidad para distinguir dos fuentes que se encuentren a poca distancia). Estas variables están relacionadas con el tamaño de array, la distribución de los micrófonos en el array, la distancia a la fuente y la frecuencia.

### 1.3.3. TEORÍA ACÚSTICA

A lo largo de este apartado, se muestra la teoría relacionada con las ondas sonoras. Se explican términos necesarios para comprender ciertos conceptos asociados con la Acústica, así como los fundamentos de la directividad sonora.

#### 1.3.3.1. FUNDAMENTOS DEL SONIDO

Uno de los pilares de este TFG es el tratamiento del sonido. Sin embargo, se debe entender qué es exactamente. Para el desarrollo de este apartado se emplearon distintos recursos como [5].

El sonido es un fenómeno ondulatorio, que se puede definir como una perturbación mecánica de las partículas en un medio elástico. Esta vibración es captada por nuestros oídos. El medio de transmisión más habitual es el aire, aunque también se puede transmitir por líquidos y sólidos.

En primer lugar, se debe analizar el comportamiento del sonido como movimiento ondulatorio de tipo longitudinal (ya que, en este tipo de ondas, el movimiento de las partículas se produce en una dirección paralela a la propagación de la onda).

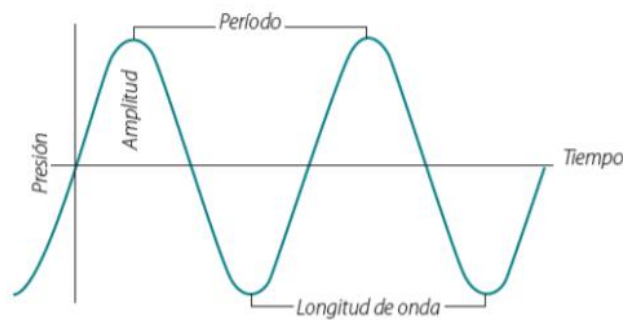


Figura 1.16. Parámetros de un movimiento ondulatorio [IX].

Los movimientos ondulatorios, de forma genérica se definen por algunos parámetros:

- La amplitud ( $A$ ) es la máxima variación de desplazamiento que se produce en la onda.
- El periodo ( $T$ ) es el tiempo que transcurre entre dos puntos consecutivos que se encuentran en la misma posición en el eje de ordenadas.
- La longitud de onda ( $\lambda$ ) es la distancia que se presenta entre dos puntos sucesivos con el mismo estado de vibración.

A partir de estos parámetros básicos, se puede obtener algunas relaciones, consiguiendo otros parámetros:

- La velocidad de propagación de la onda ( $v$ ), que es la distancia que recorre la onda en un segundo y se define como el cociente entre  $\lambda$  y  $T$ .
- La frecuencia de la onda ( $f$ ) es el número de pulsaciones que se producen en la onda por unidad de tiempo, calculado como la inversa del periodo.

El sonido, a su vez, puede separarse en sonidos de distintas frecuencias. La distribución de la energía sonora en cada una de las frecuencias que se obtiene a través de la división del sonido principal, se denomina espectro. Un sonido puede contener frecuencias bajas (graves), medias o altas (agudas).

El análisis de frecuencias, se realiza descomponiendo el sonido a estudiar en distintas bandas de frecuencia, definiendo el nivel de presión sonora de cada una de ellas. Las bandas a analizar, pueden tener un ancho constante o proporcional a la frecuencia central.

Las bandas más utilizadas son las que tienen una relación de proporcionalidad con la frecuencia central, llamado filtros por octava y por tercios de octava. En este caso, cada octava y tercio de octava se referencian en función de la frecuencia central, que es 1000 Hz.

Si definimos  $f_1$  y  $f_2$  como las frecuencias extremas de cada banda, la anchura de las bandas quedará:

- Para las bandas de octava:  $f_2 - f_1 = 0,707 \cdot f_c$  (con  $f_2 = 2 \cdot f_1$ ).
- Para las bandas de tercio de octava, se tiene:  $f_2 - f_1 = 0,232 \cdot f_c$  (con  $f_2 = \sqrt[3]{2} \cdot f_1$ ).

OCTAVAS en Hz	1/3 OCTAVAS en Hz
	16
	20
	25
31,5	31,5
	40
	50
63	63
	80
	100
125	125
	160
	200
250	250
	315
	400
500	500
	630
	800
1000	1000
	1250
	1600
2000	2000
	2500
	3150
4000	4000
	5000
	6300
8000	8000
	10000
	12500
16000	16000
	20000

Figura 1.17. Bandas en octavas y tercios de octavas [5].

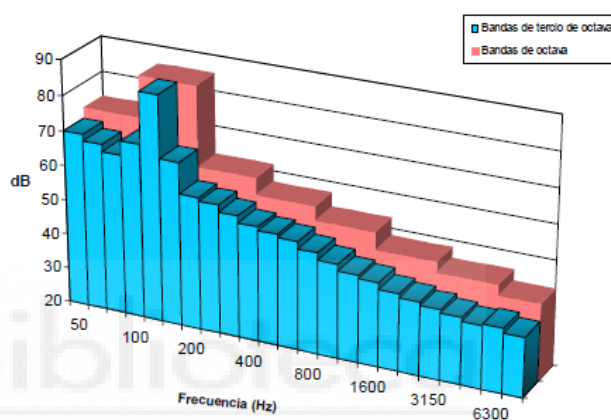


Figura 1.18. Comparación de espectro de frecuencias con en octavas y en tercio de octavas [5].

Cabe destacar, que la precisión obtenida con un filtro en tercios de octava presenta una mayor precisión que un filtro en octavas, ya que el rango de bandas empleadas es mayor. Sin embargo, a la hora de realizar un análisis con un filtro en octavas los resultados de nivel de presión sonora serán superiores a los que se obtendría con el mismo análisis en tercios de octava, puesto que en el primer filtro engloba el nivel de presión sonora de los tres tercios de octava que presenta.

Se debe comentar también el fenómeno ondulatorio del sonido en términos energético. Así, se pueden definir más términos:

- La potencia sonora  $W$  (o acústica) es la cantidad de energía que emite una fuente de sonido determinada. La potencia sonora únicamente depende de las características de la fuente sonora.
- La intensidad sonora  $I$  (o acústica) se conoce como la cantidad de energía sonora transmitida por unidad de área, en una dirección en particular. Así, esto representaría el cociente de la potencia  $W$  con el área de exposición, que generalmente será una esfera ( $4\pi r^2$ ). La intensidad también puede calcularse como el cociente entre la presión al

cuadrado y el producto de la densidad del medio con la velocidad de propagación de la onda.

- La presión sonora se define como la alteración que se produce en la presión atmosférica como consecuencia de la vibración que causan las ondas sonoras. Esta presión se determina a través de la diferencia entre la presión instantánea y la atmosférica, en un momento concreto. Una característica de la presión sonora es que es inversamente proporcional al cuadrado de la distancia.

Para medir la presión acústica, es habitual emplear un sistema de medición en consonancia con la sensibilidad del oído humano. El rango es muy extenso, por lo que se recurre al uso de una escala logarítmica, definiéndose, de este modo, el decibelio. Así, un valor de presión acústica en decibelios se calcularía como:

$$L_p = 10 \log_{10} \left[ \frac{P}{P_0} \right]^2 \quad (1)$$

Donde:

- $P_0$  es el umbral de audición humano ( $2 \cdot 10^{-5}$  Pa).
- $P$  es el valor de presión eficaz en Pa.

Por último, cabe destacar la existencia de una serie de índices acústicos básicos. La existencia de estos índices, se debe a la necesidad de dar respuesta a una serie de necesidades que puedan surgir a la hora de realizar un estudio. Dentro de la gran cantidad de índices que existen, se destaca solamente uno, que es el nivel sonoro continuo equivalente ( $L_{Aeq}(t)$ ).

El  $L_{Aeq}(t)$  se encarga de expresar el nivel de energía sonora media que es percibido por un individuo en un intervalo de tiempo determinado. Esto significa que, este indicador, realiza una equivalencia, definiendo el sonido realmente percibido con un nivel de presión sonora que se produciría en el caso de que el sonido fuese constante. De esta forma, trabajar con la señal se torna más sencilla. Este indicador suele emplearse con una ponderación frecuencial de tipo A.

$$L_{Aeq}(t) = 10 \log_{10} \left( \frac{1}{T} \int_T \left[ \frac{P_1}{P_0} \right]^2 dt \right) \quad (2)$$

Donde:

- T es el tiempo total de la medición en s.
- $P_i$  es la presión sonora instantánea en Pa.
- $P_0$  es la presión de referencia.

Sin embargo, este índice a su vez puede ponderarse temporalmente, usando un filtro tipo Fast, Slow o Impulse [X]. Cuando se desea realizar esto, la expresión a emplear es la siguiente.

$$L_{Ateq}(t) = 10 \log_{10} \left[ \frac{(1/\tau) \int_{t_s}^t P_A^2(\epsilon) e^{-(t-\epsilon)/\tau} d\epsilon}{P_0^2} \right] \quad (3)$$

Donde:

- $P_A^2(\epsilon)$  es la presión sonora instantánea con ponderación A al cuadrado.
- $\epsilon$  es una variable de integración ficticia.
- $P_0$  es la presión de referencia.
- $\tau$  constante nominal para ponderación temporal según el tipo que queramos usar.

### 1.3.3.2. DIRECTIVIDAD

Antes de proceder con la explicación de la teoría referente a la directividad, se debe diferenciar entre fuentes omnidireccionales y direccionales. Las fuentes omnidireccionales son aquellas que llevan a cabo emisiones iguales en todas las direcciones. En contraposición, las fuentes direccionales son aquellas en donde las emisiones son mayores en unas direcciones que en otras [3].

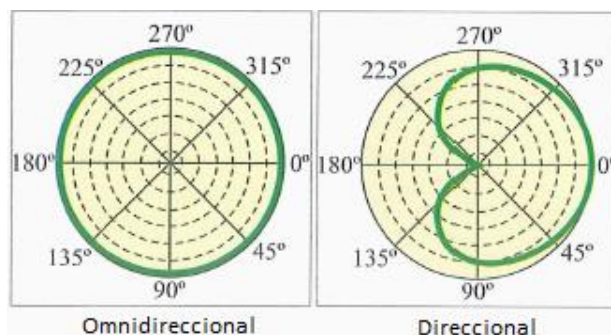


Figura 1.19. Comparación de la emisión de una fuente omnidireccional y direccional [XI].

Sin embargo, no existen las fuentes omnidireccionales perfectas, por lo tanto, todas las fuentes presentan cierto grado de directividad en mayor o menor medida. Para describir el comportamiento de la fuente, se usa lo que se conoce como coeficiente de directividad. Este coeficiente se define como la relación entre la intensidad sonora en un punto de una dirección específica y la intensidad que habría si la fuente fuese omnidireccional, teniendo la misma potencia sonora.

$$Q(\theta, \phi) = \frac{I(\theta, \phi)}{I} \quad (4)$$

Donde:

- $I(\theta, \phi)$  es la intensidad en una dirección concreta de una fuente direccional.
- $I$  es la intensidad de una fuente omnidireccional.

Si se tiene en cuenta la definición de intensidad sonora, se puede observar que tiene la siguiente forma.

$$I = \frac{\dot{W}}{4\pi r^2} \quad (5)$$

Donde:

- $\dot{W}$  es la potencia sonora de la fuente en  $W$ .
- $4\pi r^2$  es la superficie de una esfera.

Y expresada a cierta distancia de la fuente, se puede expresar como:



$$I = \frac{P_{\text{eff}}^2}{\rho_0 c} \quad (6)$$

Donde:

- $P_{\text{eff}}$  es la presión acústica en Pa.
- $\rho_0 c$  es la impedancia acústica específica del medio (resistencia de un medio físico al paso de ondas sonoras, en nuestro caso el aire).

Si se desea comparar la intensidad sonora de una fuente direccional con respecto a otra omnidireccional, se puede realizar un cociente empleando en el numerador y en el denominador la expresión (6) refiriéndola a cada fuente. Por lo tanto, quedaría:

$$\text{Relación} = \frac{P(\theta, \phi)^2 / \rho_0 c}{P^2 / \rho_0 c} = \frac{P(\theta, \phi)^2}{P^2} \quad (7)$$

Donde:

- $P(\theta, \phi)$  es la presión sonora en Pa en un punto determinado de una dirección específica de una fuente direccional.
- $P$  es la presión sonora en Pa de la fuente omnidireccional.

De esta forma, se puede calcular el coeficiente de directividad por medio de valores de presión sonora en Pa. Pero no solo eso, también se puede determinar a partir de la presión sonora en dB. Por lo tanto, la expresión quedaría de la siguiente forma:

$$Q(\theta, \phi) = \frac{I(\theta, \phi)}{I} = \frac{P(\theta, \phi)^2}{P^2} = 10^{\frac{Lp(\theta, \phi) - Lp}{10}} \quad (8)$$

Donde:

- $Lp(\theta, \phi)$  es la presión sonora en un punto determinado de una dirección específica de una fuente direccional en dB.
- $Lp$  es la presión sonora en dB en el punto de recepción si la emisión fuera de una fuente omnidireccional.

Una vez conocido como calcular el coeficiente de directividad, solo resta conocer como es la expresión para conseguir el índice de directividad, el cual se expresa en dB y es el que se usará en este trabajo para representar la directividad:

$$D(\theta, \phi) = 10 \log_{10} Q(\theta, \phi) = L_p(\theta, \phi) - L_p \quad (9)$$

Básicamente el índice de directividad consiste en convertir a dB el factor de directividad si este se ha calculado en Pa, o calcular la diferencia entre la presión sonora en dB direccional y omnidireccional. Cabe destacar, que los valores mayores que 0 del índice de directividad corresponderán con direcciones donde la emisión es mayor que la que habría en una fuente omnidireccional (la media). Por otro lado, y en el caso de valores negativos, se refieren a direcciones donde la emisión es menor que la media.

## 2. ANTECEDENTES

En este apartado, se comentan brevemente los antecedentes y aspectos previos a la investigación de este TFG. Se comienza introduciendo la problemática del sonido en los vehículos eléctricos, y las vías actuales de desarrollo en cuanto a sistemas de alarma se refiere. Se tratan las ondas sonoras y la percepción por parte del ser humano. Se realiza también una revisión bibliográfica del estado del arte hasta el momento sobre este tema, así como los reglamentos adoptados como referencia.

### 2.1. PROBLEMÁTICA DEL VEHÍCULO ELÉCTRICO

A diferencia de los coches con motores de combustión interna, los coches eléctricos son silenciosos. Esto puede parecer una ventaja, ya que con el uso de estas tecnologías la contaminación acústica en los núcleos poblaciones disminuirán. Según la OMS, las personas que conviven con este tipo de contaminación sufren de algunos daños físicos y psicológicos, como puede ser la pérdida de audición o el estrés o irritabilidad [XII].

No obstante, que los vehículos eléctricos sean silenciosos también es un problema. Un estudio publicado en 2014, estimó que los vehículos híbridos y eléctricos causan un 40% más de atropellos a peatones que los convencionales [XIII].

Similarmente, en 2008, se comparó a qué distancias tenían que permanecer los peatones de distintos vehículos para escucharlos y deducir desde qué dirección se aproximaban. Se concluyó que la distancia requerida para escuchar a los vehículos híbridos y eléctricos era un 74% mayor de la necesitada para escuchar a los convencionales [XIV].

Los colectivos más susceptibles a sufrir un accidente involucrando un coche silencioso son precisamente los más vulnerables: los niños, las personas de avanzada edad y el colectivo de personas con diversidad funcional. La ONCE manifestó en 2015 que la probabilidad de choque de un automóvil silencioso con un peatón se incrementa en un 40% respecto del choque con vehículos audibles [XV].

En esta línea, Glaeser et al. [12] llevaron a cabo un ensayo de detectabilidad de vehículos silenciosos con 12 personas con discapacidad visual. A velocidades muy bajas, de 10 km/h, el vehículo tan solo se detectaba a unos pocos metros de distancia del sujeto (limitando su capacidad de reacción). A partir de una velocidad de 30 km/h, el sonido percibido provenía en gran medida de la rodadura del coche y el efecto aerodinámico del mismo.

Altinsoy [13], también concluyó que el tiempo de reacción del peatón ante la aproximación de un vehículo era mayor si el coche es eléctrico, en una prueba en la que 37 participantes compararon 14 vehículos con motores convencionales, 4 híbridos y 6 eléctricos.

Este problema de sonoridad en los coches eléctricos fue tratado, en primer lugar, por la legislación japonesa. En 2010 se establecieron las directrices y requerimientos básicos que los dispositivos emisores debían cumplir, como el tipo de sonido y duración, entre otros [14].

En siguiente lugar, se desarrolló la legislación de Estados Unidos. En 2014, la NHTSA aprobó la obligatoriedad de incorporación de señales de aviso (warning sounds) en los vehículos eléctricos, permitiendo a los fabricantes una moratoria de 3 años para adaptarse a la norma. Estas señales sonoras debían emitirse cuando el vehículo circulase por debajo de los 30 km/h. Además, debían cumplir ciertas directrices, como que el sonido emitido tenía que ser similar al de un vehículo convencional o que la desactivación manual estuviese prohibida [15].

La Unión Europea estableció las bases, en 2014, del desarrollo de un sistema propio de alerta sonora en vehículos, activado para velocidades menores a 20 km/h. Estos sistemas de ruido artificial y simulado son conocidos como AVAS (Acoustic Vehicle Alerting System), cuya traducción sería SAAV.

## 2.2. TIPOS DE SAAV

En general, esta tecnología consiste en la emisión de sonido mediante altavoces colocados en el vehículo, normalmente en la parte delantera, manejados a través de una unidad de control que se encarga de almacenar los sonidos a emitir. El sonido deberá ser continuo y conseguir que el vehículo sea detectable, sin resultar molesto. Su objetivo principal será el de avisar al peatón,

sin producir contaminación acústica. El sonido será fácilmente audible, localizable y socialmente aceptable. El diseño del SAAV resulta complejo, ya que son numerosos los parámetros que se deben tener en cuenta a la hora de diseñar los mismos.

Existen numerosas aplicaciones del SAAV que han sido diseñadas por fabricantes especializados en el tema.

La compañía Enhanced Vehicle Acoustics ideó un sistema basado en pequeños altavoces colocados dentro del alojamiento de las ruedas [XVI]. Este sistema emite sonidos de forma direccional, cambiando la emisión en función de la dirección que el vehículo tome. Los altavoces trabajan mientras el vehículo circula en modo eléctrico a bajas velocidades.

Brigade ha desarrollado un sistema de sonido de varias frecuencias para vehículos silenciosos. Esta tecnología utiliza sonidos con frecuencias de sonido blanco. La emisión se percibe como un siseo suave para el oído y al alejarse el vehículo, el sonido desaparece. Además, el sistema produce una emisión direccional del sonido, haciéndolo solamente perceptible en las zonas problemáticas. El tono y el volumen aumentan automáticamente con la velocidad del vehículo imitando a un motor de combustión interna, hasta los 30 km/h [XVII].



Figura 2.01. Altavoz de la marca Brigade [XVII].



Figura 2.02. Unidad de control de sonido de la marca Brigade [XVII].

Kendrion, también desarrolla altavoces activos, los cuales son instalados en vehículos híbridos y eléctricos [XVIII]. El sistema de altavoces dispone de un altavoz y un controlador, que reproduce un sonido que cumple con los requisitos globales de nivel de sonido. Este sistema posee un diseño robusto que le protege ante las diferencias de temperatura, además de quedar protegido ante la entrada de agentes externos. A su vez, el sistema puede presentar uno o dos canales permitiendo de este modo la instalación de varios sonidos. El software de diseño del sonido, Kendrion Automotive, permite la creación del ruido emitido por un vehículo de una marca

concreta. Para ello, se basan en variables como la velocidad del motor, la carga, y la posición del pedal.

La compañía alemana Hella también ha desarrollado un SAAV que puede instalarse en vehículos con diferentes finalidades: vehículos de pasajeros, comerciales o autobuses [XIX]. El SAAV de esta compañía es fabricado con una carcasa impermeable y en su interior se introduce un sensor que emite un sonido estándar, aunque también pueden definirse ciertas características del sonido por el cliente. A medida que aumenta la velocidad, se incrementa la intensidad del sonido y una vez que el vehículo supera los 20 km/h, se inactiva. A su vez, el sistema de advertencia genera una señal perceptible cuando el vehículo circula marcha atrás. Este SAAV simula un ruido artificial de al menos 56 decibelios.

A pesar de que las empresas mencionadas anteriormente han proporcionado soluciones eficientes, los propios fabricantes de coches también han desarrollado sus propios sistemas SAAV. Algunos ejemplos se presentan a continuación:

- General Motors introdujo su primer sistema de advertencia en el 2010, cuya activación se realizaba manualmente por el conductor. El sistema se denominó Pedestrian-Friendly Alert System [XX].
- En ese mismo año, Hyundai desarrolló el VESS. El modelo Sonata Hybrid, cuya fabricación en cadena de producción comenzó en 2011, incorporaba este sistema [XXI].
- Por otro lado, el fabricante Lotus incorporó en 2010 el sistema HALOsonic Internal and External Electronic Sound Synthesis a su modelo Evora 414E Híbrido [XX]. Este SAAV comprende una colección de sonidos en colaboración con Harman International. Este sistema genera el ruido del motor, que puede ser perceptible en el interior del habitáculo a través del sistema de audio en el interior del coche. A su vez, el sonido producido es emitido al exterior a través de altavoces incorporados en la parte delantera y trasera del vehículo. Se permite que el usuario escoja el sonido que prefiera entre cuatro opciones distintas.
- El SAAV ideado por Nissan, y aplicado entre otros en el Leaf y Fuga, recibe el nombre de VSP. Esta alerta comprende un sonido para la marcha normal y otro para la marcha atrás. Su sistema es capaz de alertar a los peatones evitando que los conductores se distraigan. Esto se consiguió a través de la colaboración entre Nissan e instituciones reconocidas en psicología cognitiva y acústica, como la Federación Nacional de Ciegos de EEUU y el Instituto de Oftalmología de Detroit. El altavoz está localizado en el guardabarros del lado del conductor y emite un tono de frecuencia variable en función de la aceleración o

desaceleración del vehículo. El sistema se desactiva cuando el vehículo alcanza los 30 km/h [XXII].

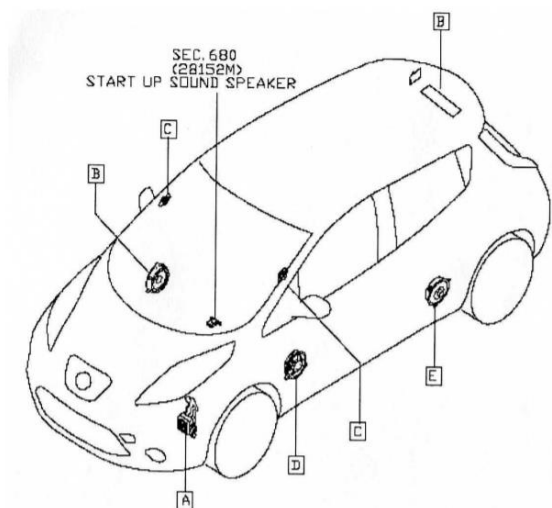


Figura 2.03. Disposición de los elementos del SAAV en el modelo Nissan Leaf [XXIV].

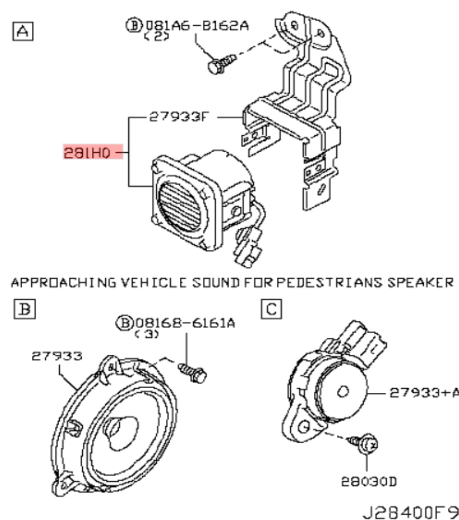


Figura 2.04. Componentes del SAAV del modelo Nissan Leaf [XXIV].

- Toyota contrató a Fujitsu para realizar su SAAV. Una versión preliminar del sistema se introdujo en los Prius. Este emitía un sonido sintetizado del motor eléctrico cuando el vehículo circulaba en modo eléctrico por debajo de los 25 km/h. El tono de este sonido aumentaba y descendía con la velocidad. A partir del 2012, esta versión se convirtió en el sistema VPNS, instalado en todos los automóviles de la familia Prius [XXIII].
- Jaguar diseñó un SAAV para su modelo I-PACE [XXV]. El sistema emite un sonido que se puede escuchar a velocidades de hasta 20 km/h. Este sonido, que es audible y discreto, supera el mínimo exigido de 56 dB y se emite a través de un altavoz situado detrás de la parrilla delantera. El sonido aumenta en tono y volumen según se acelera. Además, cuando se circula marcha atrás, este se acompaña de un tono adicional que indica el cambio de dirección. En cambio, el sistema no permite la percepción del sonido en el interior del vehículo.

Por otro lado, también se han desarrollado proyectos con el objetivo de encontrar una solución viable al problema de la baja detectabilidad de los vehículos, como es el caso de eVADER [XX]. Este proyecto fue financiado por la UE y tuvo como resultado la obtención de un prototipo de sistema de alerta inteligente. Este sistema inteligente, es capaz de hallar los

usuarios de la vía que se encuentren en una situación de riesgo y emite una señal de advertencia hacia ellos. De esta manera, la contaminación acústica producida en los alrededores es mínima.

Además, el sistema es capaz de optimizar el volumen al que debe ser emitido cada uno de los sonidos de advertencia. Esto se realiza a través del análisis del ruido de fondo, consiguiendo así que la alerta se perciba con un nivel sonoro correcto.

Diversas compañías automovilísticas se involucraron en el proyecto. Algunos ejemplos fueron Renault, Nissan o PSA Peugeot Citroën. Además, otras instituciones como universidades o institutos también cooperaron en el mismo.

El sistema conseguido a través de esta investigación, está basado en seis altavoces instalados en la parte delantera del vehículo. Estos altavoces generan una emisión direccional que apunta hacia unas direcciones mientras que en otras se anula. A su vez, el dispositivo es capaz de detectar a los demás usuarios de la vía, a través de radares y cámaras instalados en el vehículo. También se recopila información acerca del ruido ambiental a través de los micrófonos. Todos los datos recopilados son enviados a la unidad de control del sistema. Esta se encargará de proporcionar una respuesta satisfactoria que avise a las personas que se encuentren en una zona de riesgo.

Sin embargo, para asegurar la alerta del individuo, la respuesta generada por el sistema de advertencia se acompaña de un sonido omnidireccional de bajo nivel. El siguiente paso de este proyecto, tiene por objeto integrar un método de alerta para cuando se circule marcha atrás.

### **2.3. ESTADO DEL ARTE**

Para continuar, merece la pena realizar una pequeña revisión bibliográfica sobre algunos de los artículos más relevantes publicados hasta el momento, acerca de los SAAV y su aplicación en los coches eléctricos.

Wogalter et al. [16] realizaron tres estudios, en los que se trató los SAAV de forma cualitativa. En el primero, se preguntó a 378 participantes sus opiniones sobre los coches de energía alternativa, así como las posibles advertencias sonoras. En el segundo estudio, otro grupo de participantes evaluó el nivel de aceptabilidad de 14 sonidos. En el tercero se evaluaron 18 sonidos reales, agrupados en 6 categorías.

Un 72% de los participantes consideraron la compra de un vehículo eléctrico y un 70% opinó que la falta de sonido era un peligro para los peatones. De los sonidos reales probados, la categoría que más aceptación tuvo fue la que emulaba un motor, seguida de los ruidos blancos.

Fleury et al. [17] diseñaron un experimento para probar 5 hipótesis, 3 de las cuáles ya se obtuvieron de varios estudios (entre los que destacan Misdariis et al. [18,19] y Emerson et al. [20]). La primera y más básica es que la incorporación de un SAAV a un vehículo eléctrico incrementa su detectabilidad. Según la segunda, un aumento de tono también mejora la detectabilidad del vehículo. En la tercera, se estableció que las variaciones de frecuencia reflejando aceleración y desaceleración debían mejorar la detectabilidad.

Así, los investigadores añadieron dos hipótesis propias más, aunque se nutren de Sun et al. [21] y Chamard et al. [22]. Según una, los participantes del estudio debían sobreestimar la velocidad de un coche a 10 km/h, e incluso más si emiten sonidos con tonos mayores. En la segunda, se quiso demostrar que la eficiencia del SAAV percibida era mayor para un sonido modulado con tonalidad, más que para un sonido neutral.

En total, se consiguieron probar las hipótesis planteadas, mediante dos estudios. Una limitación de esta investigación es que solamente se emplearon participantes jóvenes, por lo que la población adulta y anciana (aquellos que han sufrido una degeneración de sus facultades) fueron relegados.

Paritzel et al. [23], dentro del proyecto eVADER (financiado por la UE), diseñaron un conjunto de señales de advertencia modificando tres características fundamentales. Los cambios realizados fueron la modulación de frecuencia, modulación de amplitud y tipo de componentes predominantes (tonos o armónicos). Estas señales fueron probadas en escenarios de carreteras más realistas, con diferentes condiciones meteorológicas. A su vez, se utilizaron en el estudio personas sin problemas de visibilidad y una muestra relativamente amplia de personas con algún tipo de discapacidad visual.

La presencia de SAAV contribuyeron a una mayor mejora en las respuestas de las personas ciegas que en las de las personas sin problemas visuales, pero solo en condiciones de escucha ideales (sin interferencia de lluvia, por ejemplo). Los efectos de la variación de timbre fueron similares para ambos grupos, apoyando la idea de obtener un SAAV universal que satisfaga a las dos agrupaciones. La conclusión a la que se llegó en el estudio fue que aquellos sonidos con ciertas componentes armónicas con modulación en amplitud, pero sin modulación de frecuencia, era una de las mejores opciones para mejorar la detectabilidad. Sin embargo, fueron dos sonidos de advertencia con una amplitud modulada muy irregular las que mejores resultados obtuvieron en los tests empíricos, detectándose al mismo nivel que un motor diésel. Estas dos señales coincidían también con los dos menores niveles de presión acústica.



Van der Rots et al. [24] estudiaron la aplicabilidad de algoritmos para la generación de sonido direccional, mediante bobinas de voz instaladas en el exterior de los vehículos. Se realizaron unas pruebas experimentales y numéricas, con el fin de conseguir un sonido capaz de advertir a los peatones y a la vez no contribuir a la contaminación acústica. Los métodos de generación SPM (sound power minimisation) y AED (acoustic energy difference) obtuvieron la mayor direccionalidad.

A su vez se concluyó que, buscando la optimización del haz de sonido para distancias largas del array de origen, la forma de ese haz también se comportaba de forma adecuada para distancias más cortas. No obstante, el haz de sonido se podía optimizar de forma más concreta para cada distancia.

En Poveda et al. [25], se realizó un test de audición en una situación de cruce de carretera. Este experimento se ejecutó en un laboratorio, mediante el uso de auriculares. Se presentaron varios sonidos de advertencia, en combinación con ruidos de fondo. Existieron dos variantes básicas de señales de advertencia testeadas: una que emulaba el sonido de un motor de combustión interna, y otra con un pequeño número de componentes tonales y un poco de modulación. En total, se probaron 8 sonidos de advertencia. Los ambientes seleccionados para la simulación fueron una vía de tres carriles con semáforo, una calle peatonal de una zona comercial, y un parque con voces de niños y adultos.

En aquellas zonas más ruidosas, hubo más errores de detección. No hubo una relación significativa entre los errores de detección y la edad de los participantes en el estudio de escucha. Los errores de detección fueron ligeramente mayores en los vehículos eléctricos o híbridos, cuando circulaban en modo eléctrico, comparado con los otros modos. Aquellos ambientes más ruidosos enmascararon la aproximación de los vehículos, dificultando su detección.

En Steinbach et al. [26], se realizó una revisión de la influencia de los SAAV en la seguridad de los viandantes más vulnerables. En los ensayos, un mayor nivel de SAAV pasaba menos desapercibido en el ruido de fondo, detectándose más fácilmente. Sin embargo, también demostró que un salto en el nivel de emisión mejoraba significativamente la detección en la parada y reanudación de la marcha del coche. Un cambio a frecuencias elevadas, según los ensayos, producían una detección más temprana. A pesar de que se estudiaron más parámetros, se concluyó que el principal aumento de detectabilidad se producía debido al salto de nivel de emisión, mientras que el resto de ellos desempeñaban papeles menos relevantes.

## 2.4. NORMATIVA

En la actualidad, con la transformación del mercado automovilístico a vías más sostenibles, surgen problemas en la regulación de los SAAV. Así, la redacción de normativas está justificada para homogeneizar y establecer unos requisitos mínimos de calidad y características técnicas que deben cumplir los fabricantes de automóviles. Las directivas que tratan los SAAV más importantes son las de China, EEUU y la UE. En la siguiente figura, se resumen algunas diferencias de parámetros entre ambas.

Conditions	US	EU	China
Idle	✓	-	-
10 km/h	✓	✓	✓
20 km/h	✓	✓	✓
30 km/h	✓	-	-
Reverse	✓	✓	✓

Metrics	US	EU	China
1: Min overall level	Option 2	✓	✓
2: Max overall level	-	forward	forward
3: Third Octave Bands	✓	✓	✓
4: Pitch Shift	-	✓	✓
5: Volume Shift	✓	-	-

Min levels dB(A)	US	EU	China
Idle	44		
10 km/h	51	50	52
20 km/h	57	56	56
30 km/h	62		
Reverse	48	47	49

Figura 2.05. Comparativa entre normativas [27].

Por cuestión de localización, los ensayos realizados en este TFG siguen la normativa europea. La normativa de la UE recibe el nombre de Reglamento nº138 del CEPE, publicado en 2017 [28]. Este reglamento es aplicable a los vehículos eléctricos de las categorías M y N que tienen la capacidad de circular sin el funcionamiento de un motor de combustión. Se establece que el SAAV debe funcionar en el intervalo de 0 – 20 km/h. Toda la información relativa a este reglamento puede resumirse en unos pocos apartados, relegando al propio documento los aspectos más específicos.

### 2.4.1. CARACTERÍSTICAS ACÚSTICAS DE LOS SONIDOS

Los sonidos de los SAAV deberán cumplir todos los requisitos de los ensayos que se definen en el Anexo 3 del Reglamento. El fabricante puede dar a elegir al conductor diversos sonidos de SAAV, siempre que cumplan con la norma. Cuando el vehículo se encuentre parado podrá emitir sonido. No obstante, se permite la desactivación temporal del SAAV cuando el

vehículo se encuentre inmóvil, siempre que vuelva a activarse cuando el vehículo vuelva a arrancar. Los SAAV no emitirán, cuando circule hacia delante, un nivel sonoro global superior a los 75 dB(A) para una distancia de 2 m. Además, debe cumplir con los umbrales mínimos de sonidos que se encuentran en la tabla de la Figura 2.06.

#### **2.4.2. ENSAYO DE VELOCIDAD CONSTANTE**

Las velocidades de ensayo para la homologación son de 10 km/h y 20 km/h. El sonido emitido por el vehículo debe tener al menos dos de las bandas del tercio de octava de acuerdo con la Figura 2.06. Al menos una de esas bandas estará por debajo o dentro de la de 1600 Hz. Asimismo, deberá contar con unos niveles de presión acústica mínimos, según las bandas elegidas, de acuerdo con las Columnas 3 o 4 de la Figura 2.06.

Los ensayos a velocidad constante se pueden realizar, o bien con el vehículo desplazándose hacia delante, o simulando la velocidad de este aplicando una señal externa dirigida al SAAV mientras el vehículo se encuentra parado.

#### **2.4.3. ENSAYO DE MARCHA ATRÁS**

El vehículo, marcha atrás, debe emitir un sonido con nivel acústico mínimo al establecido en la Columna 5 de la Figura 2.06. Estos ensayos, a su vez, pueden realizarse con el vehículo desplazándose hacia atrás o con la velocidad simulada mediante señal externa dirigida al SAAV (con el vehículo parado).

#### **2.4.4. CAMBIOS DE FRECUENCIA PARA LA ACELERACIÓN Y DESACELERACIÓN**

El sonido emitido debe presentar, como mínimo, un tono que varíe proporcionalmente con la velocidad dentro de cada relación de transmisión con un porcentaje de al menos un 0.8% por cada km/h en el intervalo de velocidades de 5-20 km/h.

Los ensayos de cambio de frecuencia pueden realizarse según cinco formas posibles, de los que el fabricante elige uno:

- En el Método A, se realiza un ensayo del vehículo moviéndose en una pista al aire libre.
- En el B, se realiza con el vehículo parado, en una pista al aire libre y con simulación del movimiento dirigida al SAAV mediante un generador externo de señales.

- En el Método C, se ensaya el vehículo en movimiento en una instalación interior, sobre banco dinamométrico.
- En el D, se ensaya el vehículo parado en instalación interior con simulación del movimiento dirigida al SAAV mediante un generador externo de señales.
- En el Método E se ensaya el SAAV sin vehículo alguno, en instalación interior y con simulación del movimiento dirigida al SAAV.

**Requisitos relativos al nivel sonoro mínimo en dB(A)**

Frecuencia en Hz		Ensayo de velocidad constante, punto 3.3.2 (10 km/h)	Ensayo de velocidad constante, punto 3.3.2 (20 km/h)	Ensayo marcha atrás, punto 3.3.3
Columna 1	Columna 2	Columna 3	Columna 4	Columna 5
Global		50	56	47
Banda de tercio de octava	160	45	50	Biblioteca Miguel Hernández
	200	44	49	
	250	43	48	
	315	44	49	
	400	45	50	
	500	45	50	
	630	46	51	
	800	46	51	
	1 000	46	51	
	1 250	46	51	
	1 600	44	49	
	2 000	42	47	
	2 500	39	44	
	3 150	36	41	
	4 000	34	39	
5 000	31	36		

Figura 2.06: Cuadro 2 del Reglamento nº 138 [28].

### 3. OBJETIVOS

Los sistemas SAAV han supuesto una propuesta para abordar el problema de la baja detectabilidad en los vehículos eléctricos. Por ello, se realizó un ensayo pass-by, haciendo circular un coche eléctrico al que se le instaló un sistema de alerta que contenía distintos sonidos de advertencia. Cada sonido de advertencia se estudió a distintas velocidades, según define la normativa vigente en la UE.

Las señales y datos que se obtuvieron en los ensayos fueron obtenidas y procesadas por medio de una cámara acústica. De esta forma, a través de la información que se obtuvo, se pueden distinguir dos partes en este TFG. La primera consiste en determinar el nivel sonoro continuo equivalente máximo y el espectro de frecuencias de un intervalo que contenga dicho punto, para todas las señales recogidas. Por otro lado, la segunda parte desarrolla el estudio de la directividad vertical para cada medición realizada durante los ensayos.

El objetivo principal de la primera parte es obtener el nivel de presión sonora equivalente máximo en dB(A), para compararlo con normativa y determinar si el sonido de advertencia se encuentra dentro de los máximos y mínimos niveles establecidos. A su vez, el espectro de frecuencias podrá ser comparado con la normativa, y aumentar el rango de precisión en la validez del sonido. También se extraerán conclusiones del análisis de todos los sonidos de advertencia estudiados y así designar la eficiencia de cada sonido.

En la segunda parte, la finalidad es determinar la directividad vertical generada por cada sonido de advertencia, a través del cálculo del índice de directividad. De esta forma se conseguirá establecer un rango en el que el vehículo emitirá, por medio del sonido de advertencia activado, de forma más significativa y en que zonas emitirá en menor medida. También se expresará las diferencias en cuanto a forma de emisión existente entre los sonidos de advertencia del estudio.

Con todos los datos, el objetivo global del proyecto es la obtención de una serie de datos que permita definir como de eficaces serían los sonidos de advertencia al ser instalados en un vehículo eléctrico, en cada uno de los parámetros estudiados.

Para la consecución de dichos objetivos, conviene realizar varias tareas tal y como quedan reflejadas a continuación:

**A) Recopilación de datos de los ensayos.**

- Conocer la forma en que se realizan los ensayos pass-by.
- Conocer el modo de funcionamiento de la cámara acústica y el software de la misma.
- Desarrollo de los ensayos y recopilación de datos.

**B) Desarrollo de programas en Matlab que permitan la manipulación y transformación de los datos.**

- Conocer el marco teórico de los procedimientos a emplear y resultados a obtener.
- Generar los programas en Matlab que permitan determinar el nivel sonoro equivalente máximo y el espectro de frecuencias para cada una de las tomas de datos.
- Generar los programas en Matlab que permitan determinar la directividad vertical para cada una de las tomas de datos.

**C) Análisis de resultados.**

- Almacenamiento de los resultados proporcionados por los programas realizados en Matlab.
- Comparación de los resultados de nivel sonoro máximo equivalente y espectro de frecuencias con normativa.
- Desarrollo de conclusiones tanto para lo anterior como para la directividad vertical.

## 4. METODOLOGÍA

Este apartado tiene como objetivo explicar los sonidos que se embarcaron en el vehículo, así como el conjunto de tareas que se realizaron desde el momento en que se llevaron las mediciones en los ensayos hasta la generación de todos los programas en Matlab y la obtención de resultados. A su vez, se procederá a analizar los resultados finales y a determinar posibles conclusiones.

### 4.1. SONIDOS DE ADVERTENCIA EMBARCADOS

Los sonidos de advertencia que se instalaron en este vehículo para realizar los ensayos, fueron desarrollados a través de estudios generados por compañías especializadas, investigadores expertos o a través de regulaciones internacionales [29].

El diseño de estos sonidos de alerta, se basó en la modificación de una serie de parámetros. Estos parámetros fueron: la distribución de la energía sonora dentro del espectro de frecuencias, la modulación de la señal y los mínimos valores de emisión. A su vez, se empleó como referencia un tono puro de 1000 Hz.

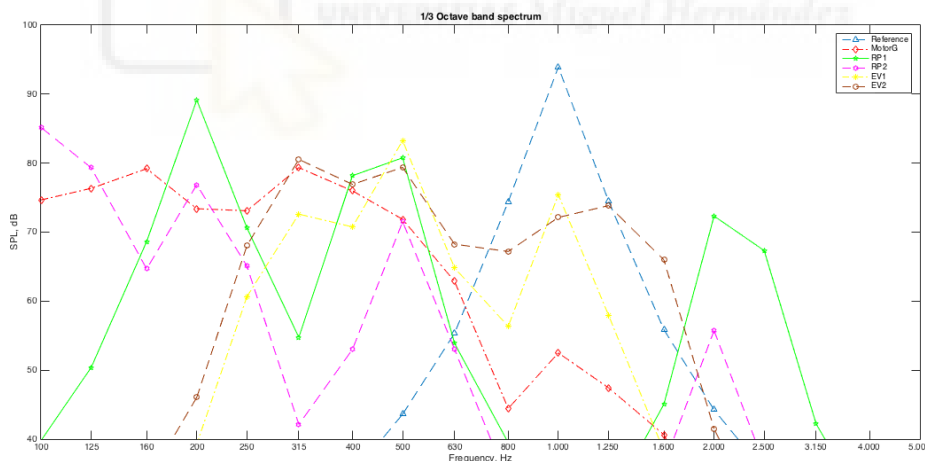


Figura 4.01. Gráfica que representa el comportamiento frecuencial de cada sonido de advertencia embarcado [29].

A continuación, se detallan cada una de las características de estos sonidos. Además, se asignará un código para cada una de estas señales.

- Sonido tonal de referencia. Este sonido se presenta como un tono puro de 1000 Hz. El código asignado a este sonido de advertencia es el E2.

- Sonido MotorGear. El creador de esta señal de advertencia fue Delta Senselab. Este sonido fue considerado como una señal ligeramente molesta. La energía de emisión se encuentra concentrada en el rango de frecuencias de 100 Hz a 500 Hz. Lleva asociado el código E3.
- Sonido propuesto 1 (RP1LL). El sonido de advertencia fue generado teniendo en consideración las restricciones de la UE. Este sonido de advertencia presenta una concentración de energía evidente en las frecuencias de 200 Hz, 400 Hz, 500 Hz y 2000 Hz. Este sonido tiene asignado el código E4.
- Sonido propuesto 2 (RP2LL). Como el sonido de advertencia RP1LL, se tuvo en cuenta las restricciones de la UE para diseñar este sonido. Este sonido lleva asociada una emisión con gran cantidad de energía por debajo de 200 Hz. El código asociado a este sonido es E1.
- Sonido de alerta 7stimulus7LL. Este sonido de advertencia fue desarrollado a través del proyecto eVADER. El sonido presenta modulación de frecuencia en forma de diente de sierra, con tres armónicos y con una extraña modulación de amplitud. Tiene asociado el código E5.
- Sonido de advertencia 3stimulus3LL. Este sonito también fue diseñado en el proyecto eVADER. Al contrario que el anterior sonido, este no presenta modulación frecuencial, con nueve armónicos. No obstante, se mantiene la modulación de amplitud rara. El código asociado es E6.

No obstante, a la hora de realizar los ensayos, también se tomaron datos de otros sonidos. La explicación de cada uno de estos, se detalla a continuación.

- Sonido producido por el vehículo sin ningún tipo de sonido de advertencia emitido. El sonido proviene básicamente de la rodadura y/o aerodinámica del coche. El código que se le asigna es el C1.
- Sonido del vehículo con emisión de la señal de advertencia del vehículo. Este sonido de alerta, fue diseñada por ingenieros de Renault (ZOE Voice\_PURE). El código que designa este sonido es C1 WS.

## 4.2. DESARROLLO DEL ENSAYO

Este apartado desarrolla la explicación acerca de los requisitos que se deben cumplir al realizar el ensayo y la forma de ejecución del mismo.



#### 4.2.1. EMPLAZAMIENTO DEL ENSAYO

La definición del lugar del ensayo permite obtener un espacio normalizado donde realizar las mediciones. El reglamento n°138 del CEPE [28] establece la posibilidad de realizar ensayos tanto en lugares exteriores como interiores. En nuestro caso, los ensayos se realizaron al aire libre.

Para las zonas de ensayo al aire libre, el reglamento establece una serie de condiciones. El emplazamiento debe ser llano, con una estructura y superficie de la pista que cumpla con lo expuesto en la norma ISO 10844:2014 [30].

Por otro lado, no puede existir ningún tipo de elemento reflectante de gran tamaño en un radio de 50 metros tomando como centro de la circunferencia la zona central de la pista. Puentes, rocas, edificios o cercas son algunos ejemplos de elementos reflectantes. Además, la pista debe estar en todo momento seca y libre de elementos absorbentes tales como nieve en polvo o materiales en forma de fragmentos.

El reglamento también destaca que cerca de los micrófonos no debe existir ningún tipo de elemento que pueda interferir en el campo acústico, ni ningún individuo que se encuentre entre el micrófono y la fuente sonora.

El lugar elegido para llevar a cabo las mediciones, fue la pista de ensayos de INTA en Madrid [XXVI]. Este emplazamiento cumple a la perfección las disposiciones anteriores.

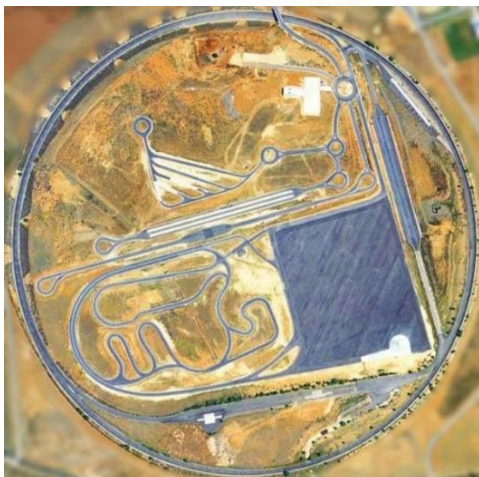


Figura 4.02. Vista en planta del INTA [XXVI].



Figura 4.03. Vista del lugar donde se realizaron las mediciones.

#### 4.2.2. CONDICIONES METEOROLÓGICAS

El objetivo de especificar en el reglamento las condiciones meteorológicas a las que deben realizarse los ensayos, es evitar la obtención de datos inusuales como consecuencia de unas condiciones no óptimas.

Para que se puedan llevar a cabo las mediciones en los ensayos, la temperatura ambiente debe situarse entre los 5°C y los 40°C.

Además, no se pueden realizar mediciones cuando la velocidad del viento, a la altura del micrófono incluyendo las rachas, sobrepasa los 5 m/s.

#### 4.2.3. POSICIÓN DE LOS MICRÓFONOS

Para obtener una visión más específica de la disposición de los dispositivos en el ensayo, conviene echar un vistazo a la Figura 4.03. Los micrófonos se deben disponer en la línea PP'. Desde la línea PP' a la línea CC', que es una línea perpendicular a PP' que sirve de referencia, debe existir una distancia de  $2,0 \text{ m} \pm 0,05 \text{ m}$ .

La altura a la que se deben disponer los micrófonos es de  $1,2 \text{ m} \pm 0,02 \text{ m}$  medido desde el suelo. La dirección de referencia debe ser horizontal, tal como se indica en CEI 61672-1:2013 [31], y además los micrófonos deben establecerse de forma perpendicular a la trayectoria del vehículo.

#### 4.2.4. CONDICIONES DE FUNCIONAMIENTO

Como dije antes, los ensayos se realizaron al aire libre. Estos se ejecutaron siguiendo las disposiciones que la normativa refleja en el ensayo en movimiento a velocidad constante en el apartado 3.3 del Anexo 3. Estos ensayos se basan en el desplazamiento hacia delante del vehículo, realizando pasadas con una velocidad constante a lo largo de todo el movimiento.

Para obtener una visión más exacta de cómo se realizó el ensayo, la Figura 4.03 refleja cómo se ejecuta el ensayo de una forma gráfica, además de designar las zonas que se definen en el mismo.

En su desplazamiento, el eje vehículo debe seguir una trayectoria lo más cercana posible a la línea CC' manteniendo una velocidad constante a lo largo de todo el recorrido. Para que la pasada sea válida, el plano delantero del vehículo debe atravesar la línea AA' cuando comience la medición. Al final de la misma, el plano trasero del vehículo debe cruzar la línea BB'. A lo

largo de toda la trayectoria del vehículo, este debe circular a una velocidad constante. Este ensayo se conoce también como ‘pass-by’.

La normativa establece que el ensayo debe hacerse a velocidades de 10 km/h y 20 km/h. Sin embargo, para obtener una mayor amplitud en los resultados, se concluyó que lo mejor era realizar mediciones con el vehículo circulando a velocidades del intervalo 5-30 km/h incrementando la velocidad de circulación con aumentos de 5 km/h. Por lo tanto, las velocidades a analizar serán 5 km/h (velocidad más baja), 10 km/h, 15 km/h, 20 km/h, 25 km/h y 30 km/h (velocidad más alta).

La tolerancia asignada para velocidades menores o iguales a 10 km/h es de  $\pm 2$  km/h, mientras que para velocidades superiores la tolerancia es de  $\pm 1$  km/h.

A su vez, se realizaron varias mediciones (entre 2 y 4 normalmente), para cada una de las velocidades a las que se circuló. El sentido de la marcha cambiaba en cada pasada, por lo que, si se realizaron dos tomas de datos, en una de ellas el sentido de circulación sería hacia la izquierda, mientras que el otro sería hacia la derecha. Esto es igualmente extrapolable cuando existen más casos, siguiendo obviamente la distribución anterior. Para las mediciones que se realizaban hacia la izquierda, se les asignaba el valor 1 o 3. En cambio, cuando el sentido de marcha era hacia la derecha, los números empleados eran el 2 y el 4.

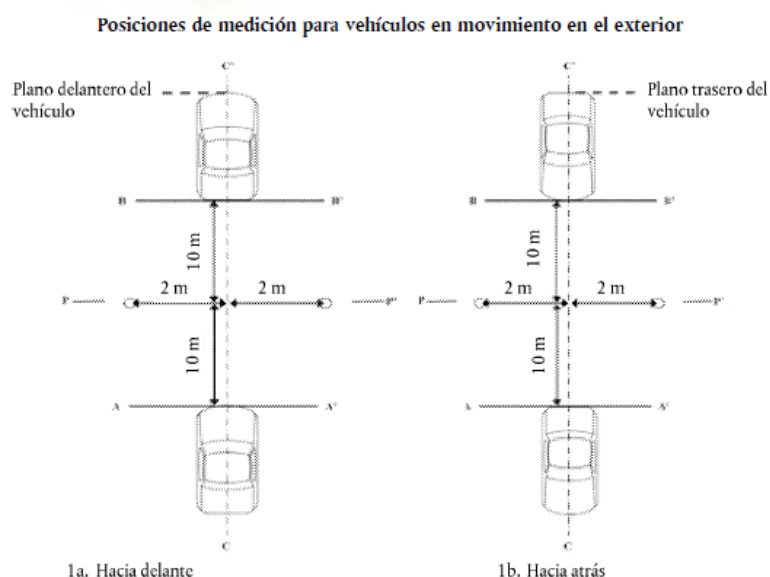


Figura 4.04. Esquema de la ejecución y disposición de los elementos del ensayo normalizado.

#### **4.2.5. OBTENCIÓN DE DATOS**

La explicación de la forma en la que los datos fueron obtenidos, se basará en una sola medición.

Cuando el vehículo realiza la pasada siguiendo la trayectoria de la línea CC', primeramente, el plano delantero del vehículo debe traspasar la línea AA' de la Figura 4.03. Es ahí cuando el array de micrófonos comienza a grabar la señal de audio que genera el sistema de advertencia del vehículo. El vehículo debe continuar en línea recta, siguiendo la trayectoria que el vehículo llevaba hasta el momento. En el instante en que el plano frontal del vehículo coincide con la línea PP', la cámara acústica toma una foto del vehículo, que posteriormente es empleada para realizar un mapa de presión sonora. El vehículo continúa circulando y el array recopilando datos, hasta que el plano trasero de este traspasa la línea BB', momento en el que el array deja de grabar.

### **4.3. DISPOSITIVOS EXPERIMENTALES**

Este apartado tiene como objetivo explicar los requisitos que debe cumplir la instrumentación que se debe emplear en los ensayos. Las disposiciones que se deben seguir, quedan establecidas en el Anexo 3 del Reglamento nº138 del CEPE [28]. A su vez, también se llevará a cabo la definición de los dispositivos utilizados en los ensayos, así como las características de los mismos.

#### **4.3.1. DISPOSITIVO DE ADQUISICIÓN DE DATOS SONOROS**

En primer lugar, se debe conocer que requisitos se establecen en cuanto al conjunto de dispositivos que se deben usar para realizar la toma de datos. Según la norma, para la medición de la presión sonora se debe usar un sonómetro o una alternativa equivalente, cumpliendo ambos casos los requisitos dispuestos para instrumentos de la clase 1 en la norma CEI 61672-1-2013 [31].

En los ensayos, se empleó una cámara acústica. La empresa suministradora de esta tecnología fue The Acoustic Camera. El modelo de array empleado fue la Ring48 AC Pro al que se le instaló una cámara Baumer [XXVII, XXVIII]. La cámara, a su vez, se conectó al sistema de adquisición de datos mcdRec [XXIX].

El modelo array de micrófonos empleado fue el tipo Ring48 AC Pro de la compañía The Acoustic Camera. Este sistema cuenta con 48 micrófonos distribuidos alrededor de una

circunferencia de 75 cm de diámetro. El rango acústico que maneja va desde 12 dB a 130 dB y el rango de frecuencias con las que puede trabajar de 164 Hz – 20 kHz. La sensibilidad que presenta este dispositivo es del orden de 20 mV/Pa.

La cámara que se instaló junto con el array, fue el modelo EXG50 de la marca Baumer. Esta cámara tiene una tasa de fotogramas de 13 fps y una resolución de  $2592 \times 1944$  px.

El sistema de adquisición de datos mcdRec presenta las siguientes características: almacenamiento de hasta 168 canales análogos recogiendo datos simultáneamente, capacidad de emplear frecuencias de muestreo entre 48 kHz y 192 kHz y una capacidad máxima de grabación de 349 s cuando se utiliza el límite inferior de las frecuencias de muestreo que es capaz de emplear el dispositivo y 87 s para el límite superior.

Además, el sistema de adquisición de datos mcdRec, permite la incorporación de distintos tipos de tarjetas de medida para realizar la adquisición de datos de distinta forma (en análogo o en digital). Además, el sistema puede presentar distintas tarjetas que presentan más o menos puntos de entradas de datos, siendo de 12 o 24 canales para mediciones en análogo y 12 en digital. En nuestro caso, se usaron dos tarjetas análogas de 24 entradas obteniendo así un punto de recolección de datos por cada micrófono del array.



Figura 4.05. Array tipo Ring 48 AC Pro [XXVII].



Figura 4.06. Cámara EXG50 de la marca Baumer [XXVIII].



Figura 4.07. Sistema de adquisición de datos mcdRec [XXIX].

Con estas características, el equipo que se va a emplear para la grabación, cumple con las especificaciones para instrumentos de la clase 1.

### 4.3.2. CALIBRACIÓN

El sistema de medición se debe comprobar por medio de un calibrador acústico que cumpla con las disposiciones para calibradores acústicos de clase 1 de la norma CEI 60942-2003 [32].

La calibración se debe realizar al principio y al final de cada jornada de ensayos. Los valores que indique el calibrador, deben presentar una diferencia igual o inferior a 0.5 dB(A). En caso contrario, las mediciones realizadas anteriormente del control, deberán omitirse.

La calibración de la cámara acústica no fue necesaria, por lo que no se usó calibrador.

### 4.3.3. INSTRUMENTACIÓN METEOROLÓGICA

Las condiciones meteorológicas por otro lado, se deben determinar con dispositivos que cumplan las siguientes condiciones de precisión:

- $\pm 1^{\circ}\text{C}$  para los aparatos encargados de medir la temperatura ambiente.
- $\pm 1\text{ m/s}$  para el dispositivo que mida la velocidad del viento en el emplazamiento.
- $\pm 5\text{ hPa}$  para el sistema que mida la presión del ambiente.
- $\pm 5\%$  en el dispositivo que tome datos de humedad relativa.

Los instrumentos deben tomar las medidas a una altura igual que la altura a la que se encuentren los micrófonos para obtener unos valores similares que a los que se producen alrededor del array. Las condiciones medidas con este, se deben extrapolar a toda el área de medición.

Para la medición de las condiciones meteorológicas, se empleó el anemómetro termohigrómetro PCE-THA 10 [XXX]. Este dispositivo presenta una estructura compacta y permite realizar la medición de distintos parámetros tales como la temperatura o la humedad. Los rangos de medición y las resoluciones del dispositivo son:

- Velocidad.
  - El rango de medición va de 0.4 m/s a 35 m/s con una tolerancia de  $\pm 3\%$ .
  - La precisión de las mediciones es de 0.1 m/s.

- Temperatura.
  - El rango de medición va de  $-15^{\circ}\text{C}$  a  $50^{\circ}\text{C}$  con error de  $\pm 0.6^{\circ}\text{C}$ .
  - La precisión es de  $0.1^{\circ}\text{C}$ .
- Humedad relativa.
  - El rango de medición va de 5% a 95% de H.R. con una tolerancia de  $\pm 3\%$  H.R. para la temperatura de  $25^{\circ}\text{C}$  y un intervalo de humedad del 30% al 95% H.R. y un error de  $\pm 5\%$  H.R. para  $25^{\circ}\text{C}$  para el intervalo de 10% a 30% H.R.
  - La precisión es de 0.1% H.R.



Figura 4.08. Anemómetro termohigrómetro PCE-THA 10 [XXX].

Las mediciones ambientales que se realizaron con el dispositivo se encontraban en todo momento dentro de los valores definidos en el Reglamento.

#### 4.3.4. CONDICIONES DEL VEHÍCULO

El vehículo que se a usar en los ensayos debe ser un vehículo cuyas características se presenten como una propuesta extrapolable a otros vehículos que vayan a comercializarse. Antes de proceder con la toma de datos, se debe hacer funcionar al vehículo en condiciones normales. Además, no se debe emplear ningún tipo de remolque en el vehículo.

Las baterías de propulsión del vehículo deben contar con la suficiente carga como para asegurar un funcionamiento correcto de todas las funciones imprescindibles. Estas quedarán

designadas por el fabricante. La temperatura a la que estas deben trabajar, debe encontrarse dentro de los marcos establecidos para permitir que las características instaladas funcionen.

Los ensayos han de llevarse a cabo con el vehículo circulando con la masa en orden de marcha, aceptando una tolerancia del 15%. Los neumáticos quedan asignados por el fabricante. Se debe asegurar que la presión de estos sea en todo momento la recomendada para la masa utilizada durante las mediciones.

El vehículo empleado para los ensayos fue el modelo Renault ZOE al que se le instaló un sistema de advertencia que poseía altavoces Keling A8 [XXXI, XXXII].

El vehículo presenta un motor eléctrico con una potencia máxima de 92 CV / 68 kW, con una capacidad útil de la batería de 41 kWh, una masa en orden de marcha de 1502 kg y unos neumáticos de dimensión 185/65 R15 92T.

Por otro lado, los altavoces Keling A8 presenta las siguientes características: respuesta de frecuencia de 20-20000 Hz, relación señal-ruido mayor de 80 dB, conexión a través de Bluetooth para distancias menores a 10 m y un peso de 0.17 kg.

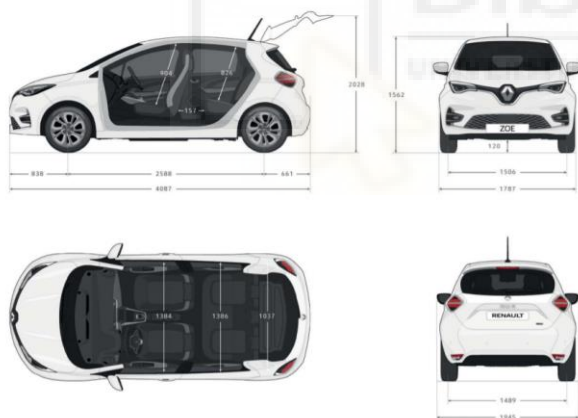


Figura 4.09. Plano con diferentes vistas del Renault ZOE [XXXI].



Figura 4.10. Altavoz Keling A8 [XXXII].

#### 4.4. PROCESAMIENTO DE DATOS

En este apartado, se describe la parte de este TFG correspondiente a la manipulación de los datos. Esto se puede dividir, a su vez, en dos tareas realizadas: obtener el nivel sonoro equivalente máximo y realizar un estudio de la directividad vertical.



#### 4.4.1. NIVEL SONORO CONTINUO EQUIVALENTE MÁXIMO

La primera tarea realizada consiste en obtener una serie de datos a partir de unas señales de audio recogidas a través de un array circular de micrófonos. El objetivo principal consiste en determinar el valor de nivel de presión sonora máximo medido en dB(A) que presenta cada una de las señales, así como el instante de tiempo en el que se produce dicho valor. En adición, se obtendrá el espectro de frecuencia referido al intervalo de tiempo resultante de tomar como extremos dicho punto y 0.125 segundos anteriores (es decir, el  $[t_i - 0.125, t_i]$ ).

Los recursos de partida fueron:

- Archivos con extensión .dat generados por medio del software de la cámara acústica donde se almacenaban los datos recogidos por ella.
- Archivos con extensión .m igualmente generados por el software de la cámara acústica en los que se definió en código Matlab, la extracción de los datos de los archivos de extensión .dat, la primera manipulación de los mismos y su representación.

Los ensayos se realizaron por medio de pruebas pass-by, en las que se realizaron mediciones en las que los vehículos recorrían distintas distancias (pasadas largas y cortas) con distintos sonidos de advertencia. A su vez, los sonidos se utilizaron a distintas velocidades (entre 5 y 30 km/h normalmente,), realizando distintas mediciones (entre 2 y 4, cambiando el sentido de la marcha del coche entre izquierda y derecha).

##### 4.4.1.1. PROGRAMA EXTRACTOR DE DATOS

Partimos de archivos de extensión .m. Estos scripts permiten obtener los datos de cada una de las señales de audio que se obtuvieron en los ensayos. Como ya se ha mencionado, el código fue creado a partir del software del array de micrófonos, que generaba un código que permitiera extraer los datos obtenidos por medio de Matlab. Al ejecutar dicho programa se obtenía una señal para cada uno de los micrófonos que se encontraban en el array. Estas señales mostraban la presión sonora en pascales (eje de ordenadas) en función del tiempo (eje de abscisas).

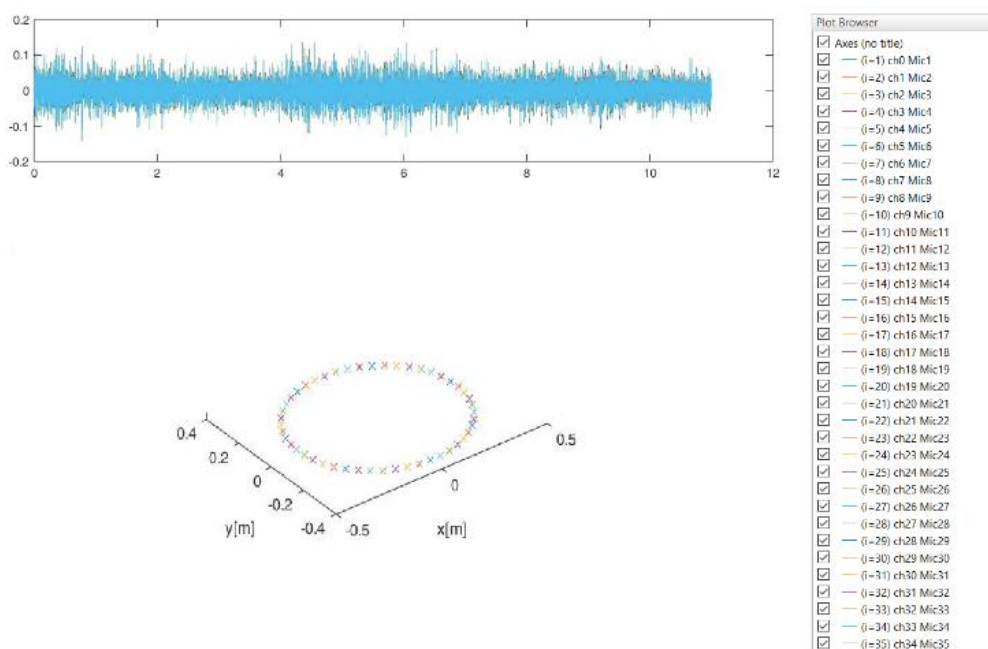


Figura 4.11. Resultados ofrecidos por los programas generados a través del software de la cámara.

En la Figura 4.11, el programa genera un gráfico en el que se representan cada una de las señales tomadas por cada micrófono. Existen 48 en total. Estas señales de cada gráfico de 48 señales proceden de la emisión de un mismo sonido de advertencia a una velocidad determinada, y en un sentido de marcha concreto desde el morro del vehículo. Por lo tanto, todas las señales son aproximadamente iguales.

De este modo, y con la finalidad de llevar a cabo un trabajo más rápido, se toma como señal de análisis aquella cuyo micrófono de recogida se encuentre coincidente con el eje de ordenadas. En este caso, decidí tomar como señal de trabajo aquella cuyo micrófono se corresponde con el número 1.

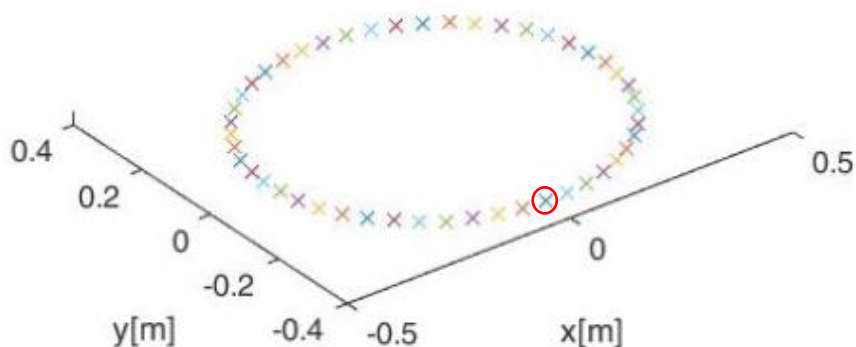


Figura 4.12. Designación del micrófono a emplear.

La tarea desempeñada con estos programas fue principalmente resolver qué hacía cada una de las partes del programa, y determinar qué variables almacenaban los datos.

<pre>%% Acoustic camera in MATLAB % Read binary channel data % % NI4 - 4.200 % % \$Date: Thu Dec 1 14:42:09 2011 % data-filename (e.g. filename = 'testmatlab.dat'); %filename = 'G:\Proyectos\cars_testexport\Photo\3Klick_2m\3Klick_2m_Channel View_2_1.dat';</pre>	INTRODUCCIÓN
<pre>curDirMFile = pwd filename = fullfile(curDirMFile, [filename '.dat'])</pre>	DEFINICIÓN ARCHIVO
<pre>% open/read file fid = fopen(filename, 'r');</pre>	APERTURA ARCHIVO
<pre>% read parameter nNbrOfChannels = fread(fid, 1, 'long') nNbrOfChannelsPlusPins = fread(fid, 1, 'long')  nNbrOfSamplesPerChannel = fread(fid, nNbrOfChannels, 'long') nNbrOfDigitalPinsPerChannel = fread(fid, nNbrOfChannels, 'long'); nChannelQuantization = fread(fid, nNbrOfChannels, 'long')  nTimeFactors = fread(fid, nNbrOfChannels, 'double'); nMapFactors = fread(fid, nNbrOfChannels, 'double');  % legend names ChannelNameLengths = fread(fid, nNbrOfChannelsPlusPins, 'long'); IsDigitalChannelName = fread(fid, nNbrOfChannelsPlusPins, 'long');  ChNames = cell(nNbrOfChannels,1); i_PlusPins = 0; nDigital = 0; nAnalog = 0;</pre>	LECTURA DE PARÁMETROS, DETERMINACIÓN DE VARIABLES Y LONGITUD

```

for i = 1:nNbrOfChannels
    % digital
    if nNbrOfDigitalPinsPerChannel(i) > 0
        for j = 1:nNbrOfDigitalPinsPerChannel(i)
            i_PlusPins = i_PlusPins + 1;
            nDigital = nDigital+1;
            ChNames{i}(j)(1:ChannelNameLengthes(i_PlusPins)) = char(fread(fid, ChannelNameLengthes(i_PlusPins), 'char'));
            ChannelNames(i_PlusPins,1:ChannelNameLengthes(i_PlusPins)) = ChNames{i}(j)(1:ChannelNameLengthes(i_PlusPins));
            DigitalChannelNames(nDigital,1:ChannelNameLengthes(i_PlusPins)) = ChNames{i}(j)(1:ChannelNameLengthes(i_PlusPins));
        end
    end
    % analog
    else
        i_PlusPins = i_PlusPins + 1;
        nAnalog = nAnalog+1;
        ChNames{i}(1:ChannelNameLengthes(i_PlusPins)) = char(fread(fid, ChannelNameLengthes(i_PlusPins), 'char'));
        ChannelNames(i_PlusPins,1:ChannelNameLengthes(i_PlusPins)) = ChNames{i}(1:ChannelNameLengthes(i_PlusPins));
        AnalogChannelNames(nAnalog,1:ChannelNameLengthes(i_PlusPins)) = ChNames{i}(1:ChannelNameLengthes(i_PlusPins));
    end
end
ChannelNames

% camera/sensor positions

% local array coordinate system
%
%      y
%      |      photo plane
%      mic      /|
%      |      / |
%      |      | |
%      z---[Cam]-----|---Object
%      /|      | |
%      / |      | /
%      / mic   | /
%      x |
%
%      array

% camera (cam) position vector in local array coordinate system (x,y,z), [m]
dCameraPositionVector = fread(fid, 3, 'double');
% camera (cam) view vector in local array coordinate system (x,y,z), [m]
dCameraViewVector = fread(fid, 3, 'double');
% camera (cam) up vector in local array coordinate system (x,y,z), [m]
dCameraUpVector = fread(fid, 3, 'double');
% camera opening angle, [Grad]
dCameraOpeningAngle = fread(fid, 1, 'double');
% Focus distance to measure object (Object), [m]
dFocus = fread(fid, 1, 'double')

% sensor (mic) positions in local array coordinate system (x,y,z), [m]
for i = 1:nNbrOfChannels
    dMicPosition(i,1:3) = zeros(1, 3, 'double');
end
for i = 1:nNbrOfChannels
    % (x,y,z) of each microfon
    dMicPosition(i,1:3) = fread(fid, 3, 'double');
end
%for i = 1:nNbrOfChannels
% ChannelNames(i,:)
% dSensorPosition = dMicPosition(i,1:3)
%end
%

```

```

'data reading ... please wait'

% read data/samples
nChannels = cell(nNbrOfChannels,1);
for i = 1:nNbrOfChannels
    % digital
    if nNbrOfDigitalPinsPerChannel(i) > 0
        for j = 1:nNbrOfDigitalPinsPerChannel(i)
            nChannels{i}(j) = {zeros(1,nNbrOfSamplesPerChannel(i), 'int0')};
        end
    % analog
    else
        % 'int16' data
        if nChannelQuantization(i) == 16
            nChannels{i} = {zeros(1,nNbrOfSamplesPerChannel(i), 'int16')};
        % 'int32' data
        elseif nChannelQuantization(i) == 32
            nChannels{i} = {zeros(1,nNbrOfSamplesPerChannel(i), 'int32')};
        end
    end
end
end

```

LECTURA Y  
ALMACENAMIENTO DATOS

```

for i = 1:nNbrOfChannels
    % digital
    if nNbrOfDigitalPinsPerChannel(i) > 0
        for j = 1:nNbrOfDigitalPinsPerChannel(i)
            nChannels{i}(j) = fread(fid, nNbrOfSamplesPerChannel(i), 'ubit1=>int8');
            fread(fid, 16-mod(nNbrOfSamplesPerChannel(i),16), 'ubit1=>int8');
        end
    % analog
    else
        % 'int16' data
        if nChannelQuantization(i) == 16
            nChannels{i} = fread(fid, nNbrOfSamplesPerChannel(i), 'int16');
        % 'int32' data
        elseif nChannelQuantization(i) == 32
            nChannels{i} = fread(fid, nNbrOfSamplesPerChannel(i), 'int32');
        end
    end
end
end
    
```

```

% close file
status = fclose(fid);
'... data read'
    
```

**CIERRE ARCHIVO EJECUTADO**

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plots / Analyses:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%return;

%simple data plot
%if MEMORY problems, plot less channels/smaller [FirstSample,LastSample]

% x-axis
for i = 1:nNbrOfChannels
    % analog
    if nNbrOfDigitalPinsPerChannel(i) == 0
        FirstSample(i) = 1;
        LastSample(i) = nNbrOfSamplesPerChannel(i);
    % digital
    else
        FirstSample(i) = 1;
        LastSample(i) = nNbrOfSamplesPerChannel(i);
        %LastSample(i) = nNbrOfSamplesPerChannel(i)/16;
    end
end
    
```

**DETERMINACIÓN  
PARÁMETROS DEL ARRAY  
DE MICRÓFONOS**

```

for i = 1:nNbrOfChannels
    % analog
    if nNbrOfDigitalPinsPerChannel(i) == 0
        xanalog(i) = nTimeFactors(i) * (FirstSample(i)-1>LastSample(i)-1);
        %xanalog = nTimeFactors(i) * (FirstSample(i)-1>LastSample(i)-1);
        %break;
    end
end
for i = 1:nNbrOfChannels
    % digital
    if nNbrOfDigitalPinsPerChannel(i) > 0
        xdigital = nTimeFactors(i) * (FirstSample(i)-1>LastSample(i)-1);
        break;
    end
end
    
```

**DETERMINACIÓN Y  
ALMACENAMIENTO DE  
DATOS DE SEÑALES**

```

% plot analog channels
'plotting ...'
subplot(3,1,1)
for i = 1:nNbrOfChannels
    nFactor = nMapFactors(i);
    if nNbrOfDigitalPinsPerChannel(i) == 0
        plot(xanalog(i), nFactor*nChannels{i}(FirstSample(i):LastSample(i)));
        %plot(xanalog, nFactor*nChannels{i}(FirstSample(i):LastSample(i)));
        %plot(nChannels{i}(FirstSample(i):LastSample(i)));
    end
    hold all;
end
% analog plot properties
if nAnalog > 0
    legend(AnalogChannelNames)
    if nAnalog > 5
        legend off
    end
end
end
    
```

**DIBUJO DE GRÁFICAS**

```

% plot digital channels
subplot(3,1,2)
for i = 1:nNbrOfChannels
    if nNbrOfDigitalPinsPerChannel(i) > 0
        for j = 1:nNbrOfDigitalPinsPerChannel(i)
            plot(xdigital, nChannels{i}{j}(FirstSample(i):LastSample(i)));
            %plot(nChannels{i}{j}(FirstSample(i):LastSample(i)));
            %stairs(xdigital, nChannels{i}{j}(FirstSample(i):LastSample(i)));
            hold all;
        end
    end
end
% digital plot properties
ylim([0 2])
if nDigital > 0
    legend(DigitalChannelNames)
    if nDigital > 5
        legend off
    end
end

% plot sensor positions
subplot(3,1,3)
for i = 1:nNbrOfChannels
    plot3(dMicPosition(i,1),dMicPosition(i,2),dMicPosition(i,3), 'x');
    hold all;
end
% properties
daspect('manual')
xlabel('x[m]')
ylabel('y[m]')
zlabel('z[m]')
legend(ChannelNames)
legend off

plotbrowser('on');

hold off;
    
```

Figura 4.13. Estructura y código del programa generada a través del software de la cámara acústica.

Descubriendo que las partes del programa donde se manejaban los datos relacionados con las señales recogidas por el array era la séptima y octava parte, realicé una modificación para obtener solamente la señal del micrófono que interesaba (Figura 4.14).

```

for i = 1:nNbrOfChannels
    % analog
    if nNbrOfDigitalPinsPerChannel(i) == 0
        FirstSample(i) = 1;
        LastSample(i) = nNbrOfSamplesPerChannel(i);
    % digital
    else
        FirstSample(i) = 1;
        LastSample(i) = nNbrOfSamplesPerChannel(i);
        %LastSample(i) = nNbrOfSamplesPerChannel(i)/16;
    end
end
for i = 1:nNbrOfChannels
    % analog
    if nNbrOfDigitalPinsPerChannel(i) == 0
        xanalog(i) = nTimeFactors(i) * (FirstSample(i)-1>LastSample(i)-1);
        %xanalog = nTimeFactors(i) * (FirstSample(i)-1>LastSample(i)-1);
        %break;
    end
end
for i = 1:nNbrOfChannels
    % digital
    if nNbrOfDigitalPinsPerChannel(i) > 0
        xdigital = nTimeFactors(i) * (FirstSample(i)-1>LastSample(i)-1);
        break;
    end
end

microphone=1;
for i = 1:microphone
    nFactor = nMapFactors(i);
    if nNbrOfDigitalPinsPerChannel(i) == 0
        data=nFactor*nChannels{i};
        %plot(xanalog, nFactor*nChannels{i}(FirstSample(i):LastSample(i)));
        %plot(nChannels{i}(FirstSample(i):LastSample(i)));
    end
end
'finished'
    
```

Figura 4.14. Modificación del programa generado por la cámara para obtener la señal del micrófono deseado.

Así, la modificación realizada se basa en la creación de una variable denominada como “microphone”, cuyo valor sea 1 (este número coincidía con el del micrófono del array a emplear). El objetivo era que, al realizar el bucle, “microphone” almacenara los datos en la variable “data” del micrófono seleccionado.

A su vez, si se decidiese cambiar el micrófono de trabajo, bastaría con cambiar el número de la variable “microphone” a aquel con el que se desee tomar dato. Los valores almacenados en la variable “data” consistían en datos de presión sonora en Pa de la señal distribuida a lo largo del tiempo de toma de datos para cada pasada en particular.

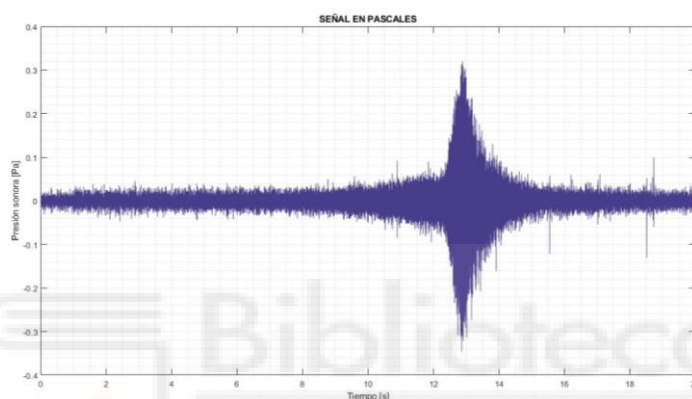


Figura 4.15. Señal en Pa obtenida con el programa modificado.

#### 4.4.1.2. PROGRAMA OPERACIONES

Obtenidos los datos de presión sonora de la señal a lo largo del tiempo, se debe llevar a cabo un análisis en profundidad de la información obtenida. Para ello, generé el programa “operaciones”, donde se realiza dicha tarea.

```
%% DEFINICIÓN DE VARIABLES.
fs=192000; %Frecuencia de muestreo.
dataA=filterA(data, fs); %Filtrado de la señal en A.
signalA=dataA.^2; %Elevamos la señal al cuadrado.
tau=0.125; %Intervalo de tiempo a aplicar el salto a nuestra señal.
p0=2*10^(-5); %Presión de referencia.
tf=length(data)/fs; %Cálculo del instante final de tiempo.

%% FILTRADO DE LA SEÑAL EN FAST.
num=1/(tau*fs); %Numerador del salto a aplicar.
den=[1 -exp(-1/(tau*fs))]; %Denominador del salto a aplicar.
signalfA=filter(num,den,signalA); %Filtrado fast de la señal con poneración A.
```

Figura 4.16. Primera parte del programa operaciones.

En esta primera parte, se definen todos aquellos parámetros necesarios para trabajar con los datos (conversiones, cálculo de otras variables, representación...). Los parámetros definidos son:

- La frecuencia de muestreo ( $f_s$ ),
- La señal de presión sonora en pascales con ponderación A. Para conseguirlo, se emplea un filtro creado por Zhivomirov [XXXIII]. Este filtro permite, a partir de los datos de presión sonora en pascales y la frecuencia de muestreo, ponderar dichos datos a frecuencia A.
- La señal ponderada anteriormente obtenida elevada al cuadrado.
- El coeficiente de ponderación temporal Fast.
- La presión sonora de referencia.
- El tiempo total de muestreo, que para este caso es igual que el último instante de tiempo en el que se tomaron los datos. Esto se puede determinar a través de la expresión:

$$\text{Tiempototal} = \frac{\text{Puntos tomados}}{\text{Puntos tomados/Tiempo}} = \text{Tiempo} \quad (10)$$

El segundo bloque se encarga de llevar a cabo el filtrado de la señal a través de una ponderación temporal de tipo Fast. Para realizar esto, se emplea la ecuación (3) con una constante nominal de 0.125. Sin embargo, Matlab dispone de otra forma de proceder.

Para obtener una expresión que Matlab sea capaz de reconocer, primero se adopta una variable  $y^2$  y se realiza la convolución de esta variable elevada al cuadrado con un salto de  $(1/\tau)e^{(-t/\tau)}$ . Por lo tanto, la expresión quedaría como

$$L_t = 10 \log_{10} \left[ \frac{h(y^2)}{P_0^2} \right] \quad (11)$$

Donde:

- $y$  es la presión sonora en Pa ponderada en A.
- $h(y^2)$  es la convolución de  $y^2$  con el salto especificado anteriormente.
- $P_0$  es la presión de referencia.



Esta sería otra forma de realizar la ponderación temporal. No obstante, esto puede realizarse de otra forma, que permite obtener resultados de una manera más automática.

El salto empleado anteriormente puede pasarse a dominio de Laplace, obteniendo como resultado  $H(s) = \frac{1/\tau}{s+1/\tau}$ . Seguidamente, usando el método de invarianza de impulso, se genera un filtro de carácter discreto útil para realizar la ponderación. Dicho filtro utilizará el siguiente salto, hallado en el dominio Z:

$$H(z) = \frac{1/(\tau \cdot fs)}{(1 - e^{-1/(\tau \cdot fs)})z^{-1}} \quad (12)$$

Donde:

- $\tau$  es la constante nominal de la ponderación temporal a emplear.
- $fs$  es la frecuencia de muestreo.

El filtro se aplicará a cada uno de los datos del vector de presión sonora ponderando así temporalmente. Esta es la forma óptima y rápida de realizar la conversión, puesto que los datos con los que estamos trabajando son discretos. Por lo tanto, solamente se aplica el salto a cada punto. Sabiendo cómo son los parámetros empleados, queda definir la forma en que Matlab pueda reconocerlo. Para ello, se definen cada uno de los coeficientes que conforman la expresión en dominio Z, establecida anteriormente.

	EXPRESIÓN	MATLAB
NUMERADOR	$\frac{1}{\tau \cdot fs}$	<code>num=1/(tau*fs)</code>
DENOMINADOR	$(1 - e^{-1/(\tau \cdot fs)})z^{-1}$	<code>den=[1 -exp(-1/(tau*fs))]</code>
RESULTADO	$H(z) = \frac{1/(\tau \cdot fs)}{(1 - e^{-1/(\tau \cdot fs)})z^{-1}}$	<code>filter(num,den,signalA)</code>

Tabla 4.01. Programación del filtro temporal en Matlab.

Para la definición del numerador en Matlab, simplemente se traduce la distribución de los parámetros de la expresión en función de las variables creadas anteriormente. El denominador se determina mediante la toma de elementos dentro del paréntesis, que están multiplicados por  $z^{-1}$  y siendo asignados a la variable “den”. Por último, se designa el filtro a emplear, realizándose por medio del comando “filter” tal y como queda reflejado en la tabla.

Hasta ahora, se tiene la señal en pascales con ponderación frecuencial A y con una ponderación temporal Fast. A pesar de poseer los datos ponderados de la forma deseada, hace falta realizar una última conversión para obtener los datos en dB(A). Para ello, se emplea la ecuación (1).

Pero, ¿por qué se debe realizar una ponderación frecuencial A y ponderación temporal Fast con los datos? Porque la normativa lo indica de este modo. El punto 1.1.1. del Anexo 3 del Reglamento nº138 del CEPE [28], indica que las mediciones se realizarán empleando una ponderación temporal ‘F’ (Fast) y una ponderación frecuencial ‘A’. Esto queda detallado en la norma CEI 61672-1-2013 [31].

#### **%% TRANSFORMACIÓN DE LA SEÑAL A dB(A) .**

```
signaldba=10*log10(signalfa/p0^2); %Transformación de la señal en A a dB(A) .
```

Figura 4.17. Conversión de  $Pa^2(A)$  a dB(A) en el programa operaciones para el cálculo del nivel sonoro equivalente máximo.

A pesar de que la representación de la información es posible, existen ensayos donde las señales presentaban datos anómalos y por ello falsificaban los resultados obtenidos. Para ello, está justificado el desarrollo del programa “anómalos”, ejecutado en la siguiente parte del programa.

Una vez se ha determinado si nuestra señal presenta datos anómalos, y determinado el verdadero máximo de la señal, se realizaron distintas representaciones con los datos. Las representaciones empleadas fueron:

- La señal de presión sonora en Pa a lo largo del tiempo.
- La señal de presión sonora en  $Pa^2$  con ponderación frecuencial A y temporal Fast.
- La señal de presión sonora en dB(A) y ponderación temporal Fast.

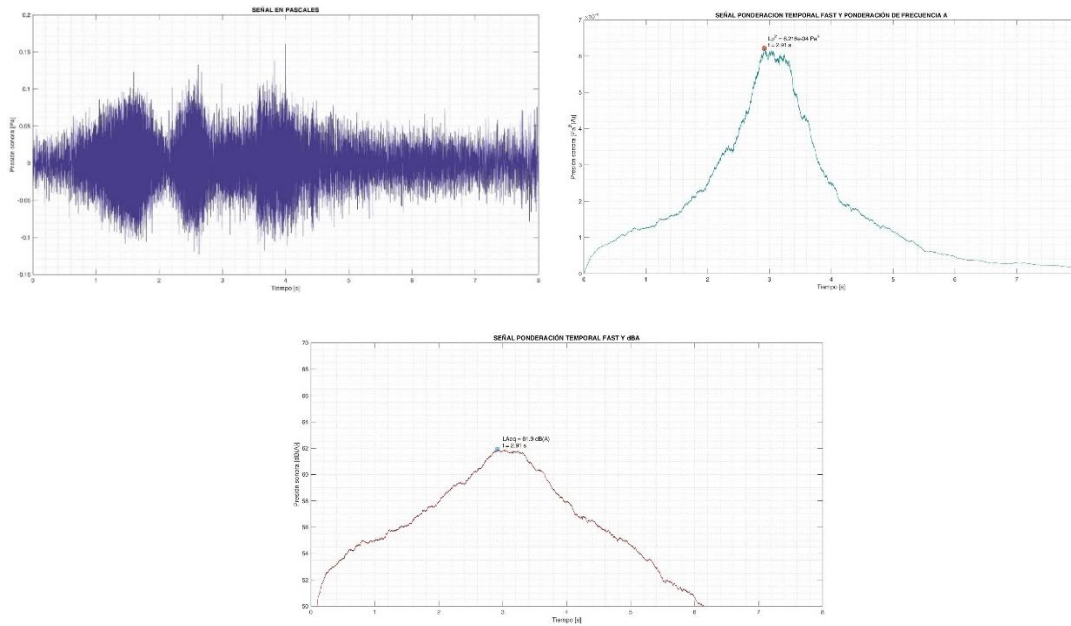


Figura 4.18. Representaciones obtenidas a través de operaciones.

```

%% REPRESENTACIÓN DE LOS DATOS.
% GRÁFICA SIN PONDERACIÓN EN PA:
figure %Representación del gráfico en una ventana independiente.
plot(time,data,'color',[72,61,139]/255) %Representación de los datos de la señal en pascales sin ninguna ponderación.
xlim([0 tf]) %Determinación de los límites del eje X para la gráfica de la señal en pascales sin ninguna ponderación.
grid minor %Colocación cuadrícula en la gráfica.
xlabel('Tiempo [s]'); %Colocación de etiqueta en el eje X para la gráfica de la señal en pascales sin ninguna ponderación.
ylabel('Presión sonora [Pa]'); %Colocación de etiqueta en el eje Y para la gráfica de la señal en pascales sin ninguna ponderación.
title('SEÑAL EN PASCALES') %Colocación de título para la gráfica de la señal en pascales sin ninguna ponderación.

% GRÁFICA CON PONDERACIÓN TEMPORAL FAST Y PONDERACIÓN DE FRECUENCIA A:
figure %Representación del gráfico en una ventana independiente.
plot(time,sigalFA,'o','MarkerIndices',posmaxFA,'MarkerSize',8,'MarkerEdgeColor',[139,0,0]/255,'MarkerFaceColor',[255,99,71]/255) %Colocación de marcador
%en punto máximo para gráfico de la señal con ponderación temporal fast y ponderación de frecuencia A.
hold on %Mantenimiento del marcador en la ventana para realizar más operaciones sin machacar lo anteriormente realizado.
plot(time,sigalFA,'color',[0,128,128]/255) %Representación de los datos de la señal con ponderación temporal fast y ponderación de frecuencia A.
hold off %Eliminación del mantenimiento para realizar operaciones.

xlim([0 tf]) %Definición de los límites del eje X para la señal ponderada en A y fast.
ylim([0 3.2*10^-3]) %Definición de los límites del eje Y para el gráfico de la señal ponderada en A y fast.
grid minor %Colocación cuadrícula en la gráfica.
xlabel('Tiempo [s]'); %Colocación de etiqueta del eje X para el gráfico de la señal ponderada en A y fast.
ylabel('Presión sonora [Pa^2(A)]'); %Colocación de etiqueta del eje Y para el gráfico de la señal ponderada en A y fast.
title('SEÑAL PONDERACIÓN TEMPORAL FAST Y PONDERACIÓN DE FRECUENCIA A') %Colocación de título para el gráfico de la señal ponderada en A y fast.

sac1=num2str(maxFA,'10.3e'); %Transformación de datos numéricos del máximo valor numérico de la señal ponderada en A y fast a array de caracteres.
sac2=num2str(tmaxFA,4); %Transformación de datos numéricos de tiempo de ese máximo valor de la señal ponderada a array de caracteres.
txt1=['lg^2 = ',sac1,' Pa^2']; %Definición del texto para marcador estableciendo valor en X.
text01=text(tmaxFA+0.07,tmaxFA+0.03*10^-3,txt1); %Colocación de texto en marcador del máximo valor de la señal ponderada en A y fast.
txt2=['t = ',sac2,' s']; %Definición del texto para marcador estableciendo valor en Y.
text02=text(tmaxFA+0.07,tmaxFA+0.01*10^-3,txt2); %Colocación de texto en marcador del valor de tiempo del máximo valor de la señal ponderada en A y fast.

% GRÁFICA CON PONDERACIÓN TEMPORAL FAST Y dB(A):
figure %Representación del gráfico en una ventana independiente.
plot(time,sigalDBA,'o','MarkerIndices',posdBAmax,'MarkerSize',8,'MarkerEdgeColor',[70,130,180]/255,'MarkerFaceColor',[135,206,235]/255) %Colocación de marcador
%en punto máximo para gráfico de la señal con ponderación temporal fast y dB(A).
hold on %Mantenimiento del marcador en la ventana para realizar más operaciones sin machacar lo anteriormente realizado.
plot(time,sigalDBA,'color',[139,0,0]/255) %Representación de los datos de la señal con ponderación temporal fast y dB(A).
hold off %Eliminación del mantenimiento para realizar operaciones.

xlim([0 tf]) %Definición de los límites del eje X para la señal ponderada en fast y dB(A).
ylim([50 70]) %Definición de los límites del eje Y para la señal ponderada en fast y dB(A).
grid minor %Colocación cuadrícula en la gráfica.
xlabel('Tiempo [s]'); %Colocación de etiqueta del eje X para el gráfico de la señal ponderada en fast y dB(A).
ylabel('Presión sonora [dB(A)]'); %Colocación de etiqueta del eje Y para el gráfico de la señal ponderada en fast y dB(A).
title('SEÑAL PONDERACIÓN TEMPORAL FAST Y DBA') %Colocación de título para el gráfico de la señal ponderada en fast y dB(A).

sac3=num2str(dBAmax,3); %Transformación de datos numéricos del máximo valor numérico de la señal ponderada en fast y dB(A) a array de caracteres.
sac4=num2str(tmaxDBA,4); %Transformación de datos numéricos de tiempo de ese máximo valor de la señal ponderada a array de caracteres.
txt3=['Leq = ',sac3,' dB(A)']; %Definición del texto para marcador estableciendo valor en X.
text03=text(tmaxDBA+0.08,dBAmax+0.8,txt3); %Colocación de texto en marcador del máximo valor de la señal ponderada en fast y dB(A).
txt4=['t = ',sac4,' s']; %Definición del texto para marcador estableciendo valor en Y.
text04=text(tmaxDBA+0.08,dBAmax+0.3,txt4); %Colocación de texto en marcador del máximo valor de la señal ponderada en fast y dB(A).
    
```

Figura 4.19. Parte que representa los distintos datos obtenidos en el programa operaciones.

Para la representación de los datos se hace necesario algunas variables, como “time” (que almacena el tiempo) o “dBAm<sub>max</sub>” (donde se almacena el valor máximo de dB(A)), que aparecerán en programas posteriores. A la hora de generar los códigos para cada gráfica, todas presentan una estructura similar en cuanto al código empleado, sobre todo la segunda y tercera.

Para realizar una representación completa, se define que en una ventana nueva (“figure”), se grafiquen los datos definidos en “plot”. La primera variable dentro del “plot” define los datos del eje X, mientras que otra define los del eje Y. A su vez, en el “plot”, también se definen ciertas características de la gráfica, como su color. Posteriormente se determinan algunas características de la ventana, como el límites o etiquetas de ejes, entre otros.

Al ejecutar el código, aparecen los gráficos con los datos representados. No obstante, para hacer más visibles los datos buscados, se estableció, en el segundo y tercer gráfico, un marcador en forma de círculo que designase los puntos máximos en dichas gráficas. Para ello, se dispuso un “plot” que se encarga de dibujar el marcador en la posición máxima. Tras esto, se emplea el comando “hold on” que se encarga de mantener la ventana intacta, sin modificar lo anteriormente realizado. Seguidamente se establece el “plot” que representa los datos en la ventana. Finalmente, se emplea un “hold off” para quitar ese bloqueo de ventana. De esta forma se consigue obtener la gráfica representada con su marcador correspondiente.

Lo siguiente a realizar es la colocación del texto en el marcador. Lo primero es convertir los valores numéricos a cadena de caracteres, ya que los marcadores solamente permiten datos en forma de array de caracteres. Para ello, se emplea el comando “num2str”. Seguidamente, se definen los textos a disponer en el marcador, guardándolo en el conjunto de variables “txt(n)”. Existe una variable por cada línea de texto a disponer en los marcadores (dos para el segundo gráfico y otras dos para el tercero). También se debe definir la posición que va a ocupar en el marcador la línea de texto por medio de “text”.

La última línea de este programa lleva a cabo la ejecución de “espfrec”, encargado de definir el espectro de frecuencia para el instante donde se produce el máximo valor de cada gráfica, así como el instante a 0.125 segundos por debajo de este.

#### **4.4.1.3. PROGRAMA ANÓMALOS**

Al ejecutar el programa operaciones, se han ejecutado distintos programas que realizaban otras tareas. El primero es “anómalos”. En este programa se aplica el criterio de Chauvenet, método que permite discernir si un dato es atípico a través de cálculos estadísticos [33]. Su aplicación puede resumirse en 5 partes:

- 1- Se calcula la media de los datos a analizar.
- 2- Se calcula la desviación estándar de la información.
- 3- Se toma el punto a analizar y se calcula la distancia a la media.
- 4- Se divide dicha distancia entre la desviación estándar obteniendo así un coeficiente.
- 5- Se compara dicho coeficiente con los coeficientes establecidos por Chauvenet (si nuestro coeficiente es mayor que el de Chauvenet el dato será anómalo. En caso contrario, no lo será).

En primer lugar, se lleva a cabo la definición de los parámetros necesarios para precisar la existencia de anómalos. Cabe destacar que el programa trabaja con las señales en  $\text{Pa}^2(\text{A})$  con ponderación Fast, aunque también hubiera servido trabajar con la señal en  $\text{dB}(\text{A})$ .

```
%% DEFINICIÓN DE VARIABLES
chauenetmuestras=[5*10^5; 5*10^6; 5*10^7]; %Número de muestras para coeficientes de Chauvenet.
chauenetcoef=[4.89; 5.33; 5.73]; %Coeficientes de chauvenet en funcion de número de muestras.
longsignalfa=length(signalfa); %Cálculo de la longitud de la señal para interpolar.
chavuse=interp1(chauenetmuestras,chauenetcoef,longsignalfa); %Interpolación para sacar el coeficiente de Chauvenet que necesitamos.
med=mean(signalfa); %Cálculo de la media.
desvest=std(signalfa); %Cálculo de la desviación estándar.
signalfaa=signalfa; %Definición de una nueva variable de señal en fast para no machacar la señal (si no, el ploteo no se haria bien).
signalfaamax=max(signalfa); %Cálculo del valor máximo de la señal en fast.
distmed=abs(signalfaamax-med); %Cálculo de la distancia a la media del valor máximo de la señal en fast.
coeficiente=distmed/desvest; %Cálculo del coeficiente a comparar con Chauvenet.
```

Figura 4.20. Definición de variables en el programa anómalos.

Las primeras dos líneas definen los coeficientes de Chauvenet para ciertas cantidades de muestras (se toman 3 coeficientes relacionados con los mayores números de muestras proporcionados por el manual(referencia)). La cantidad de datos que maneja cada señal tomada es elevada.

El código determina el número total de datos que conforman la señal por medio del comando “length”, que mide la longitud del vector de la señal. A continuación, se determina el coeficiente de Chauvenet llevando a cabo una interpolación lineal, con el comando ”interp1”.

Las siguientes líneas desarrollan la parte estadística del método. Se calcula la media con “mean” y la desviación estándar con “std”. Se almacena la señal ponderada temporalmente en otra variable para evitar que se produzcan interferencias, o se sobrescriba la señal obtenida anteriormente en el programa “operaciones”.

A continuación, se determina el máximo de la señal y se calcula la distancia existente entre dicho valor y la media (con el comando “abs” se obtiene su valor absoluto). Por último, se calcula

el coeficiente que se comparará con el coeficiente de Chauvenet, que se obtiene dividiendo la distancia media entre la desviación estándar.

Con los datos ya obtenidos se comienza a analizar, por medio de Chauvenet, si existen datos inusuales en las señales. Ahora corresponde comparar los coeficientes para determinar si existen anómalos y realizar las operaciones pertinentes: omitir el anómalo y buscar el máximo verdadero, o mantener y trabajar con el máximo definido en un principio.

```
%% DETERMINACIÓN DE PRESENCIA ANÓMALOS
if coeficiente>chauvuse %Primera condición donde se definen operaciones en caso de que coeficiente calculado sea mayor que el coeficiente de Chauvenet.

    display('Toma de datos presenta datos anómalos') %Escribir por pantalla la presencia de datos anómalos.

    while coeficiente>chauvuse %Bucle que repite la detección y eliminación de anómalos.

        postimax=find(signalFA==signalFAmax); %Determinación del punto en el que se encuentra el máximo.
        timax=time(postimax); %Determinación del tiempo donde se da dicho máximo.
        timenor=timax-0.05; %Parte de la señal por detrás a quitar.
        postimenor=find(timenor==time); %Posición del punto de la señal por detrás a quitar.
        timayor=timax+0.125; %Parte de la señal por delante a quitar.
        postimayor=find(timayor==time); %Posición del punto de la señal por delante a quitar.
        vectpos=[postimenor:1:postimayor]; %Vector de posiciones a quitar.
        signalFA(vectpos)=med; %Transformación de valores de posiciones a quitar de la señal fast al valor de media.
        signalFAmax=max(signalFA); %Cálculo del valor máximo de la señal en fast.
        distmed=abs(signalFAmax-med); %Cálculo de la distancia a la media del valor máximo de la señal en fast.
        coeficiente=distmed/desvest; %Cálculo del coeficiente a comparar con Chauvenet.

    end %Cierre de bucle.

    signaldBAa=10*log10(signalFAa/p0^2); %Cálculo de la señal modificada en dB(A).

    maxfA=max(signalFA); %Obtención del valor máximo de la señal fast modificada.
    posmaxfA=find(signalFA==maxfA); %Posición en la que se encuentra ese máximo en la señal sin modificar.
    tmaxfA=time(posmaxfA); %Obtención del tiempo para ese instante.
    tmaxfAr=round(tmaxfA,2); %Redondeo a 2 decimales el instante de tiempo donde se da el valor fast máximo.

    dBAmA=max(signaldBAa); %Obtención del valor dB(A) máximo en la señal modificada.
    posdBAmA=find(signaldBA==dBAmA); %Posición en la que se encuentra dicho dB(A) máximo en la señal sin modificar.
    tmaxdBAtime=time(posdBAmA); %Obtención del valor de tiempo para ese dB(A) máximo.
    dBAmAr=round(dBAmA,1); %Redondeo del valor dB(A) máximo a 1 decimal.
    tmaxdBAr=round(tmaxdBAtime,2); %Redondeo del instante de tiempo donde se da ese valor máximo a 2 decimales.

else %Segunda condición donde se definen operaciones en caso de que coeficiente calculado sea menor que el coeficiente de Chauvenet.

    maxfA=max(signalFA); %Obtención del valor máximo de la señal fast sin modificar.
    posmaxfA=find(signalFA==maxfA); %Posición en la que se encuentra ese máximo en la señal sin modificar.
    tmaxfA=time(posmaxfA); %Obtención del tiempo para ese instante.
    tmaxfAr=round(tmaxfA,2); %Redondeo a 2 decimales el instante de tiempo donde se da el valor fast máximo.

    dBAmA=max(signaldBA); %Obtención del valor dB(A) máximo.
    posdBAmA=find(signaldBA==dBAmA); %Posición en la que se encuentra dicho dB(A) máximo.
    tmaxdBAtime=time(posdBAmA); %Obtención del valor de tiempo para ese dB(A) máximo.
    dBAmAr=round(dBAmA,1); %Redondeo del valor dB(A) máximo a 1 decimal.
    tmaxdBAr=round(tmaxdBAtime,2); %Redondeo del instante de tiempo donde se da ese valor máximo a 2 decimales.

end %Cierre de condiciones.
```

Figura 4.21. Funciones a realizar según la existencia de anómalos o no en el programa anomalos.

Así, en el código se definen dos condiciones: una condición cuyo coeficiente calculado sea mayor que Chauvenet y otra para el caso contrario.

Cuando se detecte un dato anómalo, el proceso a seguir será:

- 1- Determinar la posición del dato inusual (que coincidirá en ese instante con el máximo).
- 2- Escoger cierto rango por delante y por detrás de dicho punto de la señal.

- 3- Cambiar los valores de las posiciones del intervalo seleccionado a valor de media.
- 4- Recalcular el coeficiente a comparar con Chauvenet y realizar la comparación de nuevo, por si hubiera aún anómalos.

Cuando se verifica que el caso estudiado cumple la primera condición, se informa de la presencia de datos anómalos con un “display”. Seguidamente, se presenta un bucle “while” que hará las operaciones designadas en su interior hasta que encuentre un punto no anómalo.

Dentro del bucle, las seis primeras líneas definen qué trozo de señal se adopta para eliminar el anómalo. Para ello, se determina la posición del valor máximo antes de modificar la señal (almacenada anteriormente en “signalAa” para poder manipularla sin que repercuta en la posteridad). Se escoge un punto a 0.05 segundos por detrás de ese máximo y se busca su posición en la señal. Igualmente, se elige otro punto a 0.125 segundos por delante, y se busca de nuevo en la señal.

Más adelante, se genera un vector que define todos los puntos existentes entre los puntos inferior y superior obtenidos anteriormente. Este vector de posiciones es posteriormente utilizado para transformar los valores de los puntos en la señal a valor de media (eliminando el anómalo y evitando que aparezca en el siguiente estudio).

Tras esto, se vuelve a buscar el valor del nuevo máximo de la señal con “max”, determinando su distancia hasta la media y calculando de nuevo el coeficiente a comparar con el coeficiente de Chauvenet correspondiente (repitiendo el proceso hasta que el máximo detectado no sea dato inusual).

En los casos sin datos anómalos, se entraría en el “else”, evitando todo lo anteriormente explicado. Para las mediciones con anómalos, los valores de la señal modificada en  $Pa^2(A)$  son transformados a  $dB(A)$ . En los casos sin anómalos, este paso no es necesario, ya que la señal obtenida en “operaciones” está en  $dB(A)$ . Además, en las mediciones sin anómalos, no se exige el uso de la variable “signalAa”. Esto se debe a que no se hace falta modificar la señal original.

La continuación del programa se basa en unas líneas comunes al procesamiento de señales con o sin anómalos. Aquí se obtiene información de la señal en  $Pa^2(A)$  y ponderación Fast, y de la señal en  $dB(A)$  y ponderación Fast. En primer lugar, se determina el máximo valor de la señal correspondiente y se obtiene su posición en el vector de datos de la señal. En segundo lugar, se define el valor del instante de tiempo en ese máximo y su posición en el vector de datos de tiempo. Por último, se redondean los valores obtenidos a los decimales definidos según convenga. En este TFG, se ha adoptado 2 decimales para los valores de tiempo, 2 decimales para el máximo de la

señal en Pa<sup>2</sup>(A) y ponderación temporal Fast y 1 decimal para el máximo de la señal en dB(A) y ponderación temporal Fast.

La mayoría de las variables obtenidas anteriormente sirven para definir el marcador de los gráficos, así como su posición en él.

#### 4.4.1.4. PROGRAMA ESPFREC

El programa ejecutado al final de “operaciones” es “espfreq”, que se encarga de generar el espectro de frecuencia en tercios de octava. En primer lugar, se definen una serie variables para poder llevar a cabo la generación del espectro de frecuencia buscado.

```
%% DEFINICIÓN DE VARIABLES A EMPLEAR.  
timemenor=tmaxdBA-0.125; %Cálculo del menor valor del intervalo de tiempo a usar para determinar el espectro de frecuencia.  
postmenor=find(time==timemenor); %Designación de la posición de dicho instante en el vector de tiempo.  
samplesuse=[postmenor:1:posdBAmx]; %Definición del vector con las posiciones a emplear del vector de presión sonora en Pa.  
samplesPa=data(samplesuse); %Determinación del vector presión sonora en Pa a utilizar.
```

Figura 4.22. Definición de variables iniciales en el programa espfreq.

De esta manera, se determina el instante de tiempo a 0.125 segundos por debajo del momento donde se produce el máximo valor en la señal. Se establece qué posición ocupa dicho instante en el vector que almacena el tiempo. Esto se realiza por medio de “find”, asignando la condición de que el momento definido anteriormente y el del vector tiempo sean iguales.

Conocida la posición de dicho momento y la posición del instante del máximo valor, se crea un vector donde se designan las posiciones del vector tiempo existentes entre los dos puntos anteriores. Este vector será empleado para obtener otro vector con valores de presión sonora en Pa, sin ponderar.

Ya se han obtenido algunos parámetros necesarios para la obtención del espectro de frecuencia. El siguiente paso es procesar los datos en la forma correcta para que el filtro y las funciones adicionales de las bibliotecas de Matlab puedan trabajarlos. El conjunto de programas se recoge dentro del paquete “Octave”, creado por Couvreur [XXXIV].

```
%% OBTENCIÓN DE LOS VALORES EN dB(A)  
[B,A] = adsgn(fs); % Configura el filtro en dB(A).  
samplesPafiltr = filter(B,A,samplesPa); % Filtra la señal acústica.  
[P,F] = filtbank(samplesPafiltr,fs,[]); % En P calcula el nivel de presión de las bandas F.  
vectrfrec=1:length(F); % Vector de apoyo para hacer equidistante los valores del eje x.
```

Figura 4.23. Obtención de parámetros para generar el espectro de frecuencias en el programa espfreq.



Este bloque primero llama a “adsgn”, que permite obtener los parámetros necesarios (numerador y denominador) para generar el filtro digital, a través de la frecuencia de muestreo. Se filtra el vector de datos de presión sonora en Pa, hallando un parámetro cuya función es obtener las variables necesarias para definir el espectro de frecuencia.

La ejecución de “filtbank” permite obtener las variables “P” y “F”, a través de la señal filtrada, la frecuencia de muestreo y el tiempo. “P” se refiere al vector fila con los niveles de presión sonora para cada una de las bandas de 1/3 de octava, expresado en dB(A). “F”, por otro lado, almacena la frecuencia de cada una de las bandas a emplear. Finalmente, se define un vector que almacene el número total de bandas.

Tras obtener los parámetros necesarios del espectro de frecuencia en tercios de octava, se procede a su representación. Para ello, se utilizarán los niveles mínimos de presión sonora, en dB(A), proporcionados por el Reglamento nº138 del CEPE [28].

```
%% GRÁFICO DEL ESPECTRO DE FRECUENCIA EN TERCIOS DE OCTAVA.
figure % Representación del gráfico en una ventana independiente.
bar(vectfrec,P,'FaceColor',[0 .5 .5],'EdgeColor',[0 .9 .9], 'LineWidth', 1.5) % Plotea con barras el nivel de presión.
hold on %Mantenimiento de la gráfica anterior en la ventana para realizar más operaciones sin machacar lo anteriormente realizado.

if (v==1) %Primera condición donde se definen operaciones para una velocidad de 5 km/h.

    nivnorm=[40 40 40 39 38 39 40 40 41 41 41 41 39 37 34 31 29 26]; %Definición de los niveles de presión sonora mínimos en dB(A) según normativa para 5 km/h.
    plot(vectfrec,nivnorm,'-ro','LineWidth',2,'MarkerSize',7,'MarkerFaceColor',[1 .6 .6],'MarkerEdgeColor','r') %Representación de los niveles anteriormente definidos.

else if (v==2) %Segunda condición donde se definen operaciones para una velocidad de 10 km/h.

    nivnorm=[45 45 45 44 43 44 45 46 46 46 44 44 42 39 36 34 31]; %Definición de los niveles de presión sonora mínimos en dB(A) según normativa para 10 km/h.
    plot(vectfrec,nivnorm,'-ro','LineWidth',2,'MarkerSize',7,'MarkerFaceColor',[1 .6 .6],'MarkerEdgeColor','r') %Representación de los niveles anteriormente definidos.

else if (v==3) %Tercera condición donde se definen operaciones para una velocidad de 15 km/h.

    nivnorm=[48 48 48 47 46 47 48 48 49 49 49 49 47 45 42 39 37 34]; %Definición de los niveles de presión sonora mínimos en dB(A) según normativa para 15 km/h.
    plot(vectfrec,nivnorm,'-ro','LineWidth',2,'MarkerSize',7,'MarkerFaceColor',[1 .6 .6],'MarkerEdgeColor','r') %Representación de los niveles anteriormente definidos.

else if (v==4) %Cuarta condición donde se definen operaciones para una velocidad de 20 km/h.

    nivnorm=[50 50 50 49 48 49 50 50 51 51 51 51 49 47 44 41 39 36]; %Definición de los niveles de presión sonora mínimos en dB(A) según normativa para 20 km/h.
    plot(vectfrec,nivnorm,'-ro','LineWidth',2,'MarkerSize',7,'MarkerFaceColor',[1 .6 .6],'MarkerEdgeColor','r') %Representación de los niveles anteriormente definidos.

else if (v==5) %Quinta condición donde se definen operaciones para una velocidad de 25 km/h.

    nivnorm=[52 52 51.6 50.6 49.6 50.6 51.6 51.6 52.6 52.6 52.6 52.6 50.6 48.6 45.6 42.6 40.6 37.6]; %Definición de los niveles de presión sonora mínimos en dB(A) según normativa para 25 km/h.
    plot(vectfrec,nivnorm,'-ro','LineWidth',2,'MarkerSize',7,'MarkerFaceColor',[1 .6 .6],'MarkerEdgeColor','r') %Representación de los niveles anteriormente definidos.

else if (v==6) %Sexta condición donde se definen operaciones para una velocidad de 30 km/h.

    nivnorm=[53 53 53 52 51 52 53 53 54 54 54 54 52 50 47 44 42 39]; %Definición de los niveles de presión sonora mínimos en dB(A) según normativa para 30 km/h.
    plot(vectfrec,nivnorm,'-ro','LineWidth',2,'MarkerSize',7,'MarkerFaceColor',[1 .6 .6],'MarkerEdgeColor','r') %Representación de los niveles anteriormente definidos.

end %Cierre de sexta condición.

end %Cierre de quinta condición.

end %Cierre de cuarta condición.

end %Cierre de tercera condición.

end %Cierre de segunda condición.

end %Cierre de condiciones.

hold off %Eliminación de mantenimiento de gráfica en ventana.
grid minor % Colocación cuadrícula en la gráfica.
set(gca,'XTick',vectfrec); % Dibuja las marcas del eje x necesarias.
set(gca,'XTickLabel',F); % Renombra las marcas con las frecuencias.
set(gca,'XTickLabelRotation',45); % Gira las marcas del eje x.
xlabel('Frecuencia [Hz]'); % Colocación de etiqueta en el eje X para el gráfico de barras.
ylabel('Presión sonora en dB(A)'); % Colocación de etiqueta en el eje Y para el gráfico de barras.
title('ESPECTRO DE FRECUENCIAS (TERCIO DE OCTAVA)') % Colocación de título para el gráfico de barras.
```

Figura 4.24. Representación completa del espectro de frecuencias en el programa espfrec.

En una ventana nueva, se define a través del comando “bar” la representación del espectro de frecuencia en un gráfico de barras. Los datos del eje X se representan mediante la variable “vectfrec”, que contiene una posición por banda.

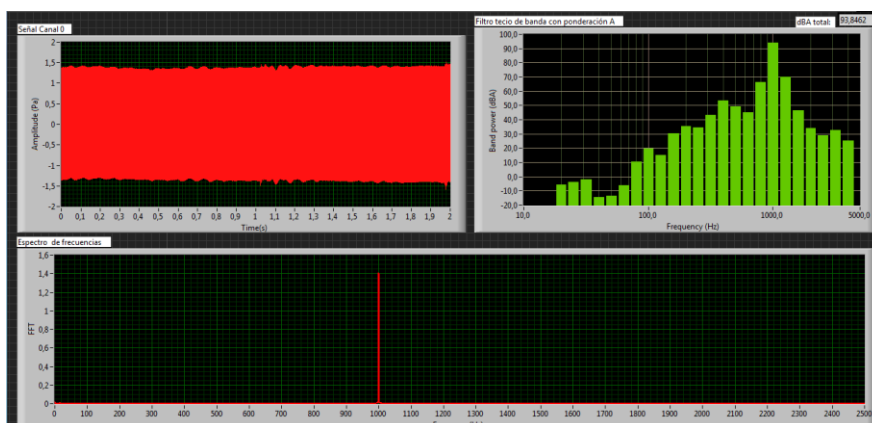
Para los valores del eje Y se utiliza la variable “P”. También se definen parámetros que cambian el diseño del gráfico (color de las barras, contorno y tamaño...). Con el “hold on” se mantiene la gráfica en la ventana sin que se produzcan cambios, sobreponiendo las operaciones.

Tras esto, se establecen una serie de condiciones: una por cada velocidad a la que se realizaron pasadas en los ensayos (6 en total). Dentro de cada condición, se establecen los niveles de presión sonora para cada banda de tercio de octava y se realiza la representación de los datos, para la velocidad correspondiente. Así, se obtiene el gráfico de barras con el nivel de presión sonora para cada banda y pasada específica en comparación con el nivel de presión sonora recogida en la normativa.

Finalmente, se configura la ventana para adecuar la presentación de los gráficos. En primer lugar, se establece la cuadrícula con “grid minor. Se configuran también tres “set”: el primero dispone en el gráfico los valores de “vectfrec”; el segundo se encarga de cambiar los valores del “vectfrec” por los valores almacenados en “F” y el último set gira las etiquetas del eje X un ángulo de 45°.

#### 4.4.1.5. VERIFICACIÓN

Al concluir el desarrollo de los programas anteriores, se realizó una prueba de verificación, para establecer si los programas realmente funcionaban y proporcionaban resultados fidedignos. Para ello se generó una señal de 1.4 Pa de amplitud emitida a una frecuencia de 1000 Hz, con una frecuencia de muestreo de 10000 datos/segundo. Los datos de esta señal quedan representados en la Figura 4.25.



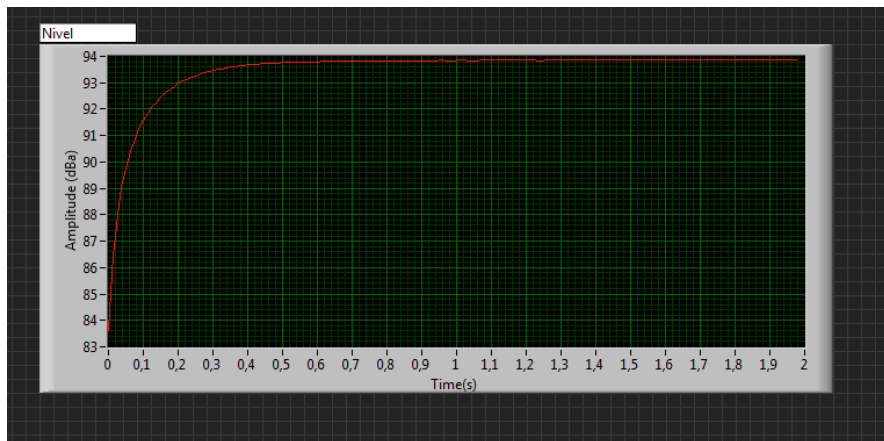
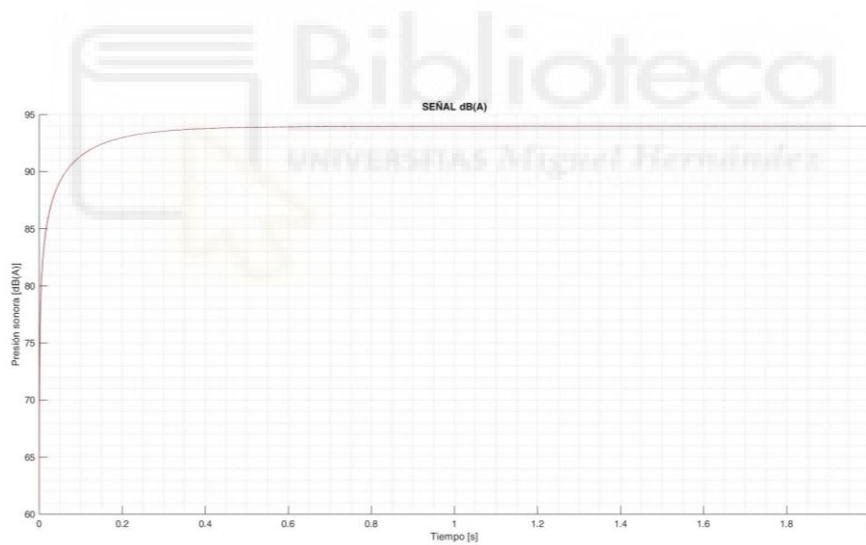


Figura 4.25. Información de la señal de verificación.

Estos son los datos con los que comparar los resultados que se obtengan en los programas generados. Para ello, se introdujo en Matlab las variables que definían la señal de verificación y se ejecutaron los programas anteriores. Las respuestas proporcionadas por estos, se visualizan en la Figura 4.26.



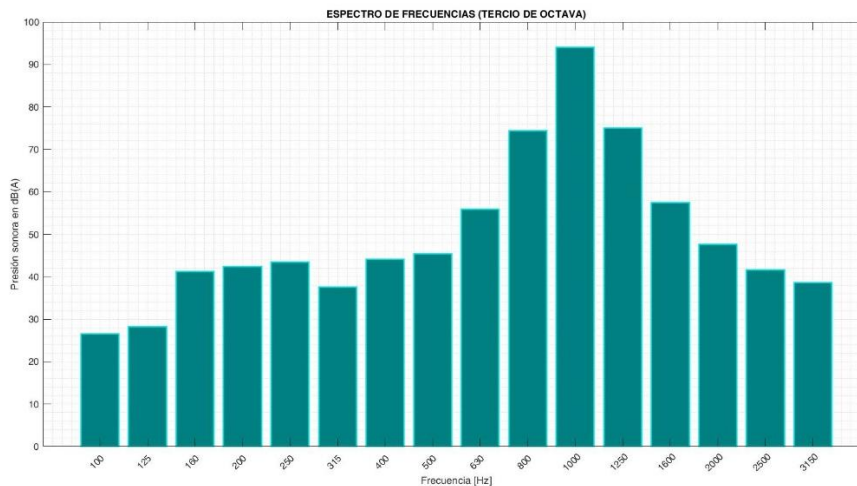


Figura 4.26. Información proporcionada por los programas creados en Matlab.

Como se puede observar, se obtienen resultados muy similares a los proporcionados para realizar la verificación (existe una pequeña disparidad en los resultados que proporciona el espectro de frecuencia, aunque la forma en la que se distribuyen las bandas es similar).

#### 4.4.1.6. ESQUEMA DE ARCHIVOS Y PROGRAMAS

Hasta este paso, se han obtenido los datos de una pasada determinada, con una señal de advertencia específica y a una velocidad definida. No obstante, para agilizar y automatizar el procesado y recolección de datos, se creó una serie de programas en forma de árbol que extendiesen el cálculo a todas las velocidades de todos los sonidos de advertencia (aunque no simultáneamente). El esquema siguiente refleja la distribución utilizada.

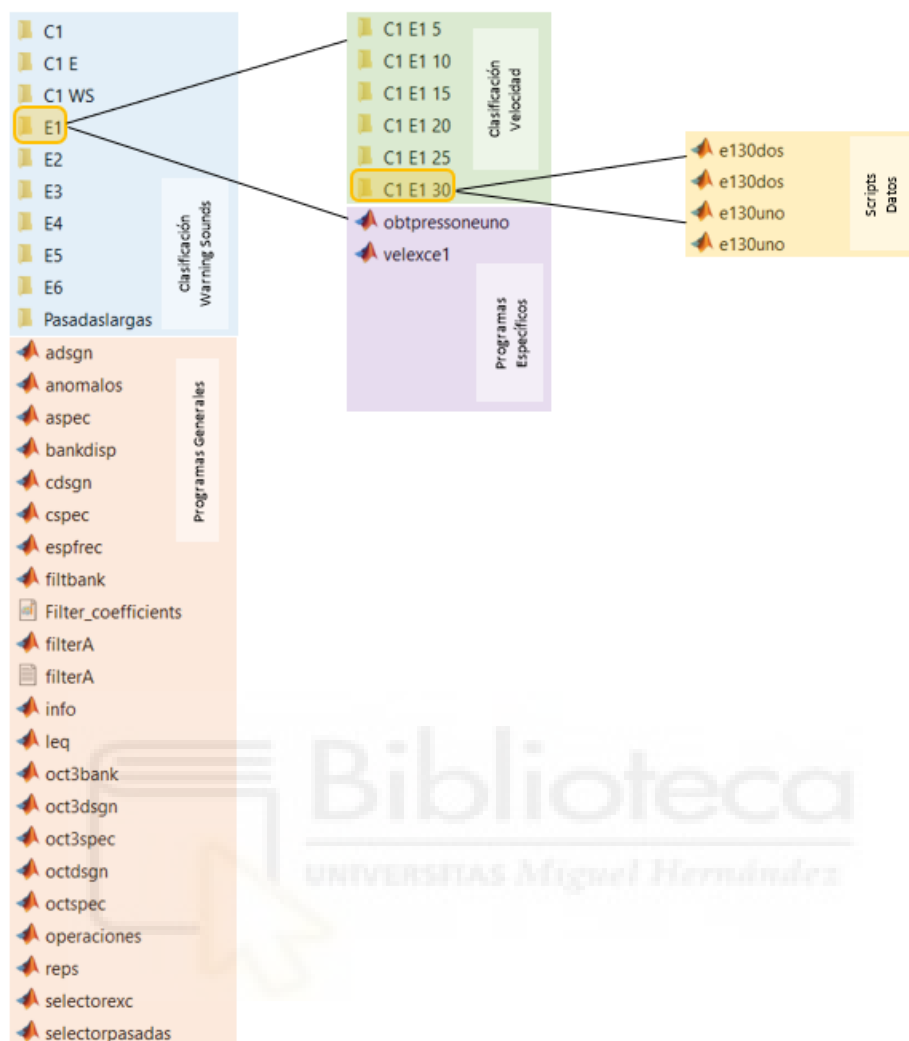


Figura 4.27. Disposición de programas y archivos para el apartado de nivel sonoro equivalente máximo.

#### 4.4.1.7. PROGRAMA SELECTORPASADAS

En el programa “selectorpasadas”, se distingue entre “Pasadas Cortas” y “Pasadas Largas”.

El primer bloque del programa determina si el usuario desea estudiar un caso de Pasada Corta o Pasada Larga.

```
%% PETICIÓN WARNING SOUND PASADAS CORTAS O LARGAS

display('1-Pasadas Cortas') %Escribir por pantalla significado del código 1.
display('2-Pasadas Largas') %Escribir por pantalla significado del código 2.

pas=input('¿Qué tipo de warning sound quiere analizar? '); %Pregunta por pantalla el tipo de warning sound a analizar y lo almacena.
```

Figura 4.28. Definición de códigos para Pasadas Cortas y Largas y pregunta del tipo de pasada a analizar.

Se hace uso de “display” para mostrar por pantalla el código correspondiente para cada tipo de pasada. Posteriormente, a través de “input”, el usuario decide qué quiere estudiar.

A continuación, se establecen cada una de las rutas a seguir para poder acceder a los Programas Específicos (encargados de obtener los datos específicos de cada velocidad y sonido de advertencia en concreto). Estos programas específicos solamente son necesarios para los casos de Pasadas Cortas, puesto que la toma de datos en las Pasadas Largas se realizó a una única velocidad (20 km/h).

```
switch pas %Estructura de control switch que realiza las funciones que se definan para cada clase de warning sound según sean pasadas largas o cortas.

    case 1 %Operaciones para opción Pasadas Cortas.

%% PETICIÓN DE WARNING SOUND PASADAS CORTAS A EJECUTAR

display('E1-Warning sound E1') %Mostrar por pantalla que código es el warning sound E1.
display('E2-Warning sound E2') %Mostrar por pantalla que código es el warning sound E2.
display('E3-Warning sound E3') %Mostrar por pantalla que código es el warning sound E3.
display('E4-Warning sound E4') %Mostrar por pantalla que código es el warning sound E4.
display('E5-Warning sound E5') %Mostrar por pantalla que código es el warning sound E5.
display('E6-Warning sound E6') %Mostrar por pantalla que código es el warning sound E6.
display('C1-Warning sound C1') %Mostrar por pantalla que código es el warning sound C1.
display('C1 E-Warning sound C1 E') %Mostrar por pantalla que código es el warning sound C1 E.
display('C1 WS-Warning sound C1 WS') %Mostrar por pantalla que código es el warning sound C1 WS.

ws=input('¿Qué warning sound quiere estudiar?(En mayúsculas) ','s'); %Pregunta por pantalla el warning sound a analizar y lo almacena.

%% LISTADO PASADAS CORTAS

switch ws %Estructura de control switch que realiza las funciones que se definan para cada warning sound de Pasadas Cortas.

    case 'E1' %Operaciones para warning sound E1.

        run('..\Tarea 1\E1\obtpresoneuno.m') %Ejecución del script con nombre 'obtpresoneuno'.

    case 'E2' %Operaciones para warning sound E2.

        run('..\Tarea 1\E2\obtpresonedos.m') %Ejecución del script con nombre 'obtpresnedos'.

    case 'E3' %Operaciones para warning sound E3.

        run('..\Tarea 1\E3\obtpresonetres.m') %Ejecución del script con nombre 'obtpresonetres'.

    case 'E4' %Operaciones para warning sound E4.

        run('..\Tarea 1\E4\obtpresonecuatro.m') %Ejecución del script con nombre 'obtpresonecuatro'.

    case 'E5' %Operaciones para warning sound E5.

        run('..\Tarea 1\E5\obtpresonecinco.m') %Ejecución del script con nombre 'obtpresonecinco'.

    case 'E6' %Operaciones para warning sound E6.

        run('..\Tarea 1\E6\obtpresoneseis.m') %Ejecución del script con nombre 'obtpresoneseis'.

    case 'C1' %Operaciones para warning sound C1.

        run('..\Tarea 1\C1\obtpresoncuno.m') %Ejecución del script con nombre 'obtpresoncuno'.

    case 'C1 E' %Operaciones para warning sound C1 E.

        run('..\Tarea 1\C1 E\obtpresoncunoe.m') %Ejecución del script con nombre 'obtpresoncunoe'.

    case 'C1 WS' %Operaciones para warning sound C1 WS.

        run('..\Tarea 1\C1 WS\obtpresoncunows.m') %Ejecución del script con nombre 'obtpresoncunows'.

    otherwise %En el caso de que el código introducido no coincida con los casos anteriores se realiza lo siguiente.

        display('ERROR') %Escribir por pantalla error.
        display('Introduzca un valor dentro del intervalo') %Escribir por pantalla que se debe hacer para que no de error.
        return %Salida de la estructura de control.

end %Cierre de la estructura de control switch para códigos de Pasadas Cortas.
```

Figura 4.29. Estructura de los casos Pasadas Cortas en el programa selectorpasadas para calcular el nivel sonoro equivalente máximo.

El bloque comienza con la definición de un “switch”, empleado para distinguir entre los dos casos de pasadas que existen. Así, se consiguen dos rutas distintas, a seguir, para manipular los datos. Seguidamente, a través del “display”, se define el código correspondiente a cada sonido de advertencia. Después, se escoge qué señal de advertencia se desea estudiar y el almacenamiento del código introducido en una variable.

Después del almacenamiento del código, se define otro “switch”, que contiene todas las rutas. Estas rutas direccionan a un conjunto de programas, que siguen la nomenclatura:

ruta = obtpression + sonido de advertencia estudiado

Por ejemplo, para estudiar una velocidad del sonido de advertencia E3, el programa proporciona los datos en “obtpressonsonetres”. Estos programas contienen las rutas para obtener los datos de todas las velocidades de cada señal de advertencia. Existe también otro caso dentro de la estructura de control (“otherwise”), activado cuando el código introducido por pantalla no coincide con ninguno de los sonidos de advertencia propuestos, mostrando por pantalla la existencia de un error.

La siguiente parte del programa se corresponde con las operaciones a realizar con los ensayos realizados con Pasadas Largas.

```
case 2 %Operaciones para opción Pasadas Largas.

%% PETICIÓN DE WARNING SOUND PASADAS LARGAS

display('P1-Warning sound P1') %Mostrar por pantalla que código es el warning sound P1.
display('P2-Warning sound P2') %Mostrar por pantalla que código es el warning sound P2.
display('P3-Warning sound P3') %Mostrar por pantalla que código es el warning sound P3.
display('P4-Warning sound P4') %Mostrar por pantalla que código es el warning sound P4.
display('P5-Warning sound P5') %Mostrar por pantalla que código es el warning sound P5.
display('P8-Warning sound P8') %Mostrar por pantalla que código es el warning sound P8.
display('P9-Warning sound P9') %Mostrar por pantalla que código es el warning sound P9.
display('P10-Warning sound P10') %Mostrar por pantalla que código es el warning sound P10.
display('P11-Warning sound P11') %Mostrar por pantalla que código es el warning sound P11.
display('P12-Warning sound P12') %Mostrar por pantalla que código es el warning sound P12.
display('P13-Warning sound P13') %Mostrar por pantalla que código es el warning sound P13.
display('P16-Warning sound P16') %Mostrar por pantalla que código es el warning sound P16.
display('P17-Warning sound P17') %Mostrar por pantalla que código es el warning sound P17.
display('P18-Warning sound P18') %Mostrar por pantalla que código es el warning sound P18.
display('P19-Warning sound P19') %Mostrar por pantalla que código es el warning sound P19.
display('P20-Warning sound P20') %Mostrar por pantalla que código es el warning sound P20.

ws=input('¿Qué warning sound quiere estudiar?(En mayúsculas) ','s'); %Pregunta por pantalla el warning sound a analizar y lo almacena.
```

```
switch ws %Estructura de control switch que realiza las funciones que se definan para cada warning sound de Pasadas Largas.

case 'P1' %Operaciones para warning sound P1.

run('..\Tarea 1\Pasadaslargas\P1\pasadalargauno.m') %Ejecución del script con nombre 'pasadalargauno'.
time=xanalog(microphone); %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre 'operaciones'.
dBAmax1=dBAmacr; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P1\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Cálculo número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre 'info'.

case 'P2' %Operaciones para warning sound P2.

run('..\Tarea 1\Pasadaslargas\P2\pasadalargados.m') %Ejecución del script con nombre 'pasadalargados'.
time=xanalog(microphone); %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre 'operaciones'.
dBAmax1=dBAmacr; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P2\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Cálculo número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre 'info'.

case 'P3' %Operaciones para warning sound P3.

run('..\Tarea 1\Pasadaslargas\P3\pasadalargatres.m') %Ejecución del script con nombre 'pasadalargatres'.
time=xanalog(microphone); %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre 'operaciones'.
dBAmax1=dBAmacr; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P3\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Cálculo de número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre 'info'.

case 'P4' %Operaciones para warning sound P4.

run('..\Tarea 1\Pasadaslargas\P4\pasadalargacuatro.m') %Ejecución del script con nombre 'pasadalargacuatro'.
time=xanalog(microphone); %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre 'operaciones'.
dBAmax1=dBAmacr; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P4\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Cálculo de número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre 'info'.

case 'P5' %Operaciones para warning sound P5.

run('..\Tarea 1\Pasadaslargas\P5\pasadalargacinco.m') %Ejecución del script con nombre 'pasadalargacinco'.
time=xanalog(microphone); %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre 'operaciones'.
dBAmax1=dBAmacr; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P5\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre 'info'.

case 'P8' %Operaciones para warning sound P8.

run('..\Tarea 1\Pasadaslargas\P8\pasadalargaocho.m') %Ejecución del script con nombre 'pasadalargaocho'.
time=xanalog(microphone); %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre 'operaciones'.
dBAmax1=dBAmacr; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P8\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre 'info'.
```



```
case 'P9' %Operaciones para warning sound P9.

run('..\Tarea 1\Pasadaslargas\P9\pasadalarganueve.m') %Ejecución del script con nombre 'pasadalarganueve'.
time=xanalog(microphone); %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre 'operaciones'.
dBAmx1=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmx dBAR; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P9\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre 'info'.

case 'P10' %Operaciones para warning sound P10.

run('..\Tarea 1\Pasadaslargas\P10\pasadalargadiez.m') %Ejecución del script con nombre 'pasadalargadiez'.
time=xanalog(microphone); %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre 'operaciones'.
dBAmx1=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmx dBAR; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P10\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre 'info'.

case 'P11' %Operaciones para warning sound P11.

run('..\Tarea 1\Pasadaslargas\P11\pasadalargaonce.m') %Ejecución del script con nombre 'pasadalargaonce'.
time=xanalog(microphone); %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre 'operaciones'.
dBAmx1=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmx dBAR; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P11\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre 'info'.

case 'P12' %Operaciones para warning sound P12.

run('..\Tarea 1\Pasadaslargas\P12\pasadalargadoce.m') %Ejecución del script con nombre 'pasadalargadoce'.
time=xanalog(microphone); %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre 'operaciones'.
dBAmx1=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmx dBAR; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P12\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre 'info'.

case 'P13' %Operaciones para warning sound P13.

run('..\Tarea 1\Pasadaslargas\P13\pasadalargatrece.m') %Ejecución del script con nombre 'pasadalargatrece'.
time=xanalog(microphone); %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre 'operaciones'.
dBAmx1=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmx dBAR; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P13\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre 'info'.

case 'P14' %Operaciones para warning sound P14.

run('..\Tarea 1\Pasadaslargas\P14\pasadalargatorce.m') %Ejecución del script con nombre 'pasadalargatorce'.
time=xanalog(microphone); %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre 'operaciones'.
dBAmx1=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmx dBAR; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P14\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre 'info'.
```

```

case 'P16' %Operaciones para warning sound P16.

run('..\Tarea 1\Pasadaslargas\P16\pasadalargadieciseis.m') %Ejecución del script con nombre 'pasadalargadieciseis'.
time=xanalog(microphone); %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre 'operaciones'.
dBAmax1=dBAmacr; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P16\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre 'info'.

case 'P17' %Operaciones para warning sound P17.

run('..\Tarea 1\Pasadaslargas\P17\pasadalargadiecisisiete.m') %Ejecución del script con nombre 'pasadalargadiecisisiete'.
time=xanalog(microphone); %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre 'operaciones'.
dBAmax1=dBAmacr; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P17\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre 'info'.

case 'P18' %Operaciones para warning sound P18.

run('..\Tarea 1\Pasadaslargas\P18\pasadalargadieciocho.m') %Ejecución del script con nombre 'pasadalargadieciocho'.
time=xanalog(microphone); %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre 'operaciones'.
dBAmax1=dBAmacr; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P18\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre 'info'.

case 'P19' %Operaciones para warning sound P19.

run('..\Tarea 1\Pasadaslargas\P19\pasadalargadiecinueve.m') %Ejecución del script con nombre 'pasadalargadiecinueve'.
time=xanalog(microphone); %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre 'operaciones'.
dBAmax1=dBAmacr; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P19\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre 'info'.

case 'P20' %Operaciones para warning sound P20.

run('..\Tarea 1\Pasadaslargas\P20\pasadalargaveinte.m') %Ejecución del script con nombre 'pasadalargaveinte'.
time=xanalog(microphone); %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre 'operaciones'.
dBAmax1=dBAmacr; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P20\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre 'info'.

otherwise %En el caso de que el código introducido no coincida con los casos anteriores se realiza lo siguiente.
display ('ERROR') %Escribir por pantalla error.
display ('Introduzca un valor dentro del intervalo') %Escribir por pantalla que se debe hacer para que no de error.
return %Salga de la estructura de control.

end %Cierre de la estructura de control switch para códigos de Pasadas Largas.

otherwise

display ('ERROR') %Escribir por pantalla error.
display ('Introduzca un valor dentro del intervalo') %Escribir por pantalla que se debe hacer para que no de error.
return %Salga de la estructura de control.

end %Cierre de la estructura de control switch para tipos de pasadas.

```

Figura 4.30. Estructura de los casos Pasadas Largas en el programa selectorpasadas para calcular el nivel sonoro equivalente máximo.

Esta parte sigue el mismo principio que el bloque anterior. Mediante un “switch”, se genera un caso para cada una de las pasadas. Se agrega otro donde se muestre el error si el código introducido por el usuario no se corresponde con ningún caso (se entraría en el “otherwise” final).

Se recuerda que todos los casos de Pasadas Largas se realizaron a la misma velocidad. Por ello, el código desarrollado en cada parte es similar al de los programas explicados en futuros apartados.

Cada caso comienza con la ejecución del programa de obtención de datos de presión sonora de la señal a lo largo del tiempo. En segundo lugar, se definen una serie de variables, como el tiempo (“time”) o velocidad de pasada. Tras esto, “operaciones” manipula y representa los datos. A continuación, se almacenan una serie de variables que posteriormente se recopilarán en un archivo Excel. Seguidamente, por medio de “dir”, se determina el nombre de los archivos con extensión .m encontrados en la carpeta de almacenamiento de datos de la señal. Finalmente, se ejecuta el programa “info”.

#### 4.4.1.8. CONJUNTO DE PROGRAMAS OBTPRESSON + SONIDO DE ADVERTENCIA

Para trabajar con los ensayos de Pasadas Cortas, es necesario el empleo de los programas “obtpresson + sonido de advertencia”.

Este programa se encarga de obtener los valores de presión sonora y de tiempo de cada señal, en función de la velocidad y el sonido de advertencia seleccionados.

```
%% PETICIÓN DE VELOCIDAD A EJECUTAR

display('1-Velocidad a 5 km/h') %Mostrar por pantalla que código es la velocidad a 5 km/h.
display('2-Velocidad a 10 km/h') %Mostrar por pantalla que código es la velocidad a 10 km/h.
display('3-Velocidad a 15 km/h') %Mostrar por pantalla que código es la velocidad a 15 km/h.
display('4-Velocidad a 20 km/h') %Mostrar por pantalla que código es la velocidad a 20 km/h.
display('5-Velocidad a 25 km/h') %Mostrar por pantalla que código es la velocidad a 25 km/h.
display('6-Velocidad a 30 km/h') %Mostrar por pantalla que código es la velocidad a 30 km/h.

v=input('¿Qué velocidad quiere estudiar? '); %Pregunta por pantalla código de velocidad a estudiar y almacenamiento en variable.

%% LISTADO VELOCIDADES
switch v %Estructura de control switch que realiza las funciones que se definan para cada caso de velocidad para warning sound El.
```

```
switch v %Estructura de control switch que realiza las funciones que se definan para cada caso de velocidad para warning sound E1.

case 1 %Operaciones para velocidad de 5 km/h.

arch=dir('C1 E1 5\*.m'); %Obtención de lista de archivos con extensión .m para 5 km/h.
numarch=length(arch); %Número de archivos que permiten obtener los datos con extensión .m para 5 km/h.

for i=1:numarch %Bucle for para cambiar el valor de i (i es el número de ensayo) para esa velocidad.

    switch i %Estructura de control switch que realice las funciones especificadas para cada ensayo de 5 km/h.

        case 1 %Operaciones para primer ensayo a 5 km/h.

            run('..\E1\C1 E1 5\el1uno.m') %Ejecución del archivo del primer ensayo.
            pressonpa1=data; %Almacenamiento de los datos de presión sonora a esa velocidad del primer ensayo en variable.
            time1=xanalog(microphone); %Almacenamiento de los datos de tiempo a esa velocidad del primer ensayo en variable.

        case 2 %Operaciones para segundo ensayo a 5 km/h.

            run('..\E1\C1 E1 5\el15dos.m') %Ejecución del archivo del segundo ensayo.
            pressonpa2=data; %Almacenamiento de los datos de presión sonora a esa velocidad del segundo ensayo en variable.
            time2=xanalog(microphone); %Almacenamiento de los datos de tiempo a esa velocidad del segundo ensayo en variable.

        end %Cierre para los ensayos a 5 km/h.

    end %Cierre del bucle.

case 2 %Operaciones para velocidad de 10 km/h.

arch=dir('C1 E1 10\*.m'); %Obtención de lista de archivos con extensión .m para 10 km/h.
numarch=length(arch); %Número de archivos que permiten obtener los datos con extensión .m para 10 km/h.

for i=1:numarch %Bucle for para cambiar el valor de i (i es el número de ensayo) para esa velocidad.

    switch i %Estructura de control switch que realice las funciones especificadas para cada ensayo de 5 km/h.

        case 1 %Operaciones para primer ensayo a 10 km/h.

            run('..\E1\C1 E1 10\el10uno.m') %Ejecución del archivo del primer ensayo.
            pressonpa1=data; %Almacenamiento de los datos de presión sonora a esa velocidad del primer ensayo en variable.
            time1=xanalog(microphone); %Almacenamiento de los datos de tiempo a esa velocidad del primer ensayo en variable.

        end %Cierre para los ensayos a 10 km/h.

    end %Cierre del bucle.

case 3 %Operaciones para velocidad de 15 km/h.

arch=dir('C1 E1 15\*.m'); %Obtención de lista de archivos con extensión .m para 15 km/h.
numarch=length(arch); %Número de archivos que permiten obtener los datos con extensión .m para 15 km/h.

for i=1:numarch %Bucle for para cambiar el valor de i (i es el número de ensayo) para esa velocidad.

    switch i %Estructura de control switch que realice las funciones especificadas para cada ensayo de 15 km/h.

        case 1 %Operaciones para primer ensayo a 15 km/h.

            run('..\E1\C1 E1 15\el15uno.m') %Ejecución del archivo del primer ensayo.
            pressonpa1=data; %Almacenamiento de los datos de presión sonora a esa velocidad del primer ensayo en variable.
            time1=xanalog(microphone); %Almacenamiento de los datos de tiempo a esa velocidad del primer ensayo en variable.

        case 2 %Operaciones para segundo ensayo a 15 km/h.

            run('..\E1\C1 E1 15\el15dos.m') %Ejecución del archivo del segundo ensayo.
            pressonpa2=data; %Almacenamiento de los datos de presión sonora a esa velocidad del segundo ensayo en variable.
            time2=xanalog(microphone); %Almacenamiento de los datos de tiempo a esa velocidad del segundo ensayo en variable.

        end %Cierre para los ensayos a 15 km/h.

    end %Cierre del bucle.
```

```
case 4 %Operaciones para velocidad de 20 km/h.

arch=dir('C1 E1 20\*.m'); % Obtención de lista de archivos con extensión .m para 20 km/h.
numarch=length(arch); % Número de archivos que permiten obtener los datos con extensión .m para 20 km/h.

for i=1:numarch %Estructura de control switch que realice las funciones especificadas para cada ensayo de 20 km/h.

    switch i %Estructura de control switch que realice las funciones especificadas para cada ensayo de 20 km/h.

        case 1 %Operaciones para segundo ensayo a 20 km/h.

            run('..\E1\C1 E1 20\el120uno.m') %Ejecución del archivo del primer ensayo.
            pressonpal=data; %Almacenamiento de los datos de presión sonora a esa velocidad del primer ensayo en variable.
            time1=xanalog(microphone); %Almacenamiento de los datos de tiempo a esa velocidad del primer ensayo en variable.

        case 2 %Operaciones para segundo ensayo a 20 km/h.

            run('..\E1\C1 E1 20\el120dos.m') %Ejecución del archivo del segundo ensayo.
            pressonpa2=data; %Almacenamiento de los datos de presión sonora a esa velocidad del segundo ensayo en variable.
            time2=xanalog(microphone); %Almacenamiento de los datos de tiempo a esa velocidad del segundo ensayo en variable.

        end %Cierre para los ensayos a 20 km/h.

    end %Cierre del bucle.

case 5 %Operaciones para velocidad de 25 km/h.

arch=dir('C1 E1 25\*.m'); % Obtención de lista de archivos con extensión .m para 25 km/h.
numarch=length(arch); % Número de archivos que permiten obtener los datos con extensión .m para 25 km/h.

for i=1:numarch %Estructura de control switch que realice las funciones especificadas para cada ensayo de 25 km/h.

    switch i %Estructura de control switch que realice las funciones especificadas para cada ensayo de 25 km/h.

        case 1 %Operaciones para segundo ensayo a 25 km/h.

            run('..\E1\C1 E1 25\el125uno.m') %Ejecución del archivo del primer ensayo.
            pressonpal=data; %Almacenamiento de los datos de presión sonora a esa velocidad del primer ensayo en variable.
            time1=xanalog(microphone); %Almacenamiento de los datos de tiempo a esa velocidad del primer ensayo en variable.

        case 2 %Operaciones para segundo ensayo a 25 km/h.

            run('..\E1\C1 E1 25\el125dos.m') %Ejecución del archivo del segundo ensayo.
            pressonpa2=data; %Almacenamiento de los datos de presión sonora a esa velocidad del segundo ensayo en variable.
            time2=xanalog(microphone); %Almacenamiento de los datos de tiempo a esa velocidad del segundo ensayo en variable.

        end %Cierre para los ensayos a 25 km/h.

    end %Cierre del bucle.

case 6 %Operaciones para velocidad de 30 km/h.

arch=dir('C1 E1 30\*.m'); % Obtención de lista de archivos con extensión .m para 30 km/h.
numarch=length(arch); % Número de archivos que permiten obtener los datos con extensión .m para 30 km/h.

for i=1:numarch %Estructura de control switch que realice las funciones especificadas para cada ensayo de 30 km/h.

    switch i %Estructura de control switch que realice las funciones especificadas para cada ensayo de 30 km/h.

        case 1 %Operaciones para segundo ensayo a 30 km/h.

            run('..\E1\C1 E1 30\el130uno.m') %Ejecución del archivo del primer ensayo.
            pressonpal=data; %Almacenamiento de los datos de presión sonora a esa velocidad del primer ensayo en variable.
            time1=xanalog(microphone); %Almacenamiento de los datos de tiempo a esa velocidad del primer ensayo en variable.

        case 2 %Operaciones para segundo ensayo a 30 km/h.

            run('..\E1\C1 E1 30\el130dos.m') %Ejecución del archivo del segundo ensayo.
            pressonpa2=data; %Almacenamiento de los datos de presión sonora a esa velocidad del segundo ensayo en variable.
            time2=xanalog(microphone); %Almacenamiento de los datos de tiempo a esa velocidad del segundo ensayo en variable.

        end %Cierre para los ensayos a 30 km/h.

    end %Cierre del bucle.

otherwise %En el caso de que el código introducido no coincida con los casos anteriores se realiza lo siguiente.

    display('ERROR') %Escribir por pantalla error.
    display('Introduzca un valor dentro del intervalo especificado') %Escribir por pantalla que se debe hacer para que no de error.
    return %Salga de la estructura de control.

end %Cierre de la estructura de control para E1.

run('..\Tarea 1\reps.m') %Ejecución del script con nombre 'reps'.
```

Figura 4.31. Programa obtpressone1.

En primer lugar, el programa muestra por pantalla el código para cada velocidad a la que se ensayó el sonido de advertencia (de 5 km/h a 25-30 km/h, con incrementos de 5 km/h). Tras establecer los códigos, se pide el ingreso de la velocidad estudiada y se almacena en la variable “v”. A continuación, un “switch” define un caso para cada una de las velocidades presentes en el sonido de advertencia a analizar. Tras la estructura de control, se averigua el nombre de los archivos con extensión .m que existen en la carpeta de cada velocidad y su cantidad. El número de archivos se almacena en la variable “numarch”.

En segundo lugar, con un bucle “for”, se varía un contador para la selección de cada archivo de datos presente en una misma velocidad. Tras el bucle, se establece otro “switch” donde cada caso se corresponde a cada uno de los archivos .m de la carpeta de velocidad a analizar. La cantidad de archivos presentes para cada velocidad varía normalmente entre dos y cuatro.

Dentro de cada caso, se ejecuta el archivo .m que almacena los datos de la señal correspondiente a la pasada estudiada. A su vez, se almacena la presión sonora en Pa en una variable y el tiempo en otra. En función del caso, las variables que almacenan los datos de presión sonora (“pressonpa”) y el tiempo (“time”) sufren modificaciones. Esta modificación se basa en la asignación del número del caso que se esté estudiando al final de la variable: en el segundo caso, estarían “pressonpa2” y “time2”. Esto se realiza para que no se produzca sobreposición de datos ni interferencias.

Seguidamente, se marca un caso extra para el “switch”, que indica error en caso de que el código introducido no sea válido. Finalmente, se ejecuta el programa “reps”.

#### 4.4.1.9. PROGRAMA REPS

El programa “reps” es el encargado de, en función de la cantidad de archivos .m, repetir el conjunto de operaciones de manipulación de datos para cada uno de los conjuntos de información de cada pasada.

```
%% DEFINICIÓN DEL VECTOR PRESIÓN PARA ENTRAR EN EL BUCLE
switch numarch %Estructura de control switch que realiza las funciones que se definan para cada caso en función del número de archivos según su extensión.

case 1 %Operaciones para casos con 1 archivo de extensión determinada.
    %% Ejecución del script para el primer archivo.
    data=pressonpal; %Almacenamiento datos vector presión sonora en variable del primer archivo.
    time=timel; %Almacenamiento datos vector tiempo en variable del primer archivo.
    run('..\Varea \operaciones.m') %Ejecución del script con nombre "operaciones"
    dBAmxl=dBAmxrx; %Almacenamiento del valor máximo de dB(A) redondeado en variable del primer archivo.
    tl=tmxrdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable del primer archivo.
    tfl=tf; %Almacenamiento del valor del último instante del vector tiempo del primer archivo.
    Pl=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava del primer archivo.

case 2 %Operaciones para casos con 2 archivos de extensión determinada.
    %% Ejecución del script para el primer archivo.
    data=pressonpal; %Almacenamiento datos vector presión sonora en variable del primer archivo.
    time=timel; %Almacenamiento datos vector tiempo en variable del primer archivo.
    run('..\Varea \operaciones.m') %Ejecución del script con nombre "operaciones"
    dBAmxl=dBAmxrx; %Almacenamiento del valor máximo de dB(A) redondeado en variable del primer archivo.
    tl=tmxrdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable del primer archivo.
    tfl=tf; %Almacenamiento del valor del último instante del vector tiempo del primer archivo.
    Pl=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava del primer archivo.
```

```

%% Ejecución del script para el segundo archivo.
data=pressonpa2; %Almacenamiento datos vector presión sonora en variable del segundo archivo.
time=time2; %Almacenamiento datos vector tiempo en variable del segundo archivo.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".
dBAmx2=dBAmxR; %Almacenamiento del valor máximo de dB(A) redondeado en variable del segundo archivo.
t2=tmx2dBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable del segundo archivo.
tf2=tf; %Almacenamiento del valor del último instante del vector tiempo del segundo archivo.
P2=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava del segundo archivo.

case 3 %Operaciones para casos con 3 archivos de extensión determinada.
%% Ejecución del script para el primer archivo.
data=pressonpa1; %Almacenamiento datos vector presión sonora en variable del primer archivo.
time=time1; %Almacenamiento datos vector tiempo en variable del primer archivo.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".
dBAmx1=dBAmxR; %Almacenamiento del valor máximo de dB(A) redondeado en variable del primer archivo.
t1=tmx1dBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable del primer archivo.
tf1=tf; %Almacenamiento del valor del último instante del vector tiempo del primer archivo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava del primer archivo.
%% Ejecución del script para el segundo archivo.
data=pressonpa2; %Almacenamiento datos vector presión sonora en variable del segundo archivo.
time=time2; %Almacenamiento datos vector tiempo en variable del segundo archivo.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".
dBAmx2=dBAmxR; %Almacenamiento del valor máximo de dB(A) redondeado en variable del segundo archivo.
t2=tmx2dBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable del segundo archivo.
tf2=tf; %Almacenamiento del valor del último instante del vector tiempo del segundo archivo.
P2=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava del segundo archivo.
%% Ejecución del script para el tercer archivo.
data=pressonpa3; %Almacenamiento datos vector presión sonora en variable del tercer archivo.
time=time3; %Almacenamiento datos vector tiempo en variable del tercer archivo.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".
dBAmx3=dBAmxR; %Almacenamiento del valor máximo de dB(A) redondeado en variable del tercer archivo.
t3=tmx3dBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable del tercer archivo.
tf3=tf; %Almacenamiento del valor del último instante del vector tiempo del tercer archivo.
P3=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava del tercer archivo.

case 4 %Operaciones para casos con 4 archivos de extensión determinada.
%% Ejecución del script para el primer archivo.
data=pressonpa1; %Almacenamiento datos vector presión sonora en variable del primer archivo.
time=time1; %Almacenamiento datos vector tiempo en variable del primer archivo.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".
dBAmx1=dBAmxR; %Almacenamiento del valor máximo de dB(A) redondeado en variable del primer archivo.
t1=tmx1dBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable del primer archivo.
tf1=tf; %Almacenamiento del valor del último instante del vector tiempo del primer archivo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava del primer archivo.
%% Ejecución del script para el segundo archivo.
data=pressonpa2; %Almacenamiento datos vector presión sonora en variable del segundo archivo.
time=time2; %Almacenamiento datos vector tiempo en variable del segundo archivo.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".
dBAmx2=dBAmxR; %Almacenamiento del valor máximo de dB(A) redondeado en variable del segundo archivo.
t2=tmx2dBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable del segundo archivo.
tf2=tf; %Almacenamiento del valor del último instante del vector tiempo del segundo archivo.
P2=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava del segundo archivo.
%% Ejecución del script para el tercer archivo.
data=pressonpa3; %Almacenamiento datos vector presión sonora en variable del tercer archivo.
time=time3; %Almacenamiento datos vector tiempo en variable del tercer archivo.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".
dBAmx3=dBAmxR; %Almacenamiento del valor máximo de dB(A) redondeado en variable del tercer archivo.
t3=tmx3dBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable del tercer archivo.
tf3=tf; %Almacenamiento del valor del último instante del vector tiempo del tercer archivo.
P3=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava del tercer archivo.
%% Ejecución del script para el cuarto archivo.
data=pressonpa4; %Almacenamiento datos vector presión sonora en variable del cuarto archivo.
time=time4; %Almacenamiento datos vector tiempo en variable del cuarto archivo.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".
dBAmx4=dBAmxR; %Almacenamiento del valor máximo de dB(A) redondeado en variable del cuarto archivo.
t4=tmx4dBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable del cuarto archivo.
tf4=tf; %Almacenamiento del valor del último instante del vector tiempo del cuarto archivo.
P4=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava del cuarto archivo.

end

run('..\Trabajo Matlab\info.m') %Ejecución del script con nombre "info".

```

Figura 4.32. Programa reps.

La variable “numarch” contiene el número de archivos con extensión .m. Dependiendo del número de archivos, se entrará en el caso que corresponda del “switch”. El número de casos es de uno a cuatro (dependiendo del número de pasadas mínimas y máximas, realizadas para cada velocidad en cada sonido de advertencia).

En cada caso, se realiza la transformación, representación y recolección de datos tal y como se ha explicado. Básicamente, se cogen las variables que almacenan el tiempo y la presión sonora y se traspasan a una nueva variable, con la que los programas generales trabajarán. Seguidamente, se ejecuta el programa “operaciones”, donde se obtienen distintos resultados. En cuanto a nomenclatura se refiere, si por ejemplo se desea obtener el nivel sonoro equivalente máximo del caso cuatro, este valor será almacenado en la variable dBAmx4. Finalmente, se ejecuta el programa “info”.

#### 4.4.1.10. PROGRAMA INFO

El programa “info” se encarga de realizar el almacenamiento de información recopilada a lo largo de los programas ejecutados anteriormente en una tabla. En esta tabla se recogen datos acerca de cada una de las pasadas estudiadas para la velocidad y sonido de advertencia seleccionados.

En la primera parte del programa, se establecen las variables de partida para poder generar la tabla de almacenamiento de los datos.

```
%% DEFINICIÓN DE VARIABLES DE PARTIDA.
tipvar = {'string','double','double','double'}; %Determinación del tipo de variable para cada columna de la tabla.
nomvar = {'Recording','dBA','Instant','Total'}; %Asignación del nombre de encabezado de columna.
tamano=[1 4]; %Definición inicial del tamaño de la tabla de información.
tabla = table('Size',tamano,'VariableTypes',tipvar,'VariableNames',nomvar); %Generación de tabla informativa.
```

Figura 4.33. Definición de variables en el programa info.

Primeramente, se establecen las clases de variables con las que se va a trabajar, que son del tipo cadena de caracteres y números de doble precisión. En la siguiente línea, se establece el nombre del encabezado de la tabla a emplear. Esto a su vez permite definir que variables se van a almacenar en la tabla. Por último, se asigna tamaño a la tabla y se crea la misma.

La siguiente parte se encarga de completar la tabla mencionada. La tabla analizada contendrá siempre 4 columnas, correspondientes a la información almacenada: nombre de la pasada, nivel sonoro equivalente máximo, instante de tiempo en el que se da ese máximo y el tiempo total de grabación de la señal. El número de filas, por el contrario, es variable dependiendo del número de pasadas realizadas a esa velocidad y con esa señal de advertencia (entre 1 y 4).



```
%% DESARROLLO DE LA TABLA INFORMATIVA.

switch numarch %Estructura de control switch que realiza las funciones que se definan para cada caso en función del número de archivos según su extensión.

case 1 %Operaciones para casos con 1 archivo de extensión determinada.

    archivo1=arch.name; %Almacenamiento del nombre del archivo .m correspondiente a la primera pasada.
    tamano = [1 4]; %Definición del tamaño de la tabla para este caso.
    tabla(1,:) = {archivo1,dBAmox1,t1,tf1} %Establecimiento de los valores que irá en la primera fila de la tabla.

case 2 %Operaciones para casos con 2 archivos de extensión determinada.

    arch1=arch(1); %Selección del primer archivo .m correspondiente a la primera pasada.
    arch2=arch(2); %Selección del primer archivo .m correspondiente a la segunda pasada.
    archivo1=arch1.name; %Almacenamiento del nombre del archivo .m correspondiente a la primera pasada.
    archivo2=arch2.name; %Almacenamiento del nombre del archivo .m correspondiente a la segunda pasada.
    tamano = [2 4]; %Definición del tamaño de la tabla para este caso.
    tabla(1,:) = {archivo2,dBAmox1,t1,tf1} %Establecimiento de los valores que irá en la primera fila de la tabla.
    tabla(2,:) = {archivo1,dBAmox2,t2,tf2} %Establecimiento de los valores que irá en la segunda fila de la tabla.

case 3 %Operaciones para casos con 3 archivos de extensión determinada.

    arch1=arch(1); %Selección del primer archivo .m correspondiente a la primera pasada.
    arch2=arch(2); %Selección del primer archivo .m correspondiente a la segunda pasada.
    arch3=arch(3); %Selección del primer archivo .m correspondiente a la tercera pasada.
    archivo1=arch1.name; %Almacenamiento del nombre del archivo .m correspondiente a la primera pasada.
    archivo2=arch2.name; %Almacenamiento del nombre del archivo .m correspondiente a la segunda pasada.
    archivo3=arch3.name; %Almacenamiento del nombre del archivo .m correspondiente a la tercera pasada.
    tamano = [3 4]; %Definición del tamaño de la tabla para este caso.
    tabla(1,:) = {archivo3,dBAmox1,t1,tf1} %Establecimiento de los valores que irá en la primera fila de la tabla.
    tabla(2,:) = {archivo2,dBAmox2,t2,tf2} %Establecimiento de los valores que irá en la segunda fila de la tabla.
    tabla(3,:) = {archivo1,dBAmox3,t3,tf3} %Establecimiento de los valores que irá en la tercera fila de la tabla.

case 4 %Operaciones para casos con 4 archivos de extensión determinada.

    arch1=arch(1); %Selección del primer archivo .m correspondiente a la primera pasada.
    arch2=arch(2); %Selección del primer archivo .m correspondiente a la segunda pasada.
    arch3=arch(3); %Selección del primer archivo .m correspondiente a la tercera pasada.
    arch4=arch(4); %Selección del primer archivo .m correspondiente a la cuarta pasada.
    archivo1=arch1.name; %Almacenamiento del nombre del archivo .m correspondiente a la primera pasada.
    archivo2=arch2.name; %Almacenamiento del nombre del archivo .m correspondiente a la segunda pasada.
    archivo3=arch3.name; %Almacenamiento del nombre del archivo .m correspondiente a la tercera pasada.
    archivo4=arch4.name; %Almacenamiento del nombre del archivo .m correspondiente a la cuarta pasada.
    tamano = [4 4]; %Definición del tamaño de la tabla para este caso.
    tabla(1,:) = {archivo4,dBAmox1,t1,tf1} %Establecimiento de los valores que irá en la primera fila de la tabla.
    tabla(2,:) = {archivo3,dBAmox2,t2,tf2} %Establecimiento de los valores que irá en la segunda fila de la tabla.
    tabla(3,:) = {archivo2,dBAmox3,t3,tf3} %Establecimiento de los valores que irá en la tercera fila de la tabla.
    tabla(4,:) = {archivo1,dBAmox4,t4,tf4} %Establecimiento de los valores que irá en la cuarta fila de la tabla.

end %Cierre de la estructura de control switch para generación de las tablas.

run('..\Tarea 1\selectorex.m') %Ejecución
```

Figura 4.34. Rellenado de la tabla informativa con los datos obtenidos previamente.

Para ello, la estructura empleada es un “switch” de cuatro casos. En cada caso se determinan primeramente los archivos que se han empleado para obtener la información a analizar. Justo después, se almacena el nombre del archivo en el conjunto de variables “archivo + número de pasada”. Después se define el verdadero tamaño de la tabla, que a su vez dependerá del número de archivos .m estudiados. A continuación, se incorporan los datos de cada pasada a la tabla. Un ejemplo de la ejecución del programa se muestra en la Figura 4.35.

2×4 [table](#)

<b>Recording</b>	<b>dBA</b>	<b>Instant</b>	<b>Total</b>
"e15uno.m"	51.9	6.57	11
"e15dos.m"	52	5.59	11

Figura 4.35. Rellenado de la tabla informativa con los datos obtenidos previamente.

La última línea del código “info” ejecuta el programa “selectorexc”.

#### 4.4.1.11. PROGRAMA SELECTOREXC

Este programa, define qué ruta seguir para ejecutar el programa que permita transferir los datos de Matlab a una hoja de cálculo en Excel.

La primera parte del programa está relacionada con los ensayos de Pasadas Cortas. Aquí, únicamente se establecen las rutas a los programas transferidores.

```
%% TRANSFERENCIA DE DATOS A HOJA DE EXCEL.
switch pas %Estructura de control switch que realiza las funciones que se definan para cada clase de warning sound según sean pasadas largas o cortas.
switch ws %Estructura de control switch que realiza las funciones que se definan para cada warning sound de Pasadas Cortas.

case 'E1' %Operaciones para warning sound E1.

    run('..\Tarea 1\E1\velexce1.m') %Ejecución del script con nombre ''velexce1'.

case 'E2' %Operaciones para warning sound E2.

    run('..\Tarea 1\E2\velexce2.m') %Ejecución del script con nombre ''velexce2'.

case 'E3' %Operaciones para warning sound E3.

    run('..\Tarea 1\E3\velexce3.m') %Ejecución del script con nombre ''velexce3'.

case 'E4' %Operaciones para warning sound E4.

    run('..\Tarea 1\E4\velexce4.m') %Ejecución del script con nombre ''velexce4'.

case 'E5' %Operaciones para warning sound E5.

    run('..\Tarea 1\E5\velexce5.m') %Ejecución del script con nombre ''velexce5'.

case 'E6' %Operaciones para warning sound E6.

    run('..\Tarea 1\E6\velexce6.m') %Ejecución del script con nombre ''velexce6'.

case 'C1' %Operaciones para warning sound C1.

    run('..\Tarea 1\C1\velexcc1.m') %Ejecución del script con nombre ''velexcc1'.
```

```
case 'Cl E' %Operaciones para warning sound Cl E.  
    run('..\Tarea 1\Cl E\velexccle.m') %Ejecución del script con nombre 'velexccle'.  
case 'Cl WS' %Operaciones para warning sound Cl WS.  
    run('..\Tarea 1\Cl WS\velexcc1ws.m') %Ejecución del script con nombre 'velexcc1ws'.  
end %Cierre de la estructura de control switch para códigos de Pasadas Cortas.
```

Figura 4.36. Estructura para Pasadas Cortas en el programa selectorexc.

Para realizar esto, se utiliza otra estructura “switch” para trabajar con diferentes sonidos de advertencia a través de un mismo programa. Dentro de esta estructura, se define un caso para cada señal de advertencia. A su vez, en cada caso, se define la ruta hasta el programa encargado de transferir los datos a Excel. Los programas transferidores solamente son necesarios en las Pasadas Cortas.

La siguiente parte tiene como función realizar la transferencia de datos para las Pasadas Largas. No existen programas transferidores para estos ensayos, y solamente existe una toma de datos para cada sonido de advertencia.

```
%% LISTADO PASADAS LARGAS.  
case 2 %Operaciones para opción Pasadas Largas.  
switch ws %Estructura de control switch que realiza las funciones que se definan para cada warning sound de Pasadas Cortas.  
case 'P1' %Operaciones para warning sound P1.  
    filename='..\Tarea 1\Pasadaslargas\P1\p1.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.  
    writetable(tabla,filename,'Sheet','Datos Ensayo P1','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.  
  
    textbandf=('Banda Tercio de Octava','dBA'); %Designación del nombre a las columnas de los datos en la hoja de cálculo.  
    P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.  
    bandf=[F:P1]; %Almacenamiento del espectro de frecuencias en variable.  
    hojal='Espectro frecuencia P1'; %Establecimiento de nombre la hoja.  
    xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.  
    xlswrite(filename,textbandf,hojal); %Transferencia del encabezado de las columnas.  
    xlswrite(filename,bandf,hojal,xlRange); %Transferencia de los datos del espectro de frecuencias.  
  
case 'P2' %Operaciones para warning sound P1.  
    filename='..\Tarea 1\Pasadaslargas\P2\p2.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.  
    writetable(tabla,filename,'Sheet','Datos Ensayo P2','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.  
  
    textbandf=('Banda Tercio de Octava','dBA'); %Designación del nombre a las columnas de los datos en la hoja de cálculo.  
    P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.  
    bandf=[F:P1]; %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.  
    hojal='Espectro frecuencia P2'; %Establecimiento de nombre la hoja.  
    xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.  
    xlswrite(filename,textbandf,hojal); %Transferencia del encabezado de las columnas.  
    xlswrite(filename,bandf,hojal,xlRange); %Transferencia de los datos del espectro de frecuencias.
```

```
case 'P3' %Operaciones para warning sound P3.

filename='..\Tarea 1\Pasadaslargas\P3\p3.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P3','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F:P1]'; %Almacenamiento del espectro de frecuencias en variable.
hojal='Espectro frecuencia P3'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hojal); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hojal,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 'P4' %Operaciones para warning sound P4.

filename='..\Tarea 1\Pasadaslargas\P4\p4.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P4','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F:P1]'; %Almacenamiento del espectro de frecuencias en variable.
hojal='Espectro frecuencia P4'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hojal); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hojal,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 'P5' %Operaciones para warning sound P5.

filename='..\Tarea 1\Pasadaslargas\P5\p5.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P5','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F:P1]'; %Almacenamiento del espectro de frecuencias en variable.
hojal='Espectro frecuencia P5'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hojal); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hojal,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 'P8' %Operaciones para warning sound P8.

filename='..\Tarea 1\Pasadaslargas\P8\p8.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P8','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F:P1]'; %Almacenamiento del espectro de frecuencias en variable.
hojal='Espectro frecuencia P8'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hojal); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hojal,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 'P9' %Operaciones para warning sound P9.

filename='..\Tarea 1\Pasadaslargas\P9\p9.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P9','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F:P1]'; %Almacenamiento del espectro de frecuencias en variable.
hojal='Espectro frecuencia P9'; %Posición donde colocar el espectro de frecuencia.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hojal); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hojal,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 'P10' %Operaciones para warning sound P10.

filename='..\Tarea 1\Pasadaslargas\P10\p10.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P10','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F:P1]'; %Almacenamiento del espectro de frecuencias en variable.
hojal='Espectro frecuencia P10'; %Posición donde colocar el espectro de frecuencia.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hojal); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hojal,xlRange); %Transferencia de los datos del espectro de frecuencias.
```

```
case 'P11' %Operaciones para warning sound P11.

filename='..\Tarea 1\Pasadaslargas\P11\p11.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P11','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf=('Banda Tercio de Octava','dBA'); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F:P1]; %Almacenamiento del espectro de frecuencias en variable.
hojal='Espectro frecuencia P11'; %Posición donde colocar el espectro de frecuencia.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hojal); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hojal,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 'P12' %Operaciones para warning sound P12.

filename='..\Tarea 1\Pasadaslargas\P12\p12.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P12','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf=('Banda Tercio de Octava','dBA'); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F:P1]; %Almacenamiento del espectro de frecuencias en variable.
hojal='Espectro frecuencia P12'; %Posición donde colocar el espectro de frecuencia.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hojal); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hojal,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 'P13' %Operaciones para warning sound P13.

filename='..\Tarea 1\Pasadaslargas\P13\p13.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P13','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf=('Banda Tercio de Octava','dBA'); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F:P1]; %Almacenamiento del espectro de frecuencias en variable.
hojal='Espectro frecuencia P13'; %Posición donde colocar el espectro de frecuencia.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hojal); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hojal,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 'P14' %Operaciones para warning sound P14.

filename='..\Tarea 1\Pasadaslargas\P14\p14.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P14','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf=('Banda Tercio de Octava','dBA'); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
bandf=[F:P1]; %Almacenamiento del espectro de frecuencias en variable.
hojal='Espectro frecuencia P14'; %Posición donde colocar el espectro de frecuencia.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hojal); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hojal,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 'P16' %Operaciones para warning sound P16.

filename='..\Tarea 1\Pasadaslargas\P16\p16.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P16','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf=('Banda Tercio de Octava','dBA'); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
bandf=[F:P1]; %Almacenamiento del espectro de frecuencias en variable.
hojal='Espectro frecuencia P16'; %Posición donde colocar el espectro de frecuencia.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hojal); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hojal,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 'P17' %Operaciones para warning sound P17.

filename='..\Tarea 1\Pasadaslargas\P17\p17.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P17','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf=('Banda Tercio de Octava','dBA'); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
bandf=[F:P1]; %Almacenamiento del espectro de frecuencias en variable.
hojal='Espectro frecuencia P17'; %Posición donde colocar el espectro de frecuencia.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hojal); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hojal,xlRange); %Transferencia de los datos del espectro de frecuencias.
```

```
case 'P18' %Operaciones para warning sound P18.

filename='..\Tarea 1\Pasadaslargas\P18\p18.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P18','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf=('Banda Tercio de Octava','dBA'); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
bandf=[F;P1]; %Almacenamiento del espectro de frecuencias en variable.
hojal='Espectro frecuencia P18'; %Posición donde colocar el espectro de frecuencia.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hojal); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hojal,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 'P19' %Operaciones para warning sound P19.

filename='..\Tarea 1\Pasadaslargas\P19\p19.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P19','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf=('Banda Tercio de Octava','dBA'); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
bandf=[F;P1]; %Almacenamiento del espectro de frecuencias en variable.
hojal='Espectro frecuencia P19'; %Posición donde colocar el espectro de frecuencia.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hojal); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hojal,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 'P20' %Operaciones para warning sound P20.

filename='..\Tarea 1\Pasadaslargas\P20\p20.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P20','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf=('Banda Tercio de Octava','dBA'); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
bandf=[F;P1]; %Almacenamiento del espectro de frecuencias en variable.
hojal='Espectro frecuencia P20'; %Posición donde colocar el espectro de frecuencia.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hojal); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hojal,xlRange); %Transferencia de los datos del espectro de frecuencias.

end %Cierre de la estructura de control switch para códigos de Pasadas Cortas.

end %Cierre de la estructura de control switch para tipos de pasadas.
```

Figura 4.37. Estructura para Pasadas Largas en el programa selectorexc.

Como en el anterior bloque, se define un “switch”, estableciendo un caso para cada sonido de advertencia estudiado. Dentro de cada caso, se define la ruta donde se aloja el archivo Excel. En segundo lugar, se transfiere a este archivo la tabla obtenida en “info”.

A continuación, se concretan los encabezados de las columnas que se dispondrán en la hoja de Excel. Posteriormente, se redondean a una cifra decimal los valores de presión sonora de cada banda en tercios de octava y se define una variable que almacene el espectro de frecuencias. Seguidamente, se establece el nombre de la hoja donde se guardan los datos y la celda a partir de la cual se dispondrá la información. Finalmente, se realiza la exportación del archivo de los encabezados y el espectro de frecuencias.

#### 4.4.1.12. CONJUNTO DE PROGRAMAS VELEXC + SONIDO DE ADVERTENCIA

Este apartado, se basa en la explicación de los programas tipo “velexc + sonido de advertencia”. Por ejemplo, para el sonido de advertencia E4, el programa se llama velexc4. Estos programas se encargan de transferir los datos al documento Excel para las Pasadas Cortas. Este tipo de programas guarda una clara similitud con el segundo bloque del programa “selectorexc”, donde la transferencia se realiza con las Pasadas Largas.

```
%% LISTADO VELOCIDADES.
switch v %Estructura de control switch que realiza las funciones que se definan para cada caso de velocidad para warning sound E1.

case 1 %Operaciones para velocidad de 5 km/h.

filename='..\E1\Cl E1 5\velcinco.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo 5km','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf=('Banda Tercio de Octava','dBA'); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F:P1]; %Almacenamiento del espectro de frecuencias en variable.
hoja1='Espectro frecuencia e15uno'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hoja1); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.

textbandf=('Banda Tercio de Octava','dBA'); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P2=round(P2,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F:P2]; %Almacenamiento del espectro de frecuencias en variable.
hoja2='Espectro frecuencia e15dos'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hoja2); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja2,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 2 %Operaciones para velocidad de 10 km/h.

filename='..\E1\Cl E1 10\eldiez.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo 10km','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf=('Banda Tercio de Octava','dBA'); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava
bandf=[F:P1]; %Almacenamiento del espectro de frecuencias en variable.
hoja1='Espectro frecuencia e10uno'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hoja1); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 3 %Operaciones para velocidad de 15 km/h.

filename='..\E1\Cl E1 15\elquince.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo 15km','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf=('Banda Tercio de Octava','dBA'); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava
bandf=[F:P1]; %Almacenamiento del espectro de frecuencias en variable.
hoja1='Espectro frecuencia e115uno'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hoja1); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.

textbandf=('Banda Tercio de Octava','dBA'); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P2=round(P2,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F:P2]; %Almacenamiento del espectro de frecuencias en variable.
hoja2='Espectro frecuencia e115dos'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hoja2); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja2,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 4 %Operaciones para velocidad de 20 km/h.

filename='..\E1\Cl E1 20\elveinte.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo 20km','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf=('Banda Tercio de Octava','dBA'); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F:P1]; %Almacenamiento del espectro de frecuencias en variable.
hoja1='Espectro frecuencia e120uno'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hoja1); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.

textbandf=('Banda Tercio de Octava','dBA'); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P2=round(P2,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F:P2]; %Almacenamiento del espectro de frecuencias en variable.
hoja2='Espectro frecuencia e120dos'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hoja2); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja2,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 5 %Operaciones para velocidad de 25 km/h.

filename='..\E1\Cl E1 25\elveinticinco.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo 25km','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf=('Banda Tercio de Octava','dBA'); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F:P1]; %Almacenamiento del espectro de frecuencias en variable.
hoja1='Espectro frecuencia e125uno'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hoja1); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.

textbandf=('Banda Tercio de Octava','dBA'); %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P2=round(P2,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F:P2]; %Almacenamiento del espectro de frecuencias en variable.
hoja2='Espectro frecuencia e125dos'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hoja2); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja2,xlRange); %Transferencia de los datos del espectro de frecuencias.
```

```

case 6 %Operaciones para velocidad de 30 km/h.

filename='..\E1\Cl E1 30\eltreinta.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo 30km','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.

textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf={F>P1}'; %Almacenamiento del espectro de frecuencias en variable.
hoja1='Espectro frecuencia e130uno'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hoja1); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.

textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P2=round(P2,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf={F>P2}'; %Almacenamiento del espectro de frecuencias en variable.
hoja2='Espectro frecuencia e130dos'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hoja2); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja2,xlRange); %Transferencia de los datos del espectro de frecuencias.

end %Cierre de la estructura de control switch para las velocidades del warning sound E1.
    
```

Figura 4.38. Programa velexcel.

En este programa, también basado en una estructura de control “switch”, existe un caso para cada velocidad de circulación. Dentro de cada caso, se define en primer lugar la ruta de acceso a la hoja de cálculo, transfiriéndose la tabla obtenida en “info”.

La siguiente parte dentro de cada caso, tendrá tantos bloques que transfieran espectros de frecuencia como mediciones se realizaron a esa velocidad con el sonido de advertencia que se esté estudiando. Todos estos bloques siguen la misma estructura. Primero se designan los encabezados de las columnas donde se insertará la información. Seguidamente se redondea el nivel de presión sonora en dB(A) a una cifra decimal y se crea una variable que almacena el espectro de frecuencias, que llevará asignada el número de la pasada. A continuación, se define el nombre de la hoja donde se transfiere el espectro y la posición donde se empezarán a desplegar los datos. Por último, se procede a la exportación de los encabezados de las columnas y los datos del espectro de frecuencias de cada pasada a su respectiva hoja.

La ejecución del programa devolvería el siguiente resultado:

<b>Recording</b>	<b>dBA</b>	<b>Instant</b>	<b>Total</b>
<b>e15uno.m</b>	51,9	6,57	11
<b>e15dos.m</b>	52	5,59	11

Figura 4.39. Tabla transferida a Excel a través del programa velexcel.



Banda Tercio de Octava	dBA	Banda Tercio de Octava	dBA
100	31,8	100	26,9
125	32,7	125	29,8
160	35,4	160	32
200	40,8	200	37,4
250	34,6	250	30,6
315	33,1	315	32,9
400	34,7	400	36,5
500	47,7	500	50,7
630	35,7	630	36,4
800	33,6	800	35,3
1000	37,5	1000	35,1
1250	38,5	1250	32,6
1600	38	1600	32,6
2000	39,1	2000	37,4
2500	32,6	2500	28,2
3150	27,9	3150	26,2
4000	29,9	4000	25,5
5000	32,6	5000	25,4
1600	38	1600	32,6
2000	39,1	2000	37,4
2500	32,6	2500	28,2
3150	27,9	3150	26,2
4000	29,9	4000	25,5
5000	32,6	5000	25,4

Figura 4.40. Espectros de frecuencia transferidos a Excel a través del programa velexcel.

Con esto concluye la explicación de cada uno de los programas generados a lo largo de este apartado. Para ofrecer una visión general del modo en el que el conjunto de los programas se interrelaciona, se ofrece en la Figura y la Figura , dos esquemas donde esto queda reflejado.

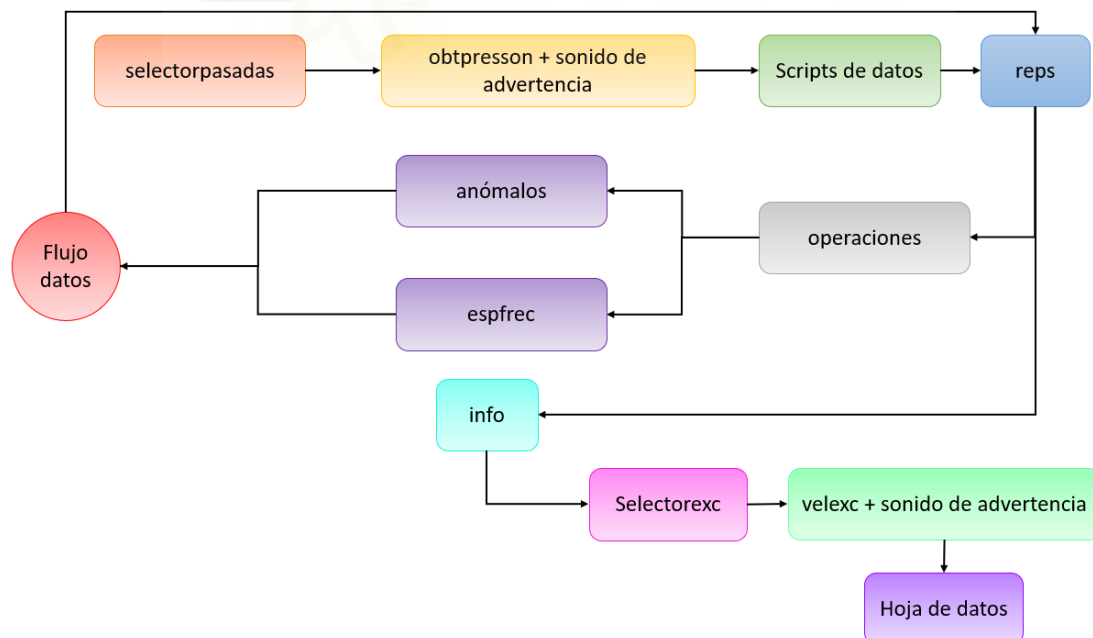


Figura 4.41. Esquema relacional para Pasadas Cortas.

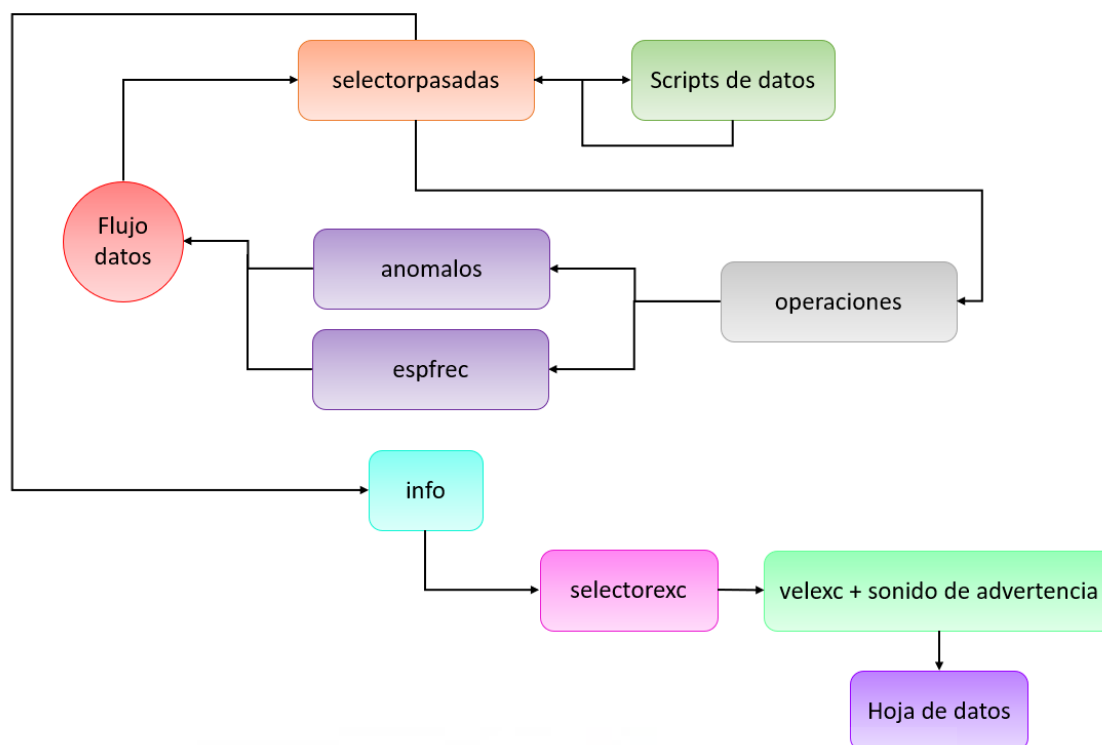


Figura 4.42. Esquema relacional para Pasadas Largas.

#### 4.4.2. DIRECTIVIDAD VERTICAL

Como continuación del trabajo, se lleva a cabo un estudio de la directividad vertical de cada ensayo.

Los recursos de partida fueron:

- Imágenes del alzado del lateral del coche, para cada una de las pasadas realizadas. En estas imágenes se representaban los niveles de presión sonora en función de la posición alrededor del coche.
- Archivos con extensión .txt, donde se definen los niveles de presión sonora en Pa alrededor del coche en la pasada y un vector de longitud considerable .

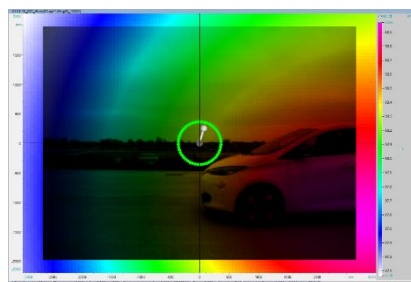


Figura 4.43. Imagen donde se representan el nivel de presión sonora en función de la posición alrededor del coche.

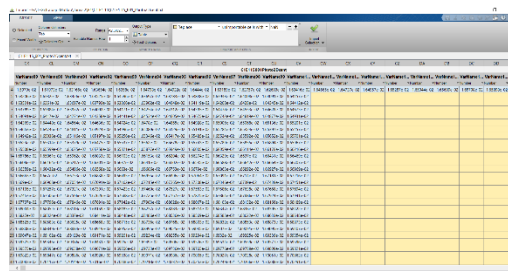


Figura 4.44. Datos contenidos en el archivo de texto.

#### 4.4.2.1. OPERACIONES DE PARTIDA

La primera tarea que se realizó fue la extracción de los datos útiles (o sea, la matriz de presión sonora en Pa). Para ello, se importó el archivo de texto a Matlab, se seleccionó la matriz necesaria y se guardó en una variable que se denominó “tabbppal”. El almacenamiento de la matriz de presión sonora en la variable anterior, se realizó para cada una de las mediciones que se realizaron durante los ensayos.

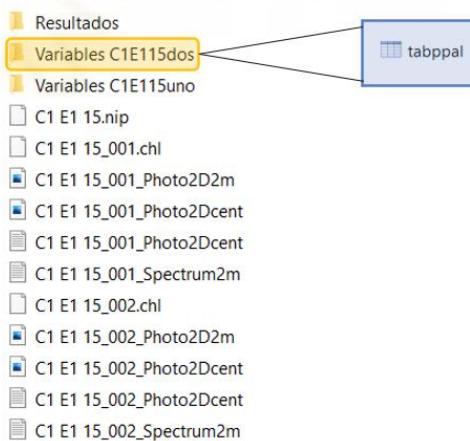
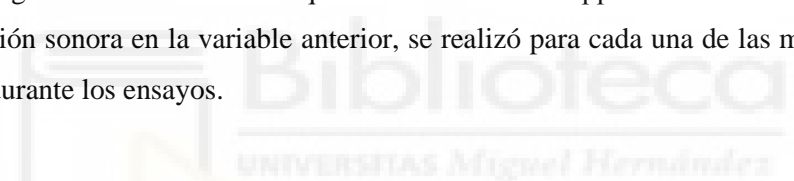


Figura 4.45. Distribución de las variables de datos.

#### 4.4.2.2. PROGRAMA PLOTTRIDIMENSIONAL

Tras obtener los datos, se prosigue verificando los datos que presenta la matriz, para asegurar que no se presentaban datos erróneos. Para ello, se creó un programa que representara los datos en 3D, llamado “plottridimensional”.

Para empezar, el programa determina si la pasada analizada requiere la eliminación de la primera fila. Esta fila servía de control, y en el caso de no eliminarse, se producirían errores en el futuro. A continuación, se calcula el tamaño de la matriz de presión sonora útil en Pa (matriz sin esa línea de control).

```
%% CÁLCULO DE DIMENSIONES A EMPLEAR.
if (pas==1 & ws==9 & v=1 & con==1) %Primera condición donde se define donde no es necesario arreglar la matriz de presión sonora.

    tabppalutil=tabppal; %Almacenamiento de la parte útil de la matriz de presión sonora en variable.
    tam=size(tabppalutil); %Cálculo del tamaño de la matriz de presión sonora.

else %Segunda condición donde se establece el procedimiento para el resto de casos.

    tambruto=size(tabppal); %Cálculo del tamaño de la matriz de presión sonora.
    tamxbruto=tambruto(2); %Determinación dimensión x de la matriz de presión sonora.
    tamybruto=tambruto(1); %Determinación dimensión y de la matriz de presión sonora.
    vectutil=[2:1:tamybruto]; %Eliminación de la primera fila de la matriz de presión sonora,
    tabppalutil=tabppal(vectutil,:); %Almacenamiento de la parte útil de la matriz de presión sonora en variable.
    tam=size(tabppalutil); % Cálculo del tamaño de la matriz de presión sonora.

end %Cierre de condiciones.

tamx=tam(2); %Determinación del tamaño en dimensión X de la matriz de presión sonora.
tamy=tam(1); %Determinación del tamaño en dimensión Y de la matriz de presión sonora.
```

Figura 4.46. Eliminación de la línea de control y determinación del tamaño de la matriz de presión sonora útil en el programa plottridimensional.

Seguidamente, y en función de los diferentes casos, se definen los vectores de los ejes X e Y. La mayor discordancia entre dimensiones se da entre sí al trabajar con Pasadas Cortas ( $pas==1$ ) o Largas ( $pas==2$ ). Para llevar a cabo la definición del vector en una dimensión (X o Y), se definen sus extremos con los extremos que quedan establecidos en los ejes de las imágenes proporcionadas, restándole una unidad al último elemento del vector (para evitar que el vector sobrepase la dimensión necesaria). A continuación, se genera el paso del vector. Para ello, se suman los extremos del eje de la imagen y se divide la suma entre el tamaño de la matriz de presión sonora en esa dimensión.

```
%% DETERMINACIÓN DE LOS VALORES DE LOS EJES.
if (pas==2 & ws==1) %Primera condición donde se designan los vectores X e Y a usar según el caso.

    X=[-1100:(2200/tamx):1099]; %Definición longitud y valores del eje X.
    Y=[800:(-1600/tamy):-799]; %Definición longitud y valores del eje Y.

else if (pas==2) %Segunda condición donde se designan los vectores X e Y a usar según el caso.

    X=[-1100:(2200/tamx):1099]; %Definición longitud y valores del eje X.
    Y=[900:(-1800/tamy):-899]; %Definición longitud y valores del eje Y.

else %Tercera condición donde se designan los vectores X e Y a usar según el caso..

    X=[-3000:(6000/tamx):2999]; %Definición longitud y valores del eje X.
    Y=[2200:(-4400/tamy):-2199]; %Definición longitud y valores del eje Y.

end %Cierre de la segunda condición.

end %Cierre de condiciones.
```

Figura 4.47. Designación de los ejes X e Y en el programa plottridimensional.

A continuación, se transforman los valores de la matriz de presión sonora de Pa a dB, según la ecuación (1).

```
%% TRANSFORMACIÓN DE LOS VALORES DE LA MATRIZ DE PRESION SONORA DE Pa a dB.  
tabppalutil2=tabppalutil.^2; %Elevación de la matriz de presión sonora al cuadrado.  
dbtabppalutil=10*log10(tabppalutil2/(20*10^(-6))^2); %Transformación de los valores de presión sonora de Pa a dB.  
  
%% DETERMINACIÓN DEL MÁXIMO Y MÍNIMO NIVEL DE PRESIÓN SONORA.  
dbmax=max(max(dbtabppalutil)); %Cálculo del valor de presión sonora en dB máximo.  
dbmin=min(min(dbtabppalutil)); %Cálculo del valor de presión sonora en dB mínimo.
```

Figura 4.48. Transformación de los datos a dB y obtención del valor máximo y mínimo de presión sonora en el programa plottridimensional.

Tras esto, se determinan cuáles son los valores máximos y mínimos en dicha matriz. Como Matlab, al hacer el máximo/mínimo de una matriz devuelve un vector fila que contiene el valor máximo/mínimo de cada columna, se vuelve a realizar el máximo/mínimo con ese vector.

Para continuar, se define el punto de referencia de partida a partir de donde comienza el estudio. Debido a la convergencia de distintos ruidos (rodadura, aerodinámica, sonido de advertencia, etc.) se adoptó un punto localizado aproximadamente en la parte central de la primera puerta para las Pasadas Cortas.

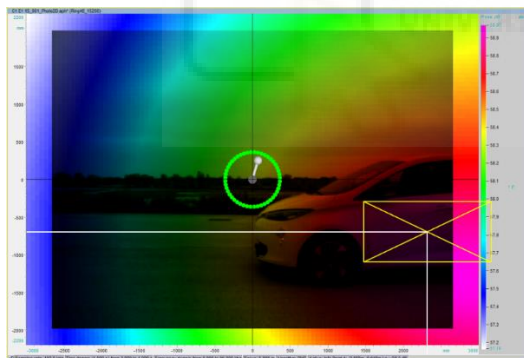


Figura 4.49. Punto inicial de trabajo para Pasadas Cortas con sentido de marcha a izquierda.

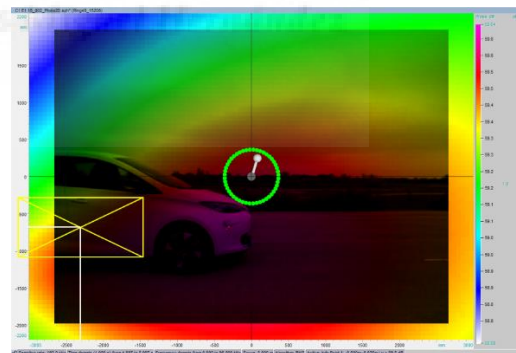


Figura 4.50. Punto inicial de trabajo para Pasadas Cortas con sentido de marcha a derecha.

Para las Pasadas Largas, en cambio, el punto de referencia de partida se encuentra ligeramente separado del morro.

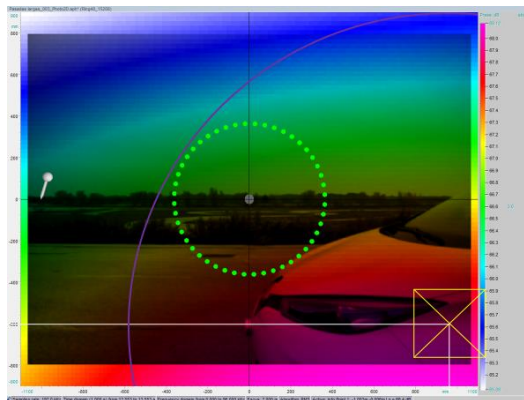


Figura 4.51. Punto inicial de trabajo para Pasadas Largas con sentido de marcha a izquierda.

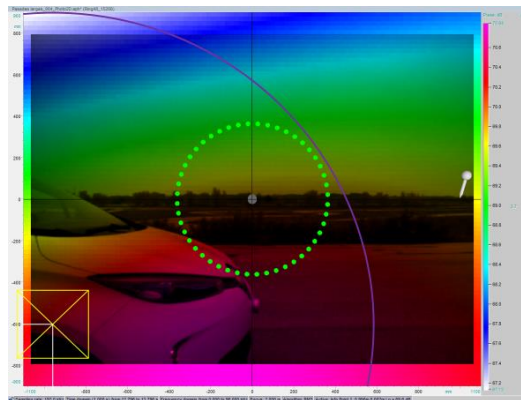


Figura 4.52. Punto inicial de trabajo para Pasadas Largas con sentido de marcha a izquierda.

Se recuerda que los parámetros modificados en los ensayos fueron el tipo de pasadas, el sonido de advertencia analizado, la velocidad o el sentido de marcha. En función de estos parámetros se definen las condiciones para cada caso específico. Dentro de cada caso, se definirán los puntos de referencia asignados en la imagen. A su vez, se trasladan los datos de la matriz de presión sonora en Pa útiles a una variable para trabajar con ella posteriormente.

```
%% DESIGNACIÓN DEL PUNTO DE REFERENCIA.
if (ws==1 & v==1 & con==1) %Primera condición donde se designa el punto de referencia a emplear según el caso.

    Lppa=tabppalutil; %Almacenamiento de la parte útil de la matriz de presión sonora en variable.
    photox=2320; %Componente X del punto de referencia de la imagen.
    photoy=-700; %Componente Y del punto de referencia de la imagen.
    posx=find(photox<=X & photox+30>=X); %Búsqueda de la posición de dicho punto en el vector X.
    posy=find(photoy=Y & photoy-40<=Y); %Búsqueda de la posición de dicho punto en el vector Y.
    cx=X(posx); %Valor del punto encontrado del vector X.
    cy=Y(posy); %Valor del punto encontrado del vector Y.

else if (pas==1 & (con==1 | con==3)) %Segunda condición donde se designa el punto de referencia a emplear según el caso.

    Lppa=tabppalutil; %Almacenamiento de la parte útil de la matriz de presión sonora en variable.
    photox=2320; %Componente X del punto de referencia de la imagen.
    photoy=-690; %Componente Y del punto de referencia de la imagen.
    posx=find(photox<=X & photox+20>=X); %Búsqueda de la posición de dicho punto en el vector X.
    posy=find(photoy=Y & photoy-20<=Y); %Búsqueda de la posición de dicho punto en el vector Y.
    cx=X(posx); %Valor del punto encontrado del vector X.
    cy=Y(posy); %Valor del punto encontrado del vector Y.

else if (pas==1 & (con==2 | con==4)) %Tercera condición donde se designa el punto de referencia a emplear según el caso.

    Lppa=tabppalutil; %Almacenamiento de la parte útil de la matriz de presión sonora en variable.
    photox=-2320; %Componente X del punto de referencia de la imagen.
    photoy=-690; %Componente Y del punto de referencia de la imagen.
    posx=find(photox>=X & photox-20<=X); %Búsqueda de la posición de dicho punto en el vector X.
    posy=find(photoy=Y & photoy-20<=Y); %Búsqueda de la posición de dicho punto en el vector Y.
    cx=X(posx); %Valor del punto de referencia encontrado en el vector X.
    cy=Y(posy); %Valor del punto de referencia encontrado en el vector Y.

else if (pas==2 & ws==1) %Cuarta condición donde se designa el punto de referencia a emplear según el caso.

    Lppa=tabppalutil; %Almacenamiento de la parte útil de la matriz de presión sonora en variable.
    photox=960; %Componente X del punto de referencia de la imagen.
    photoy=-600; %Componente Y del punto de referencia de la imagen.
    posx=find(photox<=X & photox+10>=X); %Búsqueda de la posición de dicho punto en el vector X.
    posy=find(photoy=Y & photoy-10<=Y); %Búsqueda de la posición de dicho punto en el vector Y.
    cx=X(posx); %Valor del punto de referencia encontrado en el vector X.
    cy=Y(posy); %Valor del punto de referencia encontrado en el vector Y.

else if (pas==2 & mod(ws,2)~=0) %Quinta condición donde se designa el punto de referencia a emplear según el caso.

    Lppa=tabppalutil; %Almacenamiento de la parte útil de la matriz de presión sonora en variable.
    photox=960; %Componente X del punto de referencia de la imagen.
    photoy=-600; %Componente Y del punto de referencia de la imagen.
    posx=find(photox<=X & photox+10>=X); %Búsqueda de la posición de dicho punto en el vector X.
    posy=find(photoy<=Y & photoy+10>=Y); %Búsqueda de la posición de dicho punto en el vector Y.
    cx=X(posx); %Valor del punto de referencia encontrado en el vector X.
    cy=Y(posy); %Valor del punto de referencia encontrado en el vector Y.

else %Sexta condición donde se designa el punto de referencia a emplear para el resto de casos.

    Lppa=tabppalutil; %Almacenamiento de la parte útil de la matriz de presión sonora en variable.
    photox=-960; %Componente X del punto de referencia de la imagen.
    photoy=-600; %Componente Y del punto de referencia de la imagen.
    posx=find(photox>=X & photox-10<=X); %Búsqueda de la posición de dicho punto en el vector X.
    posy=find(photoy<=Y & photoy+10>=Y); %Búsqueda de la posición de dicho punto en el vector Y.
    cx=X(posx); %Valor del punto de referencia encontrado en el vector X.
    cy=Y(posy); %Valor del punto de referencia encontrado en el vector Y.

end %Cierre de la quinta condición.

end %Cierre de la cuarta condición.

end %Cierre de la tercera condición.

end %Cierre de la segunda condición.

end %Cierre de condiciones.
```

Figura 4.53. Obtención del punto de referencia según las condiciones elegidas en el programa plottridimensional.

Tras definir los puntos de referencia de partida en la imagen, se procede a adaptarlos para que coincidan con los puntos establecidos en los vectores X e Y definidos previamente. Para ello, se usa el comando “find”, que se encarga de buscar la posición de los puntos con valores que

cumplan una serie de condiciones. Se estableció que una variación del punto de referencia de la imagen de entre 10 y 40 unidades era aceptable (teniendo en cuenta que la unidad de trabajo es el milímetro). Para conseguir el punto de referencia de partida con el que trabajar, solamente queda buscar dichos valores en el eje X e Y.

El siguiente paso, consiste en comprobar si los datos que las matrices de presión sonora representaban eran correctos o no.

```
%% REPRESENTACIÓN EN 3D.
figure %Representación del gráfico en una ventana independiente.

surf(X,Y,dbtabppalutil) %Graficación en 3D de la matriz de los datos.
shading flat %Cambio en la textura de la representación 3D.
colormap(hsv); %Cambio en el espectro de colores en la representación del gráfico 3D.

title('PRESIÓN SONORA (dB)') %Colocación de título para el gráfico 3D.
xlabel('Eje X (mm)') %Colocación de etiqueta del eje X para el gráfico 3D.
ylabel('Eje Y (mm)') %Colocación de etiqueta del eje Y para el gráfico 3D.
xlim([X(1) X(length(X))]) %Definición de los límites del eje X para el gráfico 3D.
ylim([Y(length(Y)) Y(1)]) %Definición de los límites del eje Y para el gráfico 3D.

c=colorbar; %Aplicación de barra de definición de valores numéricos para el espectro de colores.
c.Limits = [dbmin dbmax]; %Definición de los valores máximos y mínimos de dicha barra.
caxis([dbmin dbmax]); %Definición de los límites de dicha barra.
c.Label.String = 'Presión Sonora en dB'; %Colocación de etiqueta para barra.
```

Figura 4.54. Representación 3D de la matriz de presión sonora en el programa plottridimensional.

Para ello, la mejor opción es el comando “surf”, que permite graficar en 3D con una textura suave. Además, el comando “shading” permitía suavizar más aún la textura. Tras esto, se cambió el espectro de colores para que coincidiese con las imágenes proporcionadas.

El siguiente bloque dispone los títulos y etiquetas de los ejes en el gráfico. También se definen los límites de los ejes X e Y, para que no aparezcan espacios en los extremos.

Para finalizar este bloque, se define una barra donde se registran los niveles de presión sonora y su relación con el espectro de colores. Además, se definen los límites de dicha barra de forma paramétrica y se dispone una etiqueta en la misma.

Para finalizar con el programa, se dispone el punto de referencia en el gráfico 3D, proporcionando una mayor estética a la representación.



```
%% GRAFICACIÓN DEL PUNTO DE ESTUDIO.
if (pas==1) %Primera condición donde se grafica el punto de referencia empleado según el caso.

    puntestX=num2str(cx,4); %Transformación de datos numéricos a array de caracteres del elemento en X del punto de referencia.
    puntestY=num2str(cy,5); %Transformación de datos numéricos a array de caracteres del elemento en Y del punto de referencia.
    txt1=['\bf X = ',puntestX,' mm']; %Definición del texto para marcador de posición X.
    texto1=text(cx+60,cy+170,dbmax,txt1); %Designación de la posición del texto para marcador de posición X.
    txt2=['\bf Y = ',puntestY,' mm']; %Definición del texto para marcador de posición Y.
    texto2=text(cx+60,cy+70,dbmax,txt2); %Designación de la posición del texto para marcador de posición Y.
    texto3=text(cx,cy,dbmax,'\bf \bullet'); %Colocación de símbolo cruz contenida en círculo que marca el punto de estudio.
    texto3.FontSize=15; %Aumento del tamaño del símbolo cruz contenida en círculo.

else %Segunda condición donde se grafica el punto de referencia empleado para el resto de casos.

    puntestX=num2str(cx,4); %Transformación de datos numéricos a array de caracteres el elemento en X del punto de referencia.
    puntestY=num2str(cy,5); %Transformación de datos numéricos a array de caracteres el elemento en Y del punto de referencia.
    txt1=['\bf X = ',puntestX,' mm']; %Definición del texto para marcador de posición X.
    texto1=text(cx-100,cy+80,dbmax,txt1); %Designación de la posición del texto para marcador de posición X.
    txt2=['\bf Y = ',puntestY,' mm']; %Definición del texto para marcador de posición Y.
    texto2=text(cx-100,cy+30,dbmax,txt2); %Designación de la posición del texto para marcador de posición Y.
    texto3=text(cx,cy,dbmax,'\bf \bullet'); %Colocación de símbolo de cruz contenida en círculo que marca el punto de estudio.
    texto3.FontSize=15; %Aumento del tamaño del símbolo cruz contenida en círculo.

end %Cierre de condiciones.

if (pas==1 & (con==1 | con==3)) %Primera condición donde se colorean los textos definidos anteriormente según el caso.

    texto1.Color='White'; %Definición de color del marcador de posición X.
    texto2.Color='White'; %Definición de color del marcador de posición Y.
    texto3.Color='White'; %Definición de color del símbolo de cruz contenida en círculo.

else if (pas==1 & (con==2 | con==4)) %Segunda condición donde se colorean los textos definidos anteriormente según el caso.

    texto1.Color='Black'; %Definición de color del marcador de posición X.
    texto2.Color='Black'; %Definición de color del marcador de posición Y.
    texto3.Color='Black'; %Definición de color del símbolo de cruz contenida en círculo.

else if (pas==2 & mod(ws,2)~=0) %Tercera condición donde se colorean los textos definidos anteriormente según el caso.

    texto1.Color='White'; %Definición de color del marcador de posición X.
    texto2.Color='White'; %Definición de color del marcador de posición Y.
    texto3.Color='White'; %Definición de color del símbolo de cruz contenida en círculo.

else %Cuarta condición donde se colorean los textos definidos anteriormente para el resto de casos.

    texto1.Color='Black'; %Definición de color del marcador de posición X.
    texto2.Color='Black'; %Definición de color del marcador de posición Y.
    texto3.Color='Black'; %Definición de color del símbolo de cruz contenida en círculo.

end %Cierre de la tercera condición.

end %Cierre de la segunda condición.

end %Cierre de condiciones.
```

Figura 4.55. Colocación del punto de referencia en el gráfico 3D en el programa plottridimensional.

Básicamente, en esta parte se distingue primeramente entre Pasadas Largas y Cortas. Después, se transforman los valores numéricos a array de caracteres, se define el texto a disponer y su posición en el gráfico. Por último, se designa con qué color representarlo, en función de las condiciones establecidas.

Al ejecutar el programa, para un caso determinado, Matlab proporciona la siguiente salida.

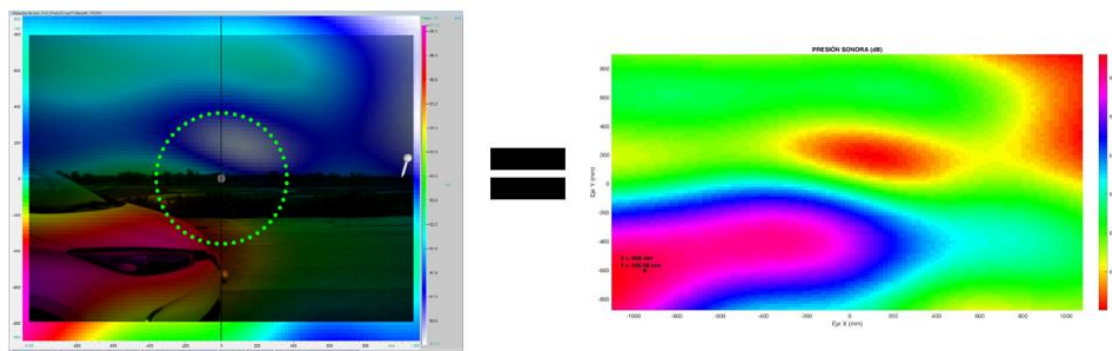


Figura 4.56. Comparación del resultado del programa plottridimensional con la imagen de la medición estudiada.

Por tanto, se puede verificar el correcto funcionamiento del programa. Aunque los colores de Matlab no sean los mismos que la imagen obtenida a través de la cámara acústica, las áreas representadas sí coinciden en forma, tamaño y lugar. De esta forma, se concluye que los datos que proporciona la matriz de presión sonora son correctos, tras revisar todas las mediciones.

#### 4.4.2.3. PROGRAMAS DIRECTIVIDADIZQ Y DIRECTIVIDADDDCHA

Hasta este momento, se ha realizado la verificación de los datos. No obstante, no se ha tratado la directividad. Para estudiarla en profundidad, se crearon los programas “directividadizq” y “directividadddcha”, encargados de calcular los parámetros necesarios para determinar la directividad de los ensayos. La función que realizan ambos programas es la misma, aunque presentan ligeras variaciones uno respecto al otro. La razón de separar el estudio de la directividad en dos programas, fue la de simplificar los programas y facilitar su entendimiento.

Antes de explicar el funcionamiento del código, conviene describir el procedimiento para obtener los valores de la matriz de presión sonora a emplear para obtener información acerca de la directividad vertical de la medición.

Se parte del punto de referencia definido anteriormente. Con este punto como centro, se define un arco de radio determinado que se encuentre alrededor del coche y que seleccione aquellos puntos a estudiar. El radio del arco debía coincidir con la distancia resultante de unir la coordenada Y del sistema de referencia inicial con el punto más alejado en dicha dimensión. Esto queda visualmente explicado en la Figura 4.57 y Figura 4.58.

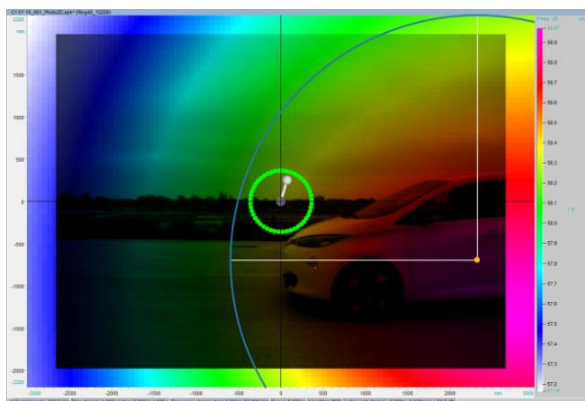


Figura 4.57. Arco generado en Pasadas Cortas.

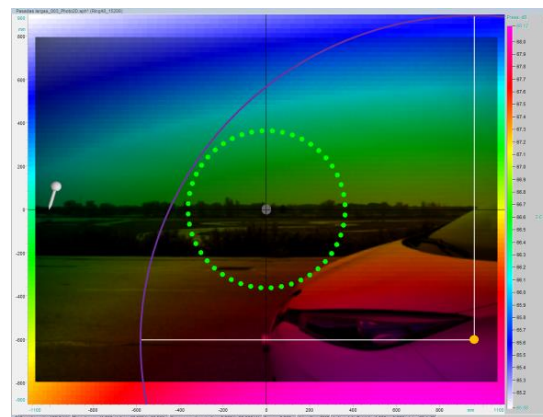


Figura 4.58. Arco generado en Pasadas Largas.

En el caso de Pasadas Cortas, no es necesario llevar a cabo un estudio del arco completo, ya que la porción por debajo de la banda de rodadura no sería perceptible por los transeúntes. Por ello, se determina un límite inferior, cuyos pasos de obtención serían:

1. Se proyecta una línea horizontal hasta la altura de la banda de rodadura del neumático del vehículo.

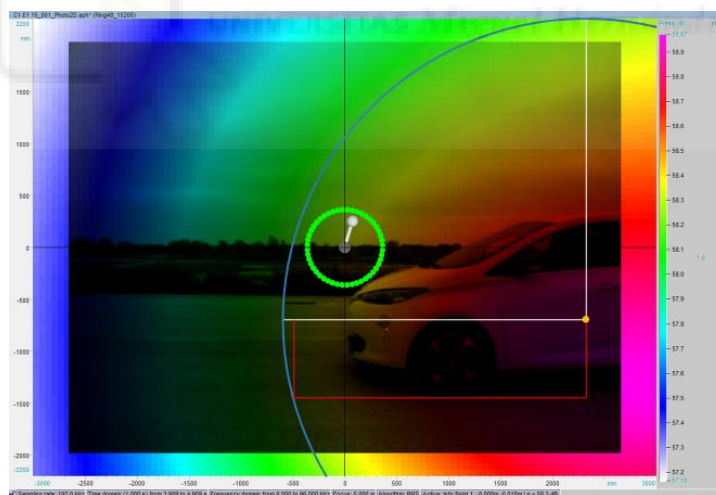


Figura 4.59. Proyección de línea horizontal hasta la banda de rodadura.

2. Se describe un radio que conecta el punto de referencia de partida con el extremo izquierdo de la línea proyectada.

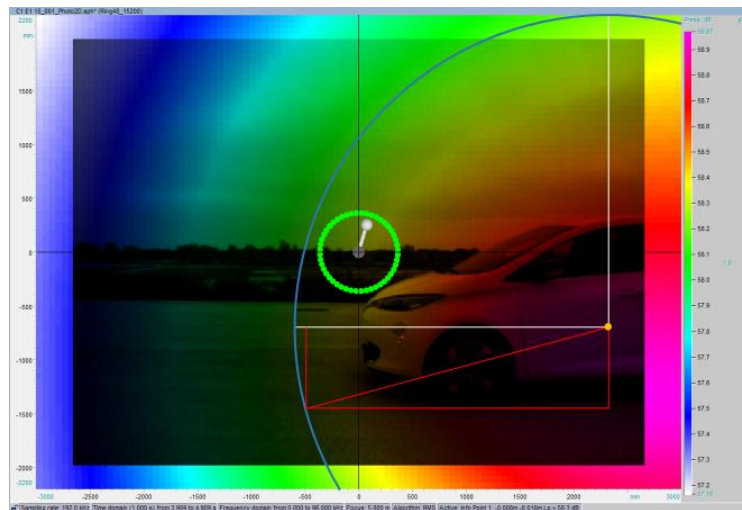


Figura 4.60. Generación del radio que existe entre el punto de referencia y la línea proyectada.

3. Finalmente, de las dos partes que se generan en el cuadrado, se escoge la superior y se describe el ángulo que define el punto máximo del arco hasta donde se debe analizar.

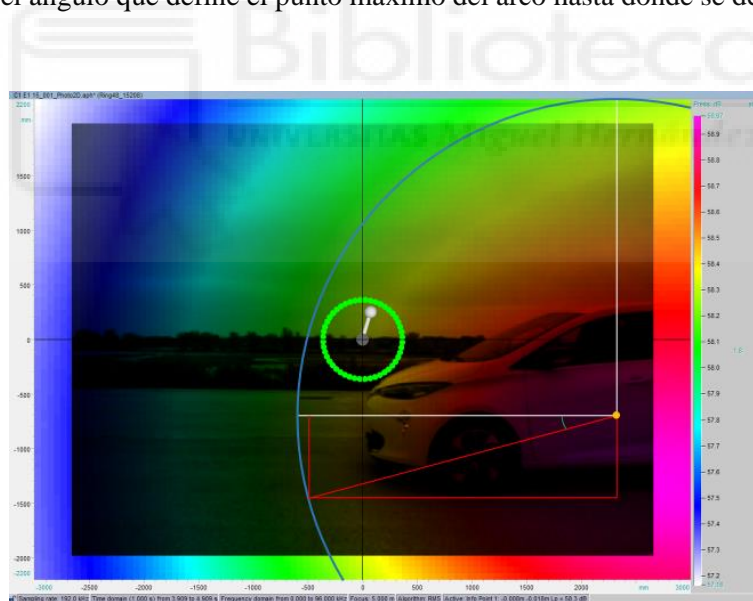


Figura 4.61. Definición del ángulo extra a analizar por debajo de  $0^\circ$ .

Por lo tanto, el arco final a emplear se refleja en la Figura 4.62.

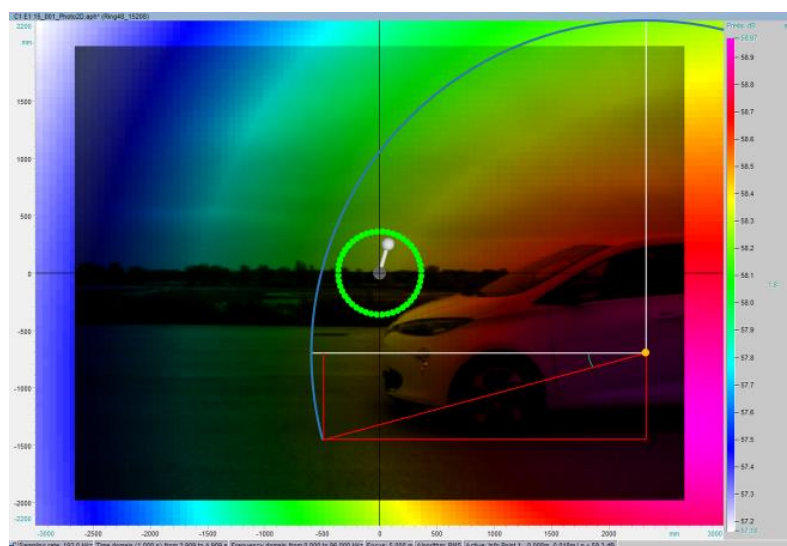


Figura 4.62. Arco resultante a estudiar en Pasadas Cortas.

Sin embargo, las operaciones explicadas anteriormente solamente deben llevarse a cabo cuando la medición es un caso de Pasada Corta. Para los casos de Pasadas Largas, el arco que se obtiene, primeramente, no se modifica. Esto es así porque las imágenes que la cámara acústica tomó en las Pasadas Cortas no presentaban las mismas dimensiones que las realizadas en las Pasadas Largas. En estas, se muestra una imagen donde la posición más baja apreciada es aproximadamente la horizontal que coincide con la mitad del morro, a la altura de la matrícula delantera tal como queda mostrado en la Figura 4.58.

Una vez conocido que datos de partida se van a emplear, se puede proceder a explicar ambos programas. Estos realizan, en primer lugar, un cambio del sistema de referencia. Su objetivo es obtener posiciones cuyo origen sea el punto de referencia definido anteriormente. De este modo, se podrá calcular qué distancia hay desde el origen al punto a estudiar de forma sencilla, que se comparará posteriormente con el radio.

```
%% DETERMINACIÓN DE NUEVO SISTEMA DE REFERENCIA.
```

```
Xm=(X-cx)/1000; %Cambio de los valores del eje X al nuevo sistema de referencia.
```

```
Ym=(Y-cy)/1000; %Cambio de los valores del eje y al nuevo sistema de referencia.
```

```
%% CÁLCULO DEL RADIO QUE SE GENERA DESDE EL ORIGEN DE NUESTRO SISTEMA DE REFERENCIA A CADA PUNTO.
for i=1:tamx %Primer bucle para recorrer el vector X nuevo de izquierda a derecha y crear nuevas variables.
    for j=1:tamy %Segundo bucle para recorrer el vector Y nuevo de arriba a abajo y crear nuevas variables.
        if Xm(i)==0 %Primera condición donde se establece el caso en el que el vector X nuevo sea 0.
            hipotenusa(j,i)=abs(Ym(j)); %Asignación de hipotenusa como valor absoluto del punto del vector Y nuevo.
        else if Ym(j)==0 %Segunda condición donde se establece el caso en el que el vector Y nuevo sea 0.
            hipotenusa(j,i)=abs(Xm(i)); %Asignación de hipotenusa el valor absoluto del punto del vector X nuevo.
        else %Tercera condición donde se establece el procedimiento para el resto de casos.
            hipotenusa(j,i)=sqrt(((Xm(i)).^2)+(Ym(j)).^2)); %Cálculo de hipotenusa donde catetos son valores de eje X e Y nuevos.
        end %Cierre de la segunda condición.
    end %Cierre de condiciones.
end %Cierre del segundo bucle.
end %Cierre del primer bucle.
```

Figura 4.63. Cambio de sistema de referencia y cálculo de distancias al nuevo origen en el conjunto directividadizq/directividaddcha.

A continuación, se desarrolló un bucle “for” anidado que recorre por cada punto del vector Y, todos los puntos del vector X. De esta forma se pasa por todas y cada una de las posiciones del plano XY. Seguidamente, se calcula la hipotenusa, estableciendo los distintos casos que se podrían dar a lo largo de la matriz. Los catetos para dicho calculo serán las distancias X e Y a dicho punto.

Si el punto estudiado presenta las dos distancias, la hipotenusa se obtendrá a partir del Teorema de Pitágoras. No obstante, habrá algunos puntos especiales, como los situados en la horizontal y vertical del sistema de referencia, donde no se contará con una de las distancias necesarias. Por ello, se definieron estos casos y como proceder en ellos. Si el punto se encuentra en la horizontal, la hipotenusa será la distancia en X al punto. Sin embargo, si el punto está en la vertical, se adopta hipotenusa la distancia Y al punto.

Con los valores de las hipotenusas definidos, lo siguiente a realizar es la comparación de estas con el radio que se establezca. Para la determinación del valor del radio, se adopta el radio de mayor longitud en la dimensión Y. Con el fin de conseguir esto, se emplea la variable “Ym” que se define como un vector descendente donde el primer dato es el máximo valor positivo del eje Y referido al nuevo sistema de referencia.

```
%% CÁLCULO DE RADIO A EMPLEAR.
```

```
radio=Ym(1); %Determinación del radio a emplear para tomar datos.
```

	1	2	3	4	5	6	7	8	9	10	11	12
1	2.8932	2.8329	2.7726	2.7123	2.6521	2.5918	2.5315	2.4712	2.4110	2.3507	2.2904	2.2301
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												

Figura 4.64. Establecimiento del valor del radio para realizar la comparación de las hipotenusas.

Por lo tanto, la mayor longitud vertical será el valor del primer punto de “Ym”.

Conociendo el valor del “radio” a emplear, se realiza la definición de los límites del sector de arco a estudiar. Esto queda reflejado en la siguiente parte de los programas. Cabe destacar que la ligera diferencia entre “directividadizq” y “directividaddcha” se reflejará a continuación en la Figura 4.65 y la Figura 4.66.

```
%% DETERMINACIÓN DE LÍMITES DE SECTOR A UTILIZAR.
if (pas==1) %Primera condición donde se establecen las directrices a seguir para los casos de Pasadas Cortas.
    if (v==1 & ws==1 & con==1) %Primera condición correspondiente al primer caso particular dentro de Pasadas Cortas para establecer los límites a estudiar.
        photoextx=0; %Definición de la coordenada X del punto límite a estudiar de la imagen del caso a analizar.
        photoexty=-1410; %Definición de la coordenada Y del punto límite a estudiar de la imagen del caso a analizar.
        posextx=find(photoextx<=X & photoextx+30>=X); %Búsqueda de la posición de la coordenada X límite dentro del vector eje X.
        posextx=find(photoextx<=Y & photoextx+10<=Y); %Búsqueda de la posición de la coordenada Y límite dentro del vector eje Y.
        nx=X(posextx); %Valor de la coordenada X del punto límite a utilizar del vector eje X.
        ny=Y(posextx); %Valor de la coordenada Y del punto límite a utilizar del vector eje Y.
        disty=abs(ny-cy); %Distancia resultante de la resta entre el punto límite a utilizar y el punto de referencia encontrado para componente Y.
        angext=asind(disty/(radio*1000)); %Ángulo extra a analizar.
        posfinal=posextx; %Búsqueda de la posición en el vector Y de esa distancia extra.
    else %Segunda condición donde se establecen los límites para el resto de casos de Pasadas Cortas.
        photoextx=0; %Definición de la coordenada X del punto límite a estudiar de la imagen del caso a analizar.
        photoexty=-1420; %Definición de la coordenada Y del punto límite a estudiar de la imagen del caso a analizar.
        posextx=find(photoextx<=X & photoextx+10>=X); %Búsqueda de la posición de la coordenada X límite dentro del vector eje X.
        posextx=find(photoextx<=Y & photoextx+10>=Y); %Búsqueda de la posición de la coordenada Y límite dentro del vector eje Y.
        nx=X(posextx); %Valor de la coordenada X del punto límite a utilizar del vector eje X.
        ny=Y(posextx); %Valor de la coordenada Y del punto límite a utilizar del vector eje Y.
        disty=abs(ny-cy); %Distancia resultante de la resta entre el punto límite a utilizar y el punto de referencia encontrado para componente Y.
        angext=asind(disty/radio); %Ángulo extra a analizar.
        posfinal=posextx; %Búsqueda de la posición en el vector Y de esa distancia extra.
    end %Cierre de condiciones para los establecer los límites en los distintos casos de Pasadas Cortas.
else %Segunda condición donde se establecen las directrices a seguir para los casos de Pasadas Largas.
    posfinal=tamy; %Definición de la posición en el vector Y de esa distancia extra como la posición máxima del vector Y.
end %Cierre de condiciones para las distintas pasadas.
```

Figura 4.65. Definición de límites del arco a emplear para directividadizq. La diferencia con directividaddcha es que en el primer programa hay un caso extra que el segundo no presenta.

```

%% DETERMINACIÓN DE LÍMITES DE SECTOR A UTILIZAR.
if (pas==1) %Primera condición donde se establecen las directrices a seguir para los casos de Pasadas Cortas.

photoextx=0; %Definición de la coordenada X del punto límite a estudiar de la imagen del caso a analizar.
photoexty=-1420; %Definición de la coordenada Y del punto límite a estudiar de la imagen del caso a analizar.
posextx=find(photoextx<=X & photoextx+10>=X); %Búsqueda de la posición de la coordenada X límite dentro del vector eje X.
posexty=find(photoexty<=Y & photoexty+10>=Y); %Búsqueda de la posición de la coordenada Y límite dentro del vector eje Y.
nx=X(posextx); %Valor de la coordenada X del punto límite a utilizar del vector eje X.
ny=Y(posexty); %Valor de la coordenada Y del punto límite a utilizar del vector eje Y.
disty=abs(ny-cy); %Distancia resultante de la resta entre el punto límite a utilizar y el punto de referencia encontrado para componente Y.
angext=asind(disty/(radio*1000)); %Ángulo extra a analizar.
posfinal=posexty; %Búsqueda de la posición en el vector Y de esa distancia extra.

else %Segunda condición donde se establecen las directrices a seguir para los casos de Pasadas Largas.

posfinal=tamy; %Definición de la posición en el vector Y de esa distancia extra como la posición máxima del vector Y.

end %Cierre de condiciones para las distintas pasadas.
    
```

Figura 4.66. Definición de límites del arco a emplear para directividaddcha. Como queda reflejado en la figura anterior, aquí ese caso extra no existe.

Este código es la adaptación del procedimiento explicado anteriormente, para determinar el sector de arco a emplear (solamente necesario para Pasadas Cortas). En primer lugar, se define un punto de la horizontal coincidente con la banda de rodadura. En este caso particular, el punto fue [0,-1420] ([0,-1410] para el caso extra).

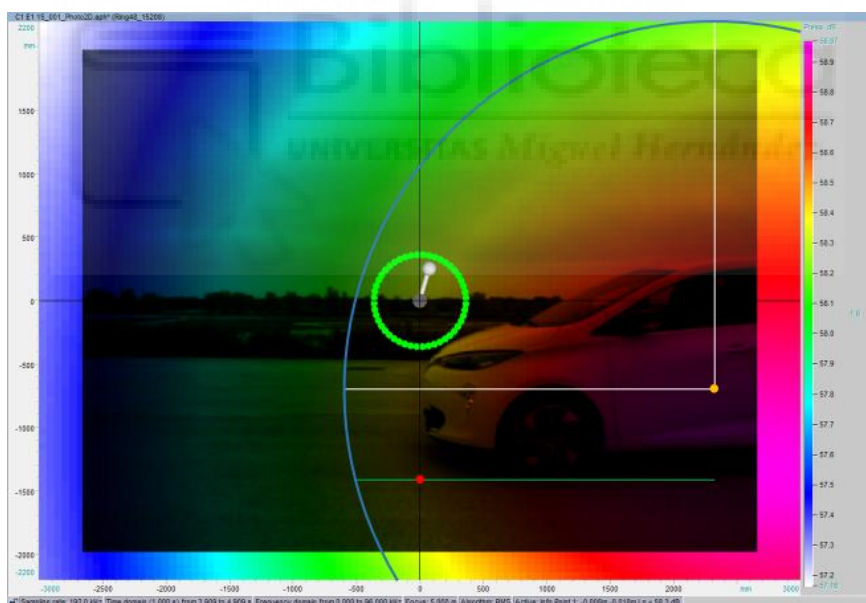


Figura 4.67. Definición de un punto de la horizontal coincidente con la banda de rodadura.

Más adelante, se adapta el punto para que coincida con los valores definidos en los vectores X e Y. Para ello, se vuelve a definir dos operaciones con el comando “find”, para la búsqueda de la posición de los valores que presenten mayor similitud con el nuevo punto en los vectores.



En siguiente lugar, se determina la distancia vertical que hay entre horizontales. Para ello, se resta la coordenada Y del punto de rodadura menos la coordenada Y del punto de referencia, haciendo a continuación su valor absoluto.

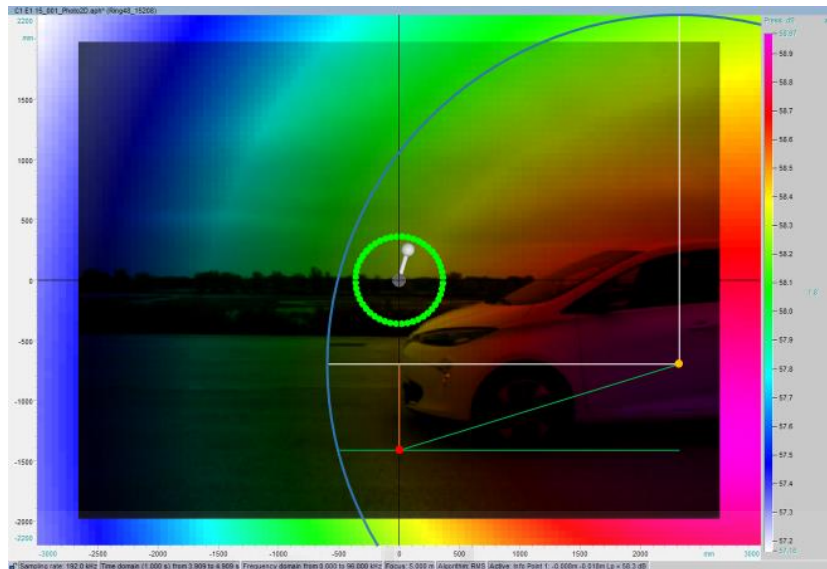


Figura 4.68. Obtención de la distancia entre horizontales.

Para obtener una mayor información del proceso, se determina el ángulo correspondiente al sector de arco extra a analizar. Se genera un triángulo donde el cateto opuesto es la distancia entre horizontales y la hipotenusa es el radio determinado en el bloque anterior.

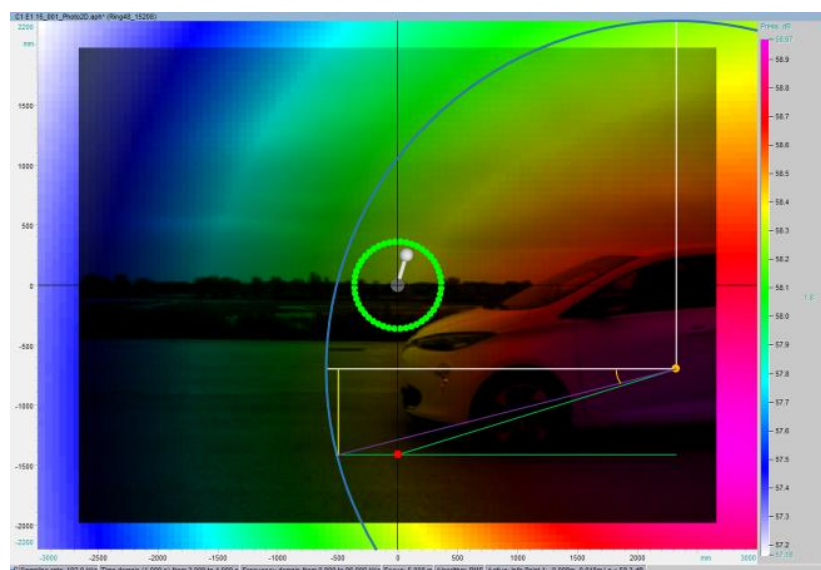


Figura 4.69. Triángulo a analizar.

Partiendo del cateto opuesto e hipotenusa, con el método “asind” de Matlab, se calcula la razón trigonométrica del arco seno y se obtiene el ángulo deseado.

La última línea se encarga de almacenar la posición límite hasta donde se quiera llevar el análisis de datos. No es necesario proseguir con trigonometría, puesto que ya se tiene posición del vector.

Para los casos de Pasadas Largas, el procedimiento es diferente. Como no es necesario recortar el arco de análisis, la posición límite a analizar coincidirá con el tamaño máximo de la matriz de presión sonora.

En “directividadizq”, existe una condición donde se detalla el caso especial ( $ws==1$  &  $v==1$  &  $con==1$ ), donde el software de la cámara acústica definió una matriz de presión sonora con dimensiones distintas a las habituales. El principal cambio se dio en el establecimiento del punto de la horizontal coincidente con la banda de rodadura y las condiciones de búsqueda de dicho punto en los ejes X e Y.

Hasta el momento, se han definido todos aquellos parámetros necesarios para comenzar el procesamiento de los datos (límites de arco, radio a comparar, distancias entre puntos, etc.). Es ahora cuando se comienzan a modificar. Para ello, se compara en primer lugar el radio definido con las hipotenusas calculadas, para definir qué puntos se emplearán y cuáles se descartarán.

```
%% DETERMINACIÓN DE PUNTOS ÚTILES DE LA MATRIZ DE PRESIÓN SONORA.
k=1; %Definición de contador para definir nuevas variables.
for i=1:tamy %Primer bucle para recorrer el eje Y de la matriz de hipotenusas de arriba a abajo, crear y modificar variables.
    for j=1:tamx %Segundo bucle para recorrer el eje X de la matriz de hipotenusas de izquierda a derecha, crear y modificar variables.
        if (hipotenusa(i,j)>=radio+0.03 || hipotenusa(i,j)<=radio-0.03) || i>posfinal %Primera condición donde se establecen puntos inútiles de la matriz de presión sonora en Pa.
            Lppa(i,j)=0; %Asignación de aquellos valores que cumplan la condición anterior del valor 0.
        else %Segunda condición donde se establecen que puntos son útiles de la matriz de presión sonora en Pa.
            vectapoymed(k)=Lppa(i,j); %Almacenamiento del valor del punto útil en variable.
            k=k+1; %Aumento del contador en una unidad.
        end %Cierre de condiciones.
    end %Cierre del segundo bucle.
end %Cierre del primer bucle.
```

Figura 4.70. Manipulación de los datos de presión sonora y generación nuevas variables en el conjunto directividadizq/directividaddcha.

Como en el cálculo de la hipotenusa, el bucle “for” anidado realiza la misma función que el anterior (recorrido de todos los puntos del plano XY).

Se realiza igualmente un cribado entre puntos útiles y no útiles. Los puntos descartados son los mayores o menores al radio definido anteriormente, más un error aceptable de 3 cm. También

se consideran como no útiles aquellos que se encuentren más allá del límite del sector de arco extra. Para estos valores el programa cambia su valor a 0, para que no genere interferencias.

Cuando los puntos sean útiles, se generará una variable en forma de vector llamado “vectapoymed”, empleado posteriormente para obtener la media de los valores de dichos puntos y así calcular el valor de presión sonora omnidireccional.

Tras hallar los valores de presión sonora a emplear, se realiza el cálculo de directividad vertical. Esto se debe a que el plano de trabajo configurado coincidía con el paso del coche, por lo tanto, no existía ningún tipo de distancia horizontal entre el plano de trabajo y el plano de circulación del coche.

```
%% CÁLCULO DEL ÍNDICE DE DIRECTIVIDAD Y TRANSFORMACIÓN DE LA MATRIZ DE PRESIÓN SONORA DE Pa A dB.  
Lppaomni=mean(vectapoymed); %Cálculo de la media de la variable donde se almacenan los valores de los puntos útiles de presión sonora en Pa.  
Q=(Lppa.^2)/(Lppaomni^2); %Cálculo del factor de directividad.  
DI=10*log10(Q); %Cálculo del índice de directividad.  
LpdB=10*log10(Lppa./(20*10^(-6))^2); %Transformación de la matriz de presión sonora de Pa a dB.
```

Figura 4.71. Manipulación de los datos de presión sonora y generación nuevas variables en el conjunto directividadizq/directividadaddcha.

En primer lugar, se determina el nivel de presión sonora omnidireccional con el comando “mean” que realiza la media de la variable definida. En este caso, “vectapoymed” ya ha guardado los valores de los puntos útiles de la matriz de presión sonora. Para determinar el factor de directividad, conviene trabajar en Pa empleando la expresión (8). Seguidamente, se calcula el índice de directividad convirtiendo los valores de Pa a dB por medio de la ecuación (9). La última línea de código, de esta parte, transforma la presión sonora de Pa a dB por medio de la ecuación (1). Este último paso se realiza para ejecutar una representación que valide la respuesta del índice de directiva del sonido de advertencia.

En este instante, se poseen los datos de directividad. Ahora queda representarlos. No obstante, no se puede hacer las cosas, así como así, puesto que, tanto para llevar a cabo una representación en coordenadas cartesianas como en coordenadas polares, Matlab requiere un vector para los datos a representar y otro vector para la posición donde colocar cada uno de los valores del otro vector. Pero lo que en realidad lo que se ha estado manejando ha sido una matriz de presión sonora, por lo que toca generar una conversión de los datos de matriz a vector.

Pero antes de explicar el código de parte, se debe echar un vistazo a la distribución de los datos según el sentido de marcha y cómo se debe realizar su traspaso según el caso.

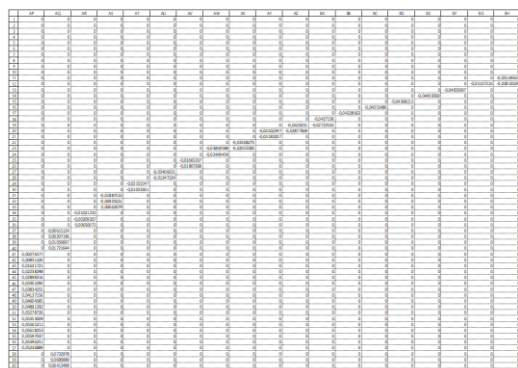


Figura 4.72. Distribución de los datos en la matriz de presión sonora con sentido de marcha a izquierda.

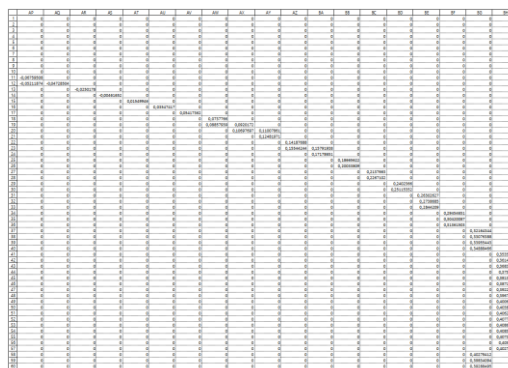


Figura 4.73. Distribución de los datos en la matriz de presión sonora con sentido de marcha a derecha.

El arco de datos se encuentra dirigido hacia la dirección del sentido de la marcha. Esto influye, en parte, en la obtención de ese vector de datos. Los datos deben estar en el orden correcto, para que cada ángulo del vector de posición coincida con el dato correcto del índice de directividad.

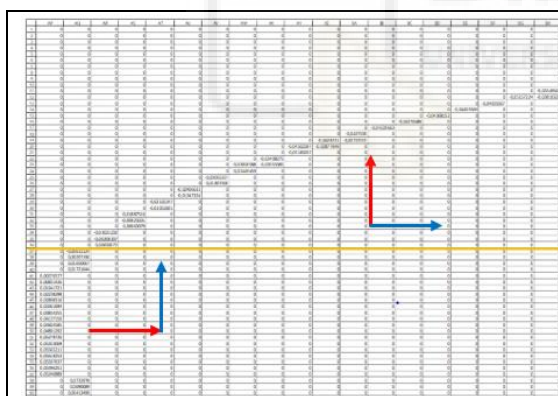


Figura 4.74. Direcciones a establecer en la recogida de valores para sentido de marcha a izquierda.

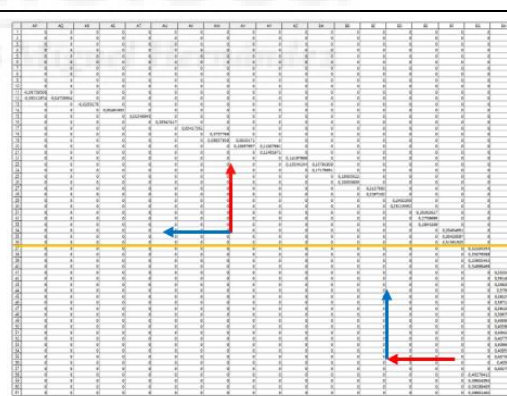


Figura 4.75. Direcciones a establecer en la recogida de valores para sentido de marcha a derecha.

Para los sentidos de marcha izquierdo y derecho, se debe definir una zona intermedia donde cambiar la dirección de recogida de los datos. Esto se debe a la existencia de zonas donde los puntos se distribuyen aproximadamente de forma vertical, y zonas donde los datos se encuentran dispuestos más o menos de forma horizontal.

En el caso del sentido izquierdo de la marcha, la extracción por debajo de la posición intermedia se realiza recorriendo las filas de abajo a arriba, pasando por todos los puntos de cada

fila de izquierda a derecha antes de pasar a la siguiente. Por otro lado, por encima de la posición intermedia, se produce un recorrido de las columnas de izquierda a derecha, pasando por todos los elementos de la columna de abajo a arriba.

En cambio, al hablar de sentidos de marcha hacia derechas, la extracción de datos por debajo de la posición intermedia se realiza de abajo a arriba, pasando por todos los puntos de cada fila de derecha a izquierda. Si la recogida se establece por encima de la posición intermedia, se produce un recorrido de las columnas de derecha a izquierda, recorriendo todos los elementos de la columna de abajo a arriba antes de pasar a la siguiente columna.

De este modo se obtiene un vector de dimensión  $n \times 1$ . Ahora ya se puede proceder a explicar el código

```

%% CÁLCULO, ARREGLO Y DEFINICIÓN DE PARÁMETROS PARA GRAFICAR.
z=1; %Definición de contador para definir nuevas variables.
posyinter=round(tamy/2); %Cálculo del punto hasta el cual realizar pasadas de izquierda a derecha.

for i=tamy:-1:posyinter %Primer bucle para recorrer el eje Y de la matriz de presión sonora en Pa de abajo a arriba, crear y modificar variables.
    for j=1:tamx %Segundo bucle para recorrer el eje X de la matriz de presión sonora en Pa de izquierda a derecha, crear y modificar variables.
        if Lppa(i,j)~=0 %Primera condición donde se establecen las directrices para puntos de la matriz de presión sonora en Pa distintos a 0.
            copuesto(z)=abs(Ym(i)); %Cálculo del cateto opuesto para puntos útiles.
            ccontiguo(z)=abs(Xm(j)); %Cálculo del cateto contiguo para puntos útiles.

            if i<=tamy & i>=posy %Primera condición para cálculo correcto de ángulo para graficar.
                theta(z)=0-atand(copuesto(z)/ccontiguo(z)); %Directriz para casos con ángulos menores o iguales a 0°.
            else %Segunda condición para cálculo correcto de ángulo para graficar.
                theta(z)=atand(copuesto(z)/ccontiguo(z)); %Directriz para casos con ángulos mayores de 0°.
            end %Cierre de condiciones.

            rho(z)=DI(i,j); %Traspaso de valores de índice de directividad de matriz a vector de 1 fila.
            rhodB(z)=LpdB(i,j); %Traspaso de valores de presión sonora en dB a vector de 1 fila.
            DI(i,j)=0; %Asignación del valor 0 para el valor que ha sido tomado.
            z=z+1; %Aumento de contador en una unidad.

        end %Cierre de condiciones.
    end %Cierre del segundo bucle.
end %Cierre del primer bucle.

for j=1:tamx %Primer bucle para recorrer el eje X de la matriz de presión sonora en Pa de izquierda a derecha, crear y modificar variables.
    for i=(posyinter-1):-1:1 %Segundo bucle para recorrer el eje Y de la matriz de presión sonora en Pa de abajo a arriba, crear y modificar variables.
        if Lppa(i,j)~=0 %Primera condición donde se establecen las directrices para puntos de la matriz de presión sonora en Pa distintos a 0.
            copuesto(z)=abs(Ym(i)); %Cálculo del cateto opuesto para puntos útiles.
            ccontiguo(z)=abs(Xm(j)); %Cálculo del cateto contiguo para puntos útiles.

            if j<=posx %Primera condición para cálculo correcto de ángulo para graficar.
                theta(z)=atand(copuesto(z)/ccontiguo(z)); %Directriz para casos con ángulos mayores de 0° y menores a 90°.
            else %Segunda condición para cálculo correcto de ángulo para graficar.
                theta(z)=(90-atand(copuesto(z)/ccontiguo(z)))+90; %Directriz para casos con ángulos mayores de 90°.
            end %Cierre de condiciones.

            rho(z)=DI(i,j); %Traspaso de valores de índice de directividad de matriz a vector de 1 fila.
            rhodB(z)=LpdB(i,j); %Traspaso de valores de presión sonora en dB a vector de 1 fila.
            z=z+1; %Aumento de contador en una unidad.

        end %Cierre de condiciones.
    end %Cierre del segundo bucle.
end %Cierre del primer bucle.
    
```

Figura 4.76. Obtención de variables para representar información en el programa directividadizq.

```

for i=tamy:-1:posyinter %Primer bucle para recorrer el eje Y de la matriz de presión sonora en Pa de abajo a arriba, crear y modificar variables.
for j=tamx:-1:1 %Segundo bucle para recorrer el eje X de la matriz de presión sonora en Pa de derecha a izquierda, crear y modificar variables.
    if Lppa(i,j)~=0 %Primera condición donde se establecen las directrices para puntos de la matriz de presión sonora en Pa distintos a 0.
        copuesto(z)=abs(Ym(i)); %Cálculo del cateto opuesto para puntos útiles.
        ccontiguo(z)=abs(Xm(j)); %Cálculo del cateto contiguo para puntos útiles.

        if i<=tamy & i>=posy %Primera condición para cálculo correcto de ángulo para graficar.
            theta(z)=0-atand(copuesto(z)/ccontiguo(z)); %Directriz para casos con ángulos menores o iguales a 0°.
        else %Segunda condición para cálculo correcto de ángulo para graficar.
            theta(z)=atand(copuesto(z)/ccontiguo(z)); %Directriz para casos con ángulos mayores de 0°.
        end %Cierre de condiciones.

        rho(z)=DI(i,j); %Traspaso de valores de índice de directividad de matriz a vector de 1 fila.
        rhodB(z)=LpdB(i,j); %Traspaso de valores de presión sonora en dB a vector de 1 fila.
        DI(i,j)=0; %Asignación del valor 0 para el valor que ha sido tomado.
        z=z+1; %Aumento de contador en una unidad.

    end %Cierre de condiciones.
end %Cierre de condiciones.
end %Cierre del primer bucle.

for j=tamx:-1:1 %Primer bucle para recorrer el eje X de la matriz de presión sonora en Pa de derecha a izquierda, crear y modificar variables.
for i=(posyinter-1):-1:1 %Segundo bucle para recorrer el eje Y de la matriz de presión sonora en Pa de abajo a arriba, crear y modificar variables.
    if Lppa(i,j)~=0 %Primera condición donde se establecen las directrices para puntos de la matriz de presión sonora en Pa distintos a 0.
        copuesto(z)=abs(Ym(i)); %Cálculo del cateto opuesto para puntos útiles.
        ccontiguo(z)=abs(Xm(j)); %Cálculo del cateto contiguo para puntos útiles.

        if j<=posx %Primera condición para cálculo correcto de ángulo para graficar.
            theta(z)=(90-atand(copuesto(z)/ccontiguo(z))+90); %Directriz para casos con ángulos mayores de 0° y menores a 90°.
        else %Segunda condición para cálculo correcto de ángulo para graficar.
            theta(z)=atand(copuesto(z)/ccontiguo(z)); %Directriz para casos con ángulos mayores de 90°.
        end %Cierre de condiciones.

        rho(z)=DI(i,j); %Traspaso de valores de índice de directividad de matriz a vector de 1 fila.
        rhodB(z)=LpdB(i,j); %Traspaso de valores de presión sonora en dB a vector de 1 fila.
        z=z+1; %Aumento de contador en una unidad.

    end %Cierre de condiciones.
end %Cierre del segundo bucle.
end %Cierre del primer bucle.

```

Figura 4.77. Obtención de variables para representar información en el programa directivaddcha.

En primer lugar, lo que hace este bloque es definir un nuevo contador que se usará para hacer el traspaso de variables de la matriz a un vector. Esto es debido a que la matriz presenta una dimensión  $m \times n$  (serán necesarios dos contadores para recorrerla) y lo que se precisa para la representación de los datos es un vector de dimensión  $n \times 1$  (solo será necesario un contador para generarlo).

A continuación, se establece la posición intermedia que se requiere para definir el cambio en la manera de realizar la extracción de datos. La información se guarda en la variable “posyinter”.

Seguidamente, se genera el primer bucle “for” anidado que, según el sentido de marcha, tendrá una forma u otra. Este primer bucle realiza la extracción de datos en la primera zona, es decir, por debajo de la posición intermedia.

En “directividadizq”, su bucle hace que el contador “i” disminuya posiciones (recorrido de abajo a arriba) pasando de fila en fila. Seguidamente el contador “j” va aumentando su valor para recorrer punto por punto los datos de la fila de izquierda a derecha antes de hacer el paso a otra fila. En cambio, en “directividaddcha”, el bucle cambia el contador “i” de la misma manera que en el otro programa, pero el contador “j” hace lo contrario, realizando la misma función que en “directividadizq” pero de derecha a izquierda (disminuyendo el valor del contador). Esto queda reflejado en la Figura 4.76 y Figura 4.77.

En ambos programas, tras la definición del primer bucle anidado, se asigna una condición donde se especifican las operaciones a realizar con los puntos útiles. Posteriormente, cuando el dato que se estudia es mayor que diferente a 0, se determina la distancia en X y en Y desde el punto de referencia hasta dicho punto. Esto se traduce en la obtención de catetos para el cálculo del ángulo existente entre nuestro origen del sistema de referencia al punto estudiado.

A continuación, se calcula el ángulo anterior en función del sector en el que se sitúe nuestro punto, consiguiendo de esta forma que no haya superposición de valores y que estos presenten una distribución lógica. El ángulo se almacena en una variable en forma de vector fila (usando el contador “z”).

El siguiente paso es almacenar el valor del punto correspondiente de la matriz que contiene el índice de directividad y el valor del punto de la matriz de presión sonora en dB en variables con forma de vector fila. Tras el almacenamiento de los puntos, se cambia el valor de estos por 0 con el objetivo de evitar errores. Finalmente, se aumenta el valor del contador en una unidad para proseguir con la tarea de almacenamiento de datos.

Posteriormente, se estableció otro bucle anidado muy similar al primero. Su objetivo era recolectar datos en la parte superior de la posición intermedia. En “directividadizq”, el contador “j” aumenta su valor, recorriendo los datos de la fila de izquierda a derecha antes de pasar a la siguiente. Por otro lado, el valor del contador “i” disminuye, yendo desde los datos que se encuentran en la parte inferior de la matriz hasta las posiciones superiores. En “directividaddcha” el contador “j” realiza el recorrido de forma contraria al de “directividadizq”, desplazándose de derecha a izquierda. No obstante, el contador “i” mantiene la forma en que recorre la matriz tal y como se define en “directividadizq”. Esto se refleja en la Figura 4.76 y Figura 4.77.

Después de la definición del segundo bucle anidado, se trabaja del mismo modo que en el primer bloque con unas ligeras modificaciones. Para aquellos puntos que sean útiles se determina que distancia existe, en X y en Y, entre dicho punto y el sistema de referencia. Con estas distancias, se calcula el ángulo que existe entre los dos puntos, teniendo en cuenta el sector angular. Existe una diferencia entre “directivaddcha” y “directivadizq” para el cálculo de los ángulos en esta parte, que se distribuyen desde un cierto margen por debajo 90° hasta un pequeño rango por encima de 90°. Esta diferencia se produce por la forma en que se distribuyen los datos en las matrices que se generan en estos programas. En “directivadizq” el arco se dirige hacia la izquierda, por ello la forma en la que calcularíamos los ángulos es de forma contraria a como se hace al adoptar el sentido levógiro. En cambio, en “directivaddcha” no ocurre esto, ya que el arco se dirige a la derecha, y el sentido de giro del arco coincide con el sentido dextrógiro. Tras su cálculo, se almacena en una variable con forma de fila.

Finalmente se extrae, para esta zona, el dato del punto correspondiente del índice de directividad y el valor del mismo punto de la matriz de presión sonora en dB. A su vez, aumentamos el contador “z” en una unidad, para acceder a la siguiente posición de la variable que almacena la información.

Una vez obtenidos los valores de forma correcta, se obtuvieron una serie de parámetros que permitiesen representar la información además de realizar esto de una forma más automática.

```
thetar=deg2rad(theta); %Transformación de la variable de grados a radianes.  
  
rhomax=max(rho); %Designación del máximo valor del índice de directividad.  
rhodBmax=max(rhodB); %Designación del máximo valor de presión sonora en dB.  
rhodBmaxr=ceil(rhodBmax,1); %Redondeo a 1 cifra decimal del valor máximo de presión sonora en dB.  
  
rhomin=min(rho); %Designación del mínimo valor del índice de directividad.  
rhodBmin=min(rhodB); %Designación del mínimo valor de presión sonora en dB.  
rhodBminr=fix(rhodBmin); %Truncamiento del valor máximo de presión sonora en dB.
```

Figura 4.78. Generación de variables para graficar en el conjunto directivadizq/directivaddcha.

Lo primero que hace el código es pasar los ángulos a radianes. Esto es debido a que Matlab solamente permite representar, en forma polar, cuando los datos de posición con ángulos se proporcionan en radianes.

Tras esto se obtienen los máximos y mínimos del vector de índice de directividad y del vector de presión sonora en dB, que fueron obtenidos en el apartado anterior. Seguidamente, se



realiza el truncamiento de dichos valores hacia el entero superior o hacia el entero inferior respectivamente.

En este instante solamente queda representar los datos. Para ello, se generó la siguiente parte.

```
%% REPRESENTACIÓN DE DIRECTIVIDAD.
%GRÁFICO EN COORDENADAS CARTESIANAS DEL ÍNDICE DE DIRECTIVIDAD.
figure %Representación del gráfico en una ventana independiente.
plot(thetar,rho,'Color',(1/255)*[119,136,153],'LineWidth',3) %Representación del índice de directividad en coordenadas cartesianas.
hold on %Mantenimiento del gráfico anterior.
plot(thetar,rho,'s','MarkerEdge',(1/255)*[128,128,128],'MarkerFace',(1/255)*[240,255,255],'MarkerSize',7) %Graficación de marcadores en cada punto.
hold off %Eliminación del mantenimiento de gráficos.
xlim([thetar(1) thetar(length(thetar))]) %Definición del extremo del eje X del gráfico en coordenadas cartesianas.
title('DIRECTIVIDAD','FontSize',15) %Asignación de título.
ylabel('Índice de Directividad (dB)') %Asignación de etiqueta para eje Y.
xlabel('Dirección (rad)') %Asignación de etiqueta para eje X.
grid minor %Aplicación de cuadrícula de tamaño pequeño.

%GRÁFICO EN COORDENADAS POLARES DEL ÍNDICE DE DIRECTIVIDAD.
figure %Representación del gráfico en una ventana independiente.
polarplot(thetar,rho,'Color',(1/255)*[188,143,143],'LineWidth',2) %Representación del índice de directividad en coordenadas polares.
hold on %Mantenimiento del gráfico anterior.
polarplot(thetar,rho,'o','MarkerEdge','k','MarkerFace',(1/255)*[255,255,224],'MarkerSize',4) %Graficación de marcadores en cada punto.
hold off %Eliminación del mantenimiento de gráficos.
rlim([rhomin-0.5 rhomax+0.12]) %Asignación de los radios extremos del gráfico en coordenadas polares.
thetalim([theta(1) theta(length(theta))]) %Asignación de los ángulos extremos del gráfico en coordenadas polares.
graf=gca; %Definición de objeto gráfico.
graf.ThetaDir='clockwise'; %Disposición de los ángulos en sentido antihorario.
graf.ThetaZeroLocation='left'; %Posición del número 0 en el gráfico polar.
title('DIRECTIVIDAD (DI)','FontSize',15) %Asignación de título.
rticklabels({'DI = -0.5','DI = 0','DI = 0.5','DI = 1'}) %Definición de las etiquetas de los resultados de índice de directividad obtenidos.
rtickangle(graf,30) %Giro de las etiquetas de la línea anterior 30°.
grid minor %Aplicación de cuadrícula de tamaño pequeño.

%GRÁFICO EN COORDENADAS POLARES DEL NIVEL DE PRESIÓN SONORA EN dB.
figure %Representación del gráfico en una ventana independiente.
polarplot(thetar,rhodB,'Color',(1/255)*[75,0,130],'LineWidth',2) %Representación del nivel de presión sonora en dB en coordenadas polares.
hold on %Mantenimiento del gráfico anterior.
polarplot(thetar,rhodB,'o','MarkerEdge','k','MarkerFace',(1/255)*[255,240,245],'MarkerSize',4) %Graficación de marcadores en cada punto.
hold off %Eliminación del mantenimiento de gráficos.
rlim([rhodBmin rhodBmax]) %Asignación de los radios extremos del gráfico en coordenadas polares.
thetalim([theta(1) theta(length(theta))]) %Asignación de los ángulos extremos del gráfico en coordenadas polares.
graf=gca; %Definición de objeto gráfico.
graf.ThetaDir='clockwise'; %Disposición de los ángulos en sentido antihorario.
graf.ThetaZeroLocation='left'; %Posición del número 0 en el gráfico polar.
title('DIRECTIVIDAD (dB)','FontSize',15) %Asignación de título.
rticklabels({'Lp = 58 dB','Lp = 58.5 dB','Lp = 59 dB','Lp = 59.5 dB','Lp = 64 dB','Lp = 59.8 dB','Lp = 60 dB'}) %Definición etiquetas de resultados en dB obtenidos.
rtickangle(graf,30) %Giro de las etiquetas de la línea anterior 30°.
grid minor %Aplicación de cuadrícula de tamaño pequeño.
```

Figura 4.79. Representación de la directividad en el programa directividadizq.

```
%% REPRESENTACIÓN DE DIRECTIVIDAD.
%GRÁFICO EN COORDENADAS CARTESIANAS DEL ÍNDICE DE DIRECTIVIDAD.
figure %Representación del gráfico en una ventana independiente.
plot(thetar,rho,'Color',(1/255)*[119,136,153],'LineWidth',3) %Representación del índice de directividad en coordenadas cartesianas.
hold on %Mantenimiento del gráfico anterior.
plot(thetar,rho,'s','MarkerEdge',(1/255)*[128,128,128],'MarkerFace',(1/255)*[240,255,255],'MarkerSize',7) %Graficación de marcadores en cada punto.
hold off %Eliminación del mantenimiento de gráficos.
xlim([thetar(1) thetar(length(thetar))]) %Definición del extremo del eje X del gráfico en coordenadas cartesianas.
title('DIRECTIVIDAD','FontSize',15) %Asignación de título.
ylabel('Índice de Directividad (DI)') %Asignación de etiqueta para eje Y.
xlabel('Dirección (rad)') %Asignación de etiqueta para eje X.
grid minor %Aplicación de cuadrícula de tamaño pequeño.

%GRÁFICO EN COORDENADAS POLARES DEL ÍNDICE DE DIRECTIVIDAD.
figure %Representación del gráfico en una ventana independiente.
polarplot(thetar,rho,'Color',(1/255)*[188,143,143],'LineWidth',2) %Representación del índice de directividad en coordenadas polares.
hold on %Mantenimiento del gráfico anterior.
polarplot(thetar,rho,'o','MarkerEdge','k','MarkerFace',(1/255)*[255,255,224],'MarkerSize',4) %Graficación de marcadores en cada punto.
hold off %Eliminación del mantenimiento de gráficos.
rlim([rhomin-0.4 rhomax+0.07]) %Asignación de los radios extremos del gráfico en coordenadas polares.
thetalim([theta(1) theta(length(theta))]) %Asignación de los ángulos extremos del gráfico en coordenadas polares.
graf=gca; %Definición de objeto gráfico.
title('DIRECTIVIDAD (DI)','FontSize',15) %Asignación de título.
rticklabels({'DI = -0.2','DI = 0','DI = 0.2'}) %Definición de las etiquetas de los resultados de índice de directividad obtenidos.
rtickangle(graf,30) %Giro de las etiquetas de la línea anterior 30°.
grid minor %Aplicación de cuadrícula de tamaño pequeño.
```

```
%GRÁFICO EN COORDENADAS POLARES DEL NIVEL DE PRESIÓN SONORA EN dB.  
figure %Representación del gráfico en una ventana independiente.  
polarplot(thetar,rhoDB,'color',(1/255)*[75,0,130],'LineWidth',2) %Representación del nivel de presión sonora en dB en coordenadas polares.  
hold on %Mantenimiento del gráfico anterior.  
polarplot(thetar,rhoDB,'o','MarkerEdge','k','MarkerFace',(1/255)*[255,240,245],'MarkerSize',4) %Graficación de marcadores en cada punto.  
hold off %Eliminación del mantenimiento de gráficos.  
rlim([rhoDBmin rhoDBmax]) %Asignación de los radios extremos del gráfico en coordenadas polares.  
thetalim([theta(1) theta(length(theta))]) %Asignación de los ángulos extremos del gráfico en coordenadas polares.  
graf=gca; %Definición de objeto gráfico.  
title('DIRECTIVIDAD (dB)','FontSize',15) %Asignación de título.  
rticklabels(['lp = 58 dB','lp = 59.5 dB','lp = 59 dB','lp = 59.5 dB','lp = 60 dB']) %Definición de las etiquetas de los resultados de presión sonora en dB obtenidos.  
rtickangle(graf,30) %Giro de las etiquetas de la línea anterior 30°.  
grid minor %Aplicación de cuadrícula de tamaño pequeño.
```

Figura 4.80. Representación de la directividad en el programa directividaddcha.

Este bloque, se encuentra dividido en tres partes distintas. Cada una de estas partes representa una determinada gráfica mostrando información específica.

La primera parte, se encarga de representar el índice de directividad en un gráfico en coordenadas cartesianas. Para ello se emplea, en primer lugar, el comando “plot”, tomando como eje X los valores de los ángulos en radianes, y como eje Y los valores del índice de directividad. Dentro del “plot” también se define el color de la línea del gráfico. Seguidamente, se usa un “hold on” para mantener la gráfica anterior en la ventana sin que se produzcan modificaciones en esta. A continuación, se vuelve a definir otro “plot” con los mismos datos que el primero, pero cambiando la forma en la que la información se representa de línea continua a puntos, definiendo, a su vez, las características de los mismos. Finalmente, se establecen otros detalles para mejorar la apariencia de la ventana, como la definición del título o los límites de los ejes.

La segunda parte, genera una gráfica donde se muestra el índice de directividad en un gráfico en coordenadas polares. Se sigue un procedimiento semejante al realizado en la primera parte, solo que con “polarplot”. Es decir, primero se crea el primer “polarplot” donde la información se representa con una línea continua, después se activa el comando “hold on” y, por último, se establece otro “polarplot” donde los datos se presentan como marcadores. Tras esto, se definen características de la gráfica como el sentido de giro del sector polar o el valor de las divisiones del mismo.

La última parte, es prácticamente igual que la segunda. La diferencia entre ambas radica en el hecho de que, mientras la segunda parte representa la emisión por medio del índice de directividad, la última muestra la directividad a través del nivel de presión sonora en dB. Lo demás se mantiene exactamente igual en las dos partes.

No obstante, existe una ligera desigualdad entre “directividaddizq” y “directividaddcha”. Esta queda referenciada en la Figura 4.77 y la Figura 4.78. Principalmente, la diferencia se debe a la manera en el que el sector polar quedará representado. Si el sentido de marcha es a izquierdas, el giro del sector polar será contrario a las agujas del reloj. Si el sentido de marcha es a derechas, se hará lo opuesto.

De este modo, se acaba con la explicación del conjunto de programas “directividadizq” y “directivaddcha”. Al ejecutar uno de los programas, los resultados que se obtienen se representan en las siguientes imágenes.

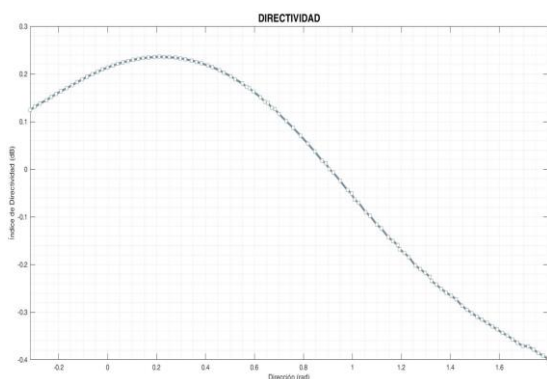


Figura 4.81. Representación del índice de directividad en coordenadas cartesianas al ejecutar directivaddcha.

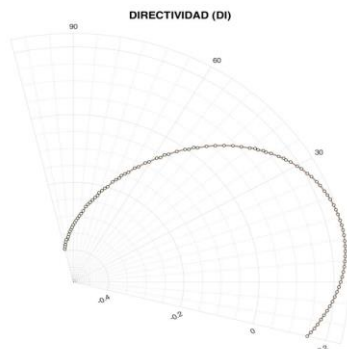


Figura 4.82. Representación del índice de directividad en coordenadas polares al ejecutar directivaddcha.

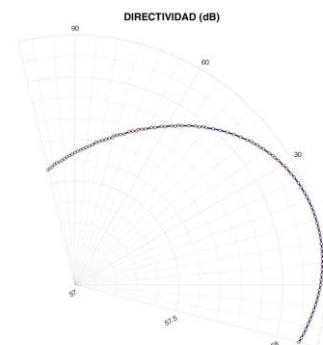


Figura 4.83. Representación de la emisión en dB en coordenadas polares al ejecutar directivaddcha.

#### 4.4.2.4. ESQUEMA DE ARCHIVOS Y PROGRAMAS.

Lo descrito hasta este momento, permite obtener los datos para una sola medición. Por ello, se procedió del mismo modo que en la tarea 1 y se desarrolló un conjunto de programas que automatizasen el proceso.

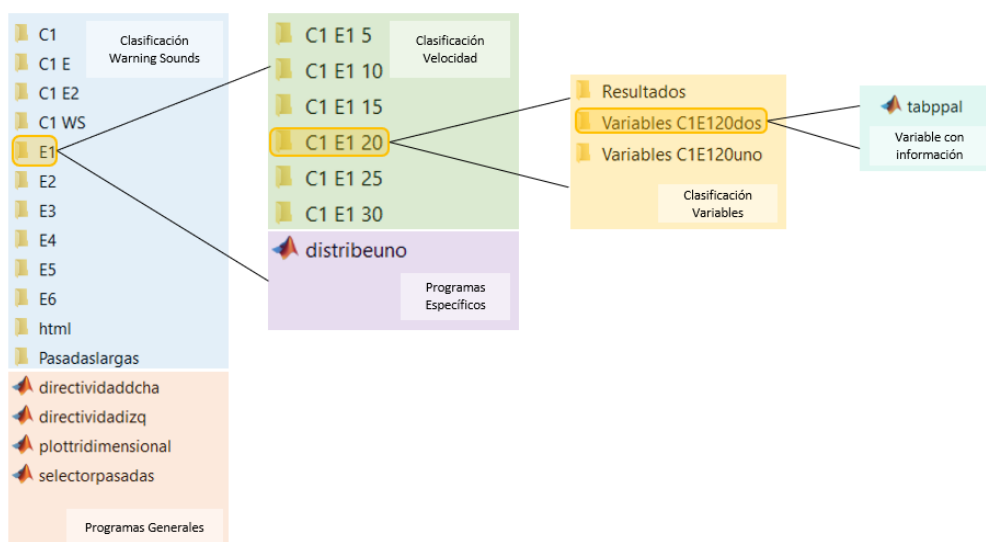


Figura 4.84. Esquema de la distribución de archivos y programas.

Se puede observar que la distribución adoptada es muy similar a la desarrollada en el apartado 4.4.1.6. de este trabajo. Se mantuvo el nombre del programa general “selectorpasadas”, debido a que su función es la misma que en la anterior tarea. No obstante, el conjunto de programas “obtpresson + sonido de advertencia” cambio su nomenclatura a “distrib + sonido de advertencia”, aunque su función se mantiene aproximadamente (es un programa distribuidor).

#### 4.4.2.5. PROGRAMA SELECTORPASADAS

El programa “selectorpasadas” se encarga de establecer las operaciones a realizar en función del tipo de pasada que se quiera estudiar.

La primera parte del programa, se encarga de preguntar al usuario si se desea hacer un estudio de un caso de Pasada Corta o Larga.

```
%% PETICIÓN WARNING SOUND PASADAS CORTAS O LARGAS

display('1-Pasadas Cortas') %Escribir por pantalla significado del código 1.
display('2-Pasadas Largas') %Escribir por pantalla significado del código 2.

pas=input('¿Qué tipo de warning sound quiere analizar? '); %Pregunta por pantalla el tipo de warning sound a analizar y lo almacena.
```

Figura 4.85. Establecimiento del tipo de pasada a estudiar en selectorpasadas para determinar la directividad vertical.

El bloque comienza presentando por pantalla el código asignado para las Pasadas Cortas y Largas. Tras esto, se pide que clase de sonido de advertencia se desea estudiar y la respuesta se almacena en una variable.

El siguiente bloque, se encarga de definir cada una de las rutas para cada sonido de advertencia a los programas específicos que se encargarán de extraer los datos de las variables de almacenamiento “tabppal”. Esto solamente ocurre para los ensayos de Pasadas Cortas ya que, se realizaron a distintas velocidades.

```
switch pas %Estructura de control switch que realiza las funciones que se definan para cada clase de warning sound según sean pasadas Largas o Cortas.

    case 1 %Operaciones para opción Pasadas Cortas.

        %% PETICIÓN DE WARNING SOUND PASADAS CORTAS A EJECUTAR

        display('1-Warning sound E1') %Mostrar por pantalla que código es el warning sound E1.
        display('2-Warning sound E2') %Mostrar por pantalla que código es el warning sound E2.
        display('3-Warning sound E3') %Mostrar por pantalla que código es el warning sound E3.
        display('4-Warning sound E4') %Mostrar por pantalla que código es el warning sound E4.
        display('5-Warning sound E5') %Mostrar por pantalla que código es el warning sound E5.
        display('6-Warning sound E6') %Mostrar por pantalla que código es el warning sound E6.
        display('7-Warning sound C1') %Mostrar por pantalla que código es el warning sound C1.
        display('8-Warning sound C1 E') %Mostrar por pantalla que código es el warning sound C1 E.
        display('9-Warning sound C1 WS') %Mostrar por pantalla que código es el warning sound C1 WS.
        display('10-Warning sound C1 E2') %Mostrar por pantalla que código es el warning sound C1 E2.

        ws=input('¿Qué warning sound quiere estudiar? '); %Pregunta por pantalla el warning sound a analizar y lo almacena.
```

```
%% LISTADO PASADAS CORTAS

switch ws %Estructura de control switch que realiza las funciones que se definan para cada warning sound de Pasadas Cortas.

    case 1 %Operaciones para warning sound E1.

        run('..\Tarea 2\E1\distribuno.m') %Ejecución del script con nombre 'distribuno'.

    case 2 %Operaciones para warning sound E2.

        run('..\Tarea 2\E2\distribedos.m') %Ejecución del script con nombre 'distribedos'.

    case 3 %Operaciones para warning sound E3.

        run('..\Tarea 2\E3\distribetres.m') %Ejecución del script con nombre 'distribetres'.

    case 4 %Operaciones para warning sound E4.

        run('..\Tarea 2\E4\distribecuatro.m') %Ejecución del script con nombre 'distribecuatro'.

    case 5 %Operaciones para warning sound E5.

        run('..\Tarea 2\E5\distribecinco.m') %Ejecución del script con nombre 'distribecinco'.

    case 6 %Operaciones para warning sound E6.

        run('..\Tarea 2\E6\distribeseis.m') %Ejecución del script con nombre 'distribeseis'.

    case 7 %Operaciones para warning sound C1.

        run('..\Tarea 2\C1\distribcuno.m') %Ejecución del script con nombre 'distribcuno'.

    case 8 %Operaciones para warning sound C1 E.

        run('..\Tarea 2\C1 E\distribcunoe.m') %Ejecución del script con nombre 'distribcunoe'.

    case 9 %Operaciones para warning sound C1 WS.

        run('..\Tarea 2\C1 WS\distribcunows.m') %Ejecución del script con nombre 'distribcunows'.

    case 10 %Operaciones para warning sound C1 E2.

        run('..\Tarea 2\C1 E2\distribcunoedos.m') %Ejecución del script con nombre 'distribcunoedos'.

    otherwise %En el caso de que el código introducido no coincida con los casos anteriores se realiza lo siguiente.

        display ('ERROR') %Escribir por pantalla error.
        display ('Introduzca un valor dentro del intervalo') %Escribir por pantalla que se debe hacer para que no de error.
        return %Salida de la estructura de control.

end %Cierre de la estructura de control switch para códigos de Pasadas Cortas.
```

Figura 4.86. Definición de las rutas a programas específicos para Pasadas Cortas en selectorpasadas para determinar la directividad vertical.

En primer lugar, se establece una estructura “switch”, con dos casos distintos. El caso uno se refiere a las operaciones asociadas con Pasadas Cortas. El caso dos tiene que ver con las mediciones de Pasadas Largas. Dentro del caso uno, a través de una serie de comandos “display”, se definen los códigos para cada uno de los sonidos de advertencia a estudiar. Después, se pregunta cual se quiere estudiar, almacenando la entrada del usuario. A continuación, se establece otro “switch”, donde hay un caso por cada sonido de advertencia empleado. Dentro de cada uno de ellos, se establece la ruta a los programas “distrib + sonido de advertencia”. No obstante, se añade el caso “otherwise”, que se activa cuando el código introducido no se corresponda con ningún sonido de advertencia y avisa de que se ha producido un error.

La siguiente parte del programa se corresponde con las acciones a realizar cuando el código introducido por el usuario es el número dos. Es decir, se requiere estudiar una medición de

Pasadas Largas. Como estos ensayos se realizaron a 20 km/h, no es necesario el empleo de programas específicos.

```
case 2 %Operaciones para opción Pasadas Largas.

%% PETICIÓN DE WARNING SOUND PASADAS LARGAS

display('1-Warning sound P1') %Mostrar por pantalla que código es el warning sound P1.
display('2-Warning sound P2') %Mostrar por pantalla que código es el warning sound P2.
display('3-Warning sound P3') %Mostrar por pantalla que código es el warning sound P3.
display('4-Warning sound P4') %Mostrar por pantalla que código es el warning sound P4.
display('5-Warning sound P5') %Mostrar por pantalla que código es el warning sound P5.
display('6-Warning sound P8') %Mostrar por pantalla que código es el warning sound P8.
display('7-Warning sound P9') %Mostrar por pantalla que código es el warning sound P9.
display('8-Warning sound P10') %Mostrar por pantalla que código es el warning sound P10.
display('9-Warning sound P11') %Mostrar por pantalla que código es el warning sound P11.
display('10-Warning sound P12') %Mostrar por pantalla que código es el warning sound P12.
display('11-Warning sound P13') %Mostrar por pantalla que código es el warning sound P13.
display('12-Warning sound P16') %Mostrar por pantalla que código es el warning sound P16.
display('13-Warning sound P17') %Mostrar por pantalla que código es el warning sound P17.
display('14-Warning sound P18') %Mostrar por pantalla que código es el warning sound P18.
display('15-Warning sound P19') %Mostrar por pantalla que código es el warning sound P19.
display('16-Warning sound P20') %Mostrar por pantalla que código es el warning sound P20.

ws=input('¿Qué warning sound quiere estudiar?(En mayúsculas) '); %Pregunta por pantalla el warning sound a analizar y lo almacena.
%% LISTADO PASADAS LARGAS

switch ws %Estructura de control switch que realiza las funciones que se definan para cada warning sound de Pasadas Largas.

    case 1 %Operaciones para warning sound P1.

        v=4; %Definición del código de velocidad del ensayo.
        con=1; %Definición del número de tomas de datos que se realizaron del warning sound.
        load('..\Tarea 2\Pasadaslargas\Variable pasadalarga1\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora.
        run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional'.
        run('..\Tarea 2\directividadizq.m') %Ejecución del script con nombre 'directividadizq'.

    case 2 %Operaciones para warning sound P2.

        v=4; %Definición del código de velocidad del ensayo.
        con=1; %Definición del número de tomas de datos que se realizaron del warning sound.
        load('..\Tarea 2\Pasadaslargas\Variable pasadalarga2\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora.
        run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional'.
        run('..\Tarea 2\directividaddcha.m') %Ejecución del script con nombre 'directividaddcha'.

    case 3 %Operaciones para warning sound P3.

        v=4; %Definición del código de velocidad del ensayo.
        con=1; %Definición del número de tomas de datos que se realizaron del warning sound.
        load('..\Tarea 2\Pasadaslargas\Variable pasadalarga3\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora.
        run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional'.
        run('..\Tarea 2\directividadizq.m') %Ejecución del script con nombre 'directividadizq'.

    case 4 %Operaciones para warning sound P4.

        v=4; %Definición del código de velocidad del ensayo.
        con=1; %Definición del número de tomas de datos que se realizaron del warning sound.
        load('..\Tarea 2\Pasadaslargas\Variable pasadalarga4\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora.
        run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional'.
        run('..\Tarea 2\directividaddcha.m') %Ejecución del script con nombre 'directividaddcha'.

    case 5 %Operaciones para warning sound P5.

        v=4; %Definición del código de velocidad del ensayo.
        con=1; %Definición del número de tomas de datos que se realizaron del warning sound.
        load('..\Tarea 2\Pasadaslargas\Variable pasadalarga5\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora.
        run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional'.
        run('..\Tarea 2\directividadizq.m') %Ejecución del script con nombre 'directividadizq'.

    case 6 %Operaciones para warning sound P8.

        v=4; %Definición del código de velocidad del ensayo.
        con=1; %Definición del número de tomas de datos que se realizaron del warning sound.
        load('..\Tarea 2\Pasadaslargas\Variable pasadalarga8\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora.
        run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional'.
        run('..\Tarea 2\directividaddcha.m') %Ejecución del script con nombre 'directividaddcha'.
```

```

case 7 %Operaciones para warning sound P9.

v=4; %Definición del código de velocidad del ensayo.
con=1; %Definición del número de tomas de datos que se realizaron del warning sound.
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga9\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora.
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional''.
run('..\Tarea 2\directividadizq.m') %Ejecución del script con nombre 'directividadizq''.

case 8 %Operaciones para warning sound P10.

v=4; %Definición del código de velocidad del ensayo.
con=1; %Definición del número de tomas de datos que se realizaron del warning sound.
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga10\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora.
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional''.
run('..\Tarea 2\directivaddcha.m') %Ejecución del script con nombre 'directivaddcha''.

case 9 %Operaciones para warning sound P11.

v=4; %Definición del código de velocidad del ensayo.
con=1; %Definición del número de tomas de datos que se realizaron del warning sound.
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga11\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora.
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional''.
run('..\Tarea 2\directividadizq.m') %Ejecución del script con nombre 'directividadizq''.

case 10 %Operaciones para warning sound P12.

v=4; %Definición del código de velocidad del ensayo.
con=1; %Definición del número de tomas de datos que se realizaron del warning sound.
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga12\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora.
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional''.
run('..\Tarea 2\directivaddcha.m') %Ejecución del script con nombre 'directivaddcha''.

case 11 %Operaciones para warning sound P13.

v=4; %Definición del código de velocidad del ensayo.
con=1; %Definición del número de tomas de datos que se realizaron del warning sound.
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga13\tabppal.mat') %Carga de la variable que contiene la matriz de presión sonora.
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional''.
run('..\Tarea 2\directividadizq.m') %Ejecución del script con nombre 'directividadizq''.

case 12 %Operaciones para warning sound P16.

v=4; %Definición del código de velocidad del ensayo.
con=1; %Definición del número de tomas de datos que se realizaron del warning sound.
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga16\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora.
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional''.
run('..\Tarea 2\directivaddcha.m') %Ejecución del script con nombre 'directivaddcha''.

case 13 %Operaciones para warning sound P17.

v=4; %Definición del código de velocidad del ensayo.
con=1; %Definición del número de tomas de datos que se realizaron del warning sound.
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga17\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora.
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional''.
run('..\Tarea 2\directividadizq.m') %Ejecución del script con nombre 'directividadizq''.

case 14 %Operaciones para warning sound P18.

v=4; %Definición del código de velocidad del ensayo.
con=1; %Definición del número de tomas de datos que se realizaron del warning sound.
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga18\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora.
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional''.
run('..\Tarea 2\directivaddcha.m') %Ejecución del script con nombre 'directivaddcha''.

case 15 %Operaciones para warning sound P19.

v=4; %Definición del código de velocidad del ensayo.
con=1; %Definición del número de tomas de datos que se realizaron del warning sound.
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga19\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora.
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional''.
run('..\Tarea 2\directividadizq.m') %Ejecución del script con nombre 'directividadizq''.

case 16 %Operaciones para warning sound P20.

v=4; %Definición del código de velocidad del ensayo.
con=1; %Definición del número de tomas de datos que se realizaron del warning sound.
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga20\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora.
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional''.
run('..\Tarea 2\directivaddcha.m') %Ejecución del script con nombre 'directivaddcha''.

otherwise %En el caso de que el código introducido no coincida con los casos anteriores se realiza lo siguiente.

display ('ERROR') %Escribir por pantalla error.
display ('Introduzca un valor dentro del intervalo') %Escribir por pantalla que se debe hacer para que no de error.
return %Salida de la estructura de control.

end %Cierre de la estructura de control switch para casos de Pasadas Largas.

otherwise %En el caso de que el código introducido no coincida con los casos anteriores se realiza lo siguiente.

display ('ERROR') %Escribir por pantalla error.
display ('Introduzca un valor dentro del intervalo') %Escribir por pantalla que se debe hacer para que no de error.
return %Salida de la estructura de control.

end %Cierre de la estructura de control switch para tipos de pasadas.

```

Figura 4.87. Definición de operaciones a realizar para Pasadas Largas en selectorpasadas para determinar la directividad vertical.

De nuevo, el bloque comienza mostrando los códigos que definen cada uno de los sonidos de advertencia. Tras ello, se pide cuál se desea estudiar, y se almacena la entrada que proporcione el usuario en una variable. Seguidamente, se establece una estructura de control “switch”, que contiene un caso por cada uno de los sonidos de advertencia que se estudiaron en las mediciones de Pasadas Largas.

Dentro de cada uno de los casos, se define la velocidad a la que fue realizada la recolección de datos y cuantas mediciones se hicieron. En segundo lugar, se importa la variable contenedora de la información de la pasada “tabppal”. A continuación, se lleva a cabo la ejecución del script “plottridimensional”. Tras esto, se ejecuta el programa “directividadizq” o “directividaddcha” en función del sentido de la marcha del vehículo en la medición. Finalmente, se establece un caso “otherwise” por la posible entrada de un código de warning sound erróneo.

Al final del programa se define otro “otherwise” pero esta vez para evitar la introducción de un código de tipo de pasada no definido.

#### **4.4.2.6. CONJUNTO DE PROGRAMAS DISTRIB + SONIDO DE ADVERTENCIA**

Suponiendo que el usuario desea investigar una señal de advertencia de las que se estudiaron haciendo Pasadas Cortas, se accederá al conjunto de programas “distrib + sonido de advertencia” que tiene un objetivo muy similar al tercer bloque del programa anteriormente analizado.



```
%% PETICIÓN DE VELOCIDAD A EJECUTAR PARA WARNING SOUND E1.
```

```
display('1-Velocidad a 5 km/h') %Mostrar por pantalla que código es la velocidad a 5 km/h.
display('2-Velocidad a 10 km/h') %Mostrar por pantalla que código es la velocidad a 10 km/h.
display('3-Velocidad a 15 km/h') %Mostrar por pantalla que código es la velocidad a 15 km/h.
display('4-Velocidad a 20 km/h') %Mostrar por pantalla que código es la velocidad a 20 km/h.
display('5-Velocidad a 25 km/h') %Mostrar por pantalla que código es la velocidad a 25 km/h.
display('6-Velocidad a 30 km/h') %Mostrar por pantalla que código es la velocidad a 30 km/h.
```

```
v=input('¿Qué velocidad quiere estudiar? '); %Pregunta por pantalla código de velocidad a estudiar y almacenamiento en variable.
```

```
%% LISTADO VELOCIDADES PARA WARNING SOUND E1.
```

```
switch v %Estructura de control switch que realiza las funciones que se definan para cada caso de velocidad para warning sound E1.
```

```
case 1 %Operaciones para velocidad de 5 km/h.
```

```
arch=dir('C1 E1 5\*Photo2Dcent.txt'); %Obtención de lista de archivos con extensión .m para 5 km/h.
numarch=length(arch); %Número de archivos que permiten obtener los datos con extensión .m para 5 km/h.
for con=1:numarch %Bucle for para ejecutar los distintos ensayos para la velocidad definida anteriormente.
```

```
switch con %Estructura de control switch que realice las funciones especificadas para cada ensayo de 5 km/h.
```

```
case 1 %Operaciones para primer ensayo a 5 km/h.
```

```
load('..\E1\C1 E1 5\Variables CIE15uno\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora del primer ensayo.
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional'.
run('..\Tarea 2\directividadizq.m') %Ejecución del script con nombre 'directividadizq'.
clear %Limpieza del espacio de trabajo.
clc %Limpieza de variables.
pas=1; %Redefinición de variable pasada con valor correspondiente.
con=1; %Redefinición de variable contador con valor correspondiente.
ws=1; %Redefinición de variable warning sound con valor correspondiente.
v=1; %Redefinición de variable velocidad con valor correspondiente.
```

```
case 2 %Operaciones para segundo ensayo a 5 km/h.
```

```
load('..\E1\C1 E1 5\Variables CIE15dos\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora del segundo ensayo.
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional'.
run('..\Tarea 2\directividaddcha.m') %Ejecución del script con nombre 'directividaddcha'.
```

```
end %Cierre de la estructura de control switch para ensayos de la primera velocidad.
```

```
end %Cierre del bucle que ejecuta los distintos ensayos.
```

```
case 2 %Operaciones para velocidad de 10 km/h.
```

```
arch=dir('C1 E1 10\*Photo2Dcent.txt'); %Obtención de lista de archivos con extensión .m para 10 km/h.
numarch=length(arch); %Número de archivos que permiten obtener los datos con extensión .m para 10 km/h.
```

```
for con=1:numarch %Bucle for para ejecutar los distintos ensayos para la velocidad definida anteriormente.
```

```
switch con %Estructura de control switch que realice las funciones especificadas para cada ensayo de 10 km/h.
```

```
case 1 %Operaciones para primer ensayo a 10 km/h.
```

```
load('..\E1\C1 E1 10\Variables CIE10uno\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora del primer ensayo.
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional'.
run('..\Tarea 2\directividadizq.m') %Ejecución del script con nombre 'directividadizq'.
```

```
end %Cierre de la estructura de control switch para ensayos de la segunda velocidad.
```

```
end %Cierre del bucle que ejecuta los distintos ensayos.
```

```
case 3 %Operaciones para velocidad de 15 km/h.
```

```
arch=dir('C1 E1 15\*Photo2Dcent.txt'); %Obtención de lista de archivos con extensión .m para 15 km/h.
numarch=length(arch); %Número de archivos que permiten obtener los datos con extensión .m para 15 km/h.
```

```
for con=1:numarch %Bucle for para cambiar el valor de i (i es el número de ensayo) para esa velocidad.
```

```
switch con %Estructura de control switch que realice las funciones especificadas para cada ensayo de 15 km/h.
```

```
case 1 %Operaciones para primer ensayo a 15 km/h.
```

```
load('..\E1\C1 E1 15\Variables CIE15uno\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora del primer ensayo.
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional'.
run('..\Tarea 2\directividadizq.m') %Ejecución del script con nombre 'directividadizq'.
clear %Limpieza del espacio de trabajo.
clc %Limpieza de variables.
pas=1; %Redefinición de variable pasada con valor correspondiente.
con=1; %Redefinición de variable contador con valor correspondiente.
ws=1; %Redefinición de variable warning sound con valor correspondiente.
v=3; %Redefinición de variable velocidad con valor correspondiente.
```

```
case 2 %Operaciones para segundo ensayo a 15 km/h.
```

```
load('..\E1\C1 E1 15\Variables CIE15dos\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora del segundo ensayo.
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional'.
run('..\Tarea 2\directividaddcha.m') %Ejecución del script con nombre 'directividaddcha'.
```

```
end %Cierre de la estructura de control switch para ensayos de la tercera velocidad.
```

```
end %Cierre del bucle que ejecuta los distintos ensayos.
```

```

case 4 %Operaciones para velocidad de 20 km/h.

arch=dir('C1 E1 20\*Photo2Dcent.txt'); % Obtención de lista de archivos con extensión .m para 20 km/h.
numarch=length(arch); % Número de archivos que permiten obtener los datos con extensión .m para 20 km/h.

for con=1:numarch %Estructura de control switch que realice las funciones especificadas para cada ensayo de 20 km/h.

    switch con %Estructura de control switch que realice las funciones especificadas para cada ensayo de 20 km/h.

        case 1 %Operaciones para primer ensayo a 20 km/h.

            load('..\E1\C1 E1 20\Variables C1E120uno\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora del primer ensayo.
            run('..\Varea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional'.
            run('..\Varea 2\directividadadizq.m') %Ejecución del script con nombre 'directividadadizq'
            clear %Limpieza del espacio de trabajo.
            clc %Limpieza de variables.
            pas=1; %Redefinición de variable pasada con valor correspondiente.
            con=1; %Redefinición de variable contador con valor correspondiente.
            ws=1; %Redefinición de variable warning sound con valor correspondiente.
            v=4; %Redefinición de variable velocidad con valor correspondiente.

        case 2 %Operaciones para segundo ensayo a 20 km/h.

            load('..\E1\C1 E1 20\Variables C1E120dos\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora del segundo ensayo.
            run('..\Varea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional'.
            run('..\Varea 2\directividadadcha.m') %Ejecución del script con nombre 'directividadadcha'.

    end %Cierre de la estructura de control switch para ensayos de la cuarta velocidad.

end %Cierre del bucle que ejecuta los distintos ensayos.

case 5 %Operaciones para velocidad de 25 km/h.

arch=dir('C1 E1 25\*Photo2Dcent.txt'); % Obtención de lista de archivos con extensión .m para 25 km/h.
numarch=length(arch); % Número de archivos que permiten obtener los datos con extensión .m para 25 km/h.

for con=1:numarch %Estructura de control switch que realice las funciones especificadas para cada ensayo de 25 km/h.

    switch con %Estructura de control switch que realice las funciones especificadas para cada ensayo de 25 km/h.

        case 1 %Operaciones para primer ensayo a 25 km/h.

            load('..\E1\C1 E1 25\Variables C1E125uno\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora del primer ensayo.
            run('..\Varea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional'.
            run('..\Varea 2\directividadadizq.m') %Ejecución del script con nombre 'directividadadizq'
            clear %Limpieza del espacio de trabajo.
            clc %Limpieza de variables.
            pas=1; %Redefinición de variable pasada con valor correspondiente.
            con=1; %Redefinición de variable contador con valor correspondiente.
            ws=1; %Redefinición de variable warning sound con valor correspondiente.
            v=5; %Redefinición de variable velocidad con valor correspondiente.

        case 2 %Operaciones para segundo ensayo a 25 km/h.

            load('..\E1\C1 E1 25\Variables C1E125dos\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora del segundo ensayo.
            run('..\Varea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional'.
            run('..\Varea 2\directividadadcha.m') %Ejecución del script con nombre 'directividadadcha'.

    end %Cierre de la estructura de control switch para ensayos de la quinta velocidad.

end %Cierre del bucle que ejecuta los distintos ensayos.

case 6 %Operaciones para velocidad de 30 km/h.

arch=dir('C1 E1 30\*Photo2Dcent.txt'); % Obtención de lista de archivos con extensión .m para 30 km/h.
numarch=length(arch); % Número de archivos que permiten obtener los datos con extensión .m para 30 km/h.

for con=1:numarch %Estructura de control switch que realice las funciones especificadas para cada ensayo de 30 km/h.

    switch con %Estructura de control switch que realice las funciones especificadas para cada ensayo de 30 km/h.

        case 1 %Operaciones para primer ensayo a 30 km/h.

            load('..\E1\C1 E1 30\Variables C1E130uno\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora del primer ensayo.
            run('..\Varea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional'.
            run('..\Varea 2\directividadadizq.m') %Ejecución del script con nombre 'directividadadizq'
            clear %Limpieza del espacio de trabajo.
            clc %Limpieza de variables.
            pas=1; %Redefinición de variable pasada con valor correspondiente.
            con=1; %Redefinición de variable contador con valor correspondiente.
            ws=1; %Redefinición de variable warning sound con valor correspondiente.
            v=6; %Redefinición de variable velocidad con valor correspondiente.

        case 2 %Operaciones para segundo ensayo a 30 km/h.

            load('..\E1\C1 E1 30\Variables C1E130dos\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora del segundo ensayo.
            run('..\Varea 2\plottridimensional.m') %Ejecución del script con nombre 'plottridimensional'.
            run('..\Varea 2\directividadadcha.m') %Ejecución del script con nombre 'directividadadcha'.

    end %Cierre de la estructura de control switch para ensayos de la sexta velocidad.

end %Cierre del bucle que ejecuta los distintos ensayos.

otherwise %En el caso de que el código introducido no coincida con los casos anteriores se realiza lo anterior.

    display('ERROR') %Escribir por pantalla error.
    display('Introduzca un valor dentro del intervalo especificado') %Escribir por pantalla que se debe hacer para que no de error.
    return %Salida de la estructura de control.

end %Cierre de la estructura de control switch para casos del warning sound E1.

```

Figura 4.88. Programa distribe1.

El programa comienza mostrando por pantalla el código que representa cada una de las velocidades a las que se tomaron datos con el sonido de advertencia seleccionado. Posteriormente, se pide el código de velocidad que se desea estudiar y se almacena en una variable. Después, se establece un “switch” donde cada uno de los casos de la estructura de control se refiere a cada una de las velocidades a las que circuló el vehículo.

En el interior de cada caso, en primer lugar, se determinan el nombre de archivos con extensión .txt por medio del comando “dir” que existen en la carpeta de la velocidad a la que se refiera y el número de archivos con dicha extensión.

A continuación, se define un bucle “for” cuyo contador cambia los valores de uno hasta el número total de archivos con extensión .txt que existan en la carpeta de la velocidad estudiada para el sonido de advertencia seleccionado. Seguidamente, se estructura otro “switch”, con un número de casos igual a la cantidad de mediciones realizadas en esa velocidad.

Para aquellas mediciones que no se correspondan con el último caso, se lleva a cabo la transferencia de los datos de la variable contenedora de información “tabppal”, la ejecución de “plottridimensional” y la ejecución de “directividadizq”/“directivaddcha” en función del sentido de la marcha. Después, se lleva a cabo una limpieza del espacio de trabajo y de la ventana de comandos, para evitar interferencias entre casos.

Tras la limpieza del espacio de trabajo y de la ventana de comandos, se redefinen una serie de variables como son: el sonido de advertencia elegido, la velocidad estudiada, el código que designa el tipo de pasada y el número asignado a la toma de datos. La razón de redefinir las variables se debe a que, al limpiar el espacio de trabajo, este se queda sin ningún tipo de variable, por lo que, para que el programa pueda continuar trabajando con el siguiente ensayo, se debe volver a establecer ciertos parámetros para que no se produzca un error.

En cambio, si se ejecuta el último caso de la velocidad estudiada, basta con extraer los datos de “tabppal”, ejecutar “plottridimensional” y ejecutar “directividadizq”/“directivaddcha” según el caso. Esto se debe a que, al no ser necesario proseguir analizando más mediciones, no es necesario llevar a cabo la limpieza del espacio de trabajo y la ventana de comandos.

Lo último que presenta el programa es un caso extra para la estructura de control “switch”. Este se activará si el código introducido no se corresponde a ninguno del listado ofrecido.

De esta forma se concluye la explicación de todos los programas para la obtención de información acerca de la directividad vertical de los ensayos. No obstante, con el objetivo de

clarificar el modo en que los programas se interrelacionan y trabajan, se proporcionan los siguientes esquemas.

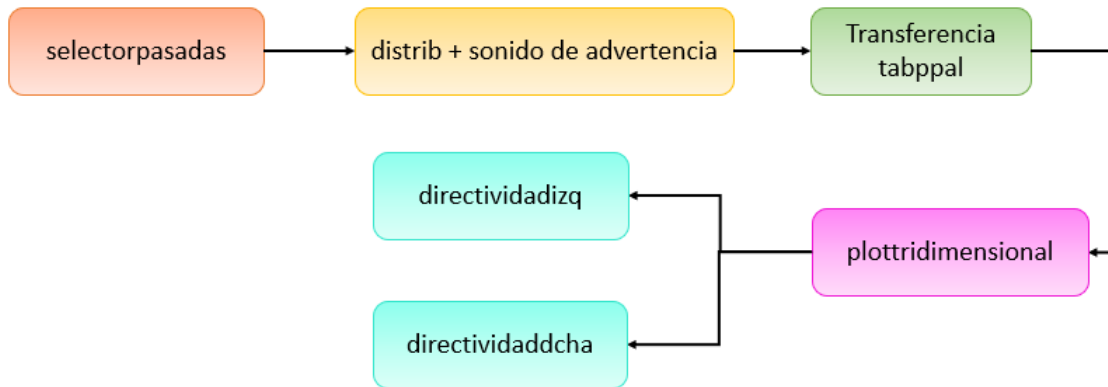


Figura 4.89. Esquema relacional para Pasadas Cortas.

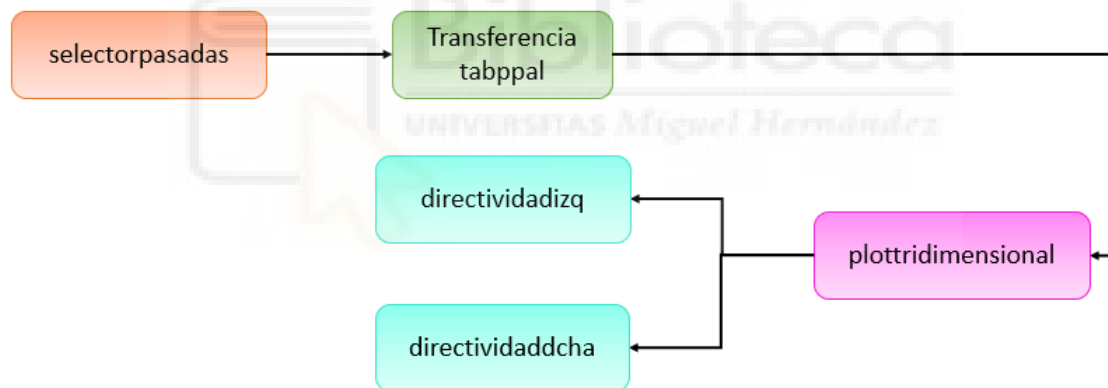


Figura 4.90. Esquema relacional para Pasadas Largas.

## 4.5. RESULTADOS

En este apartado se realizarán distintas tareas. La primera de ellas será disponer de los resultados que se han obtenido a través de la manipulación de la información por medio de los programas desarrollados en Matlab. Tras esto, se analizarán los resultados con el objetivo de extraer toda la información posible acerca del comportamiento de los sonidos de advertencia.

### 4.5.1. NUMÉRICO

Aquí se disponen los resultados que se han ido extrayendo al ejecutar los distintos programas generados.

#### 4.5.1.1. NIVEL SONORO CONTINUO EQUIVALENTE MÁXIMO

##### 4.5.1.1.1. GRÁFICA GLOBAL

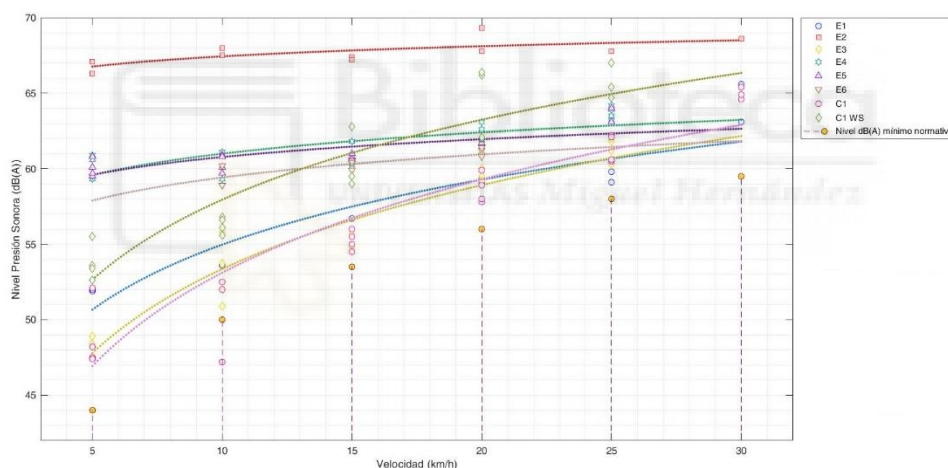


Figura 4.91. Gráfica donde se representa el nivel sonoro equivalente máximo para cada velocidad de todos los sonidos de advertencia.

**4.5.1.1.2. ESPECTRO DE FRECUENCIA**

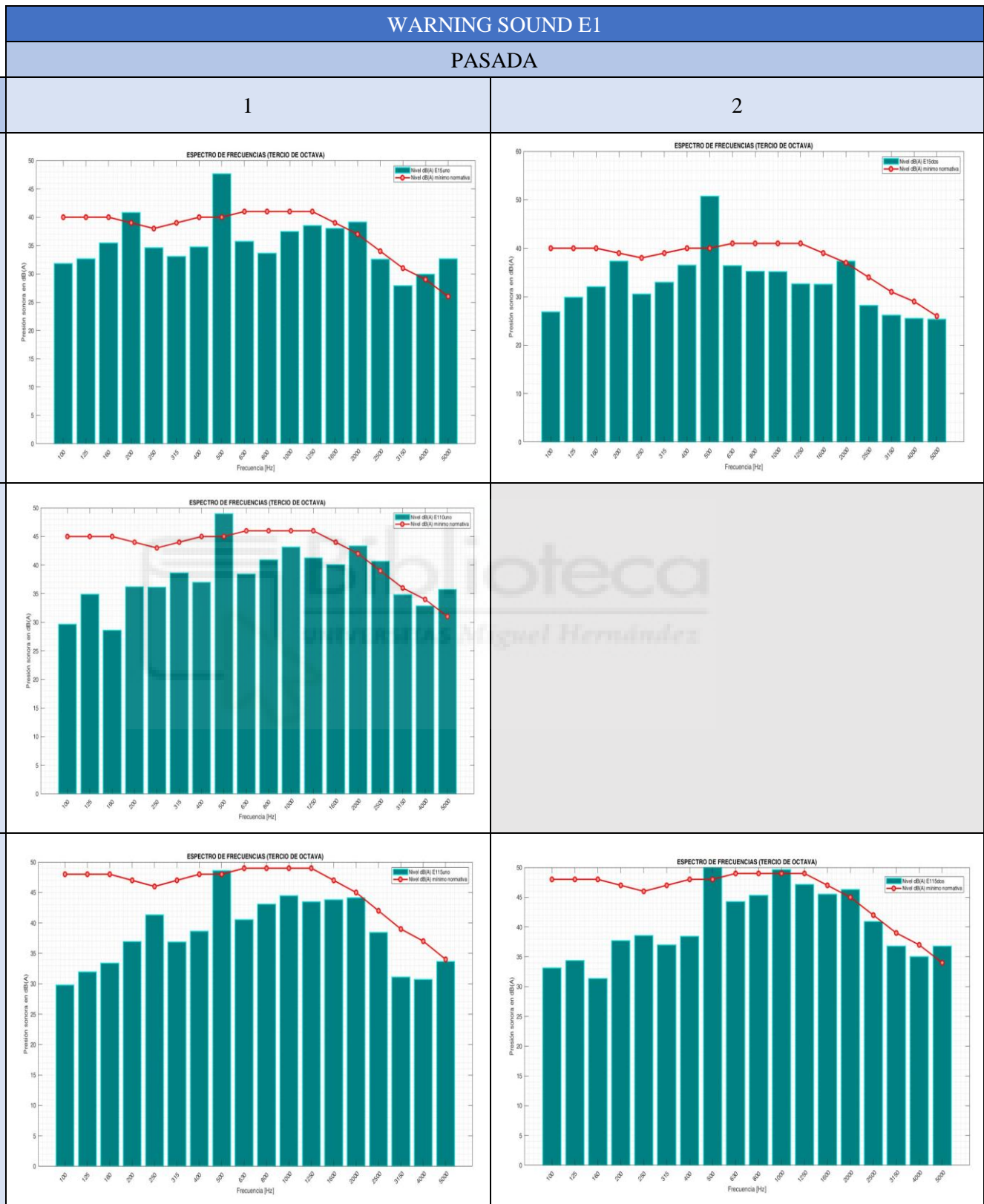


Tabla 4.02. Espectros de frecuencia para velocidades de 5-15 km/h para E1.

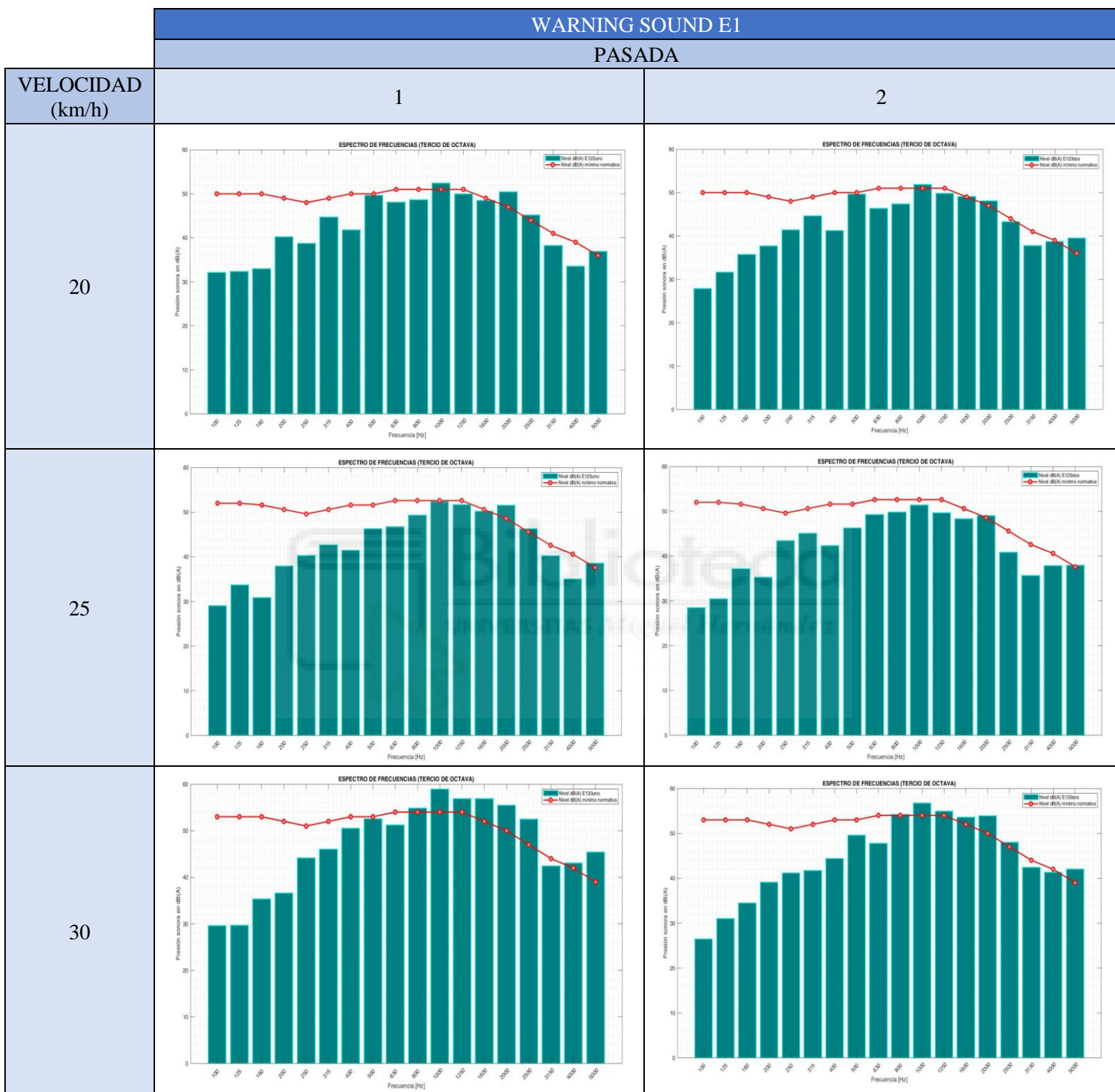


Tabla 4.03. Espectros de frecuencia para velocidades de 20-30 km/h para E1.

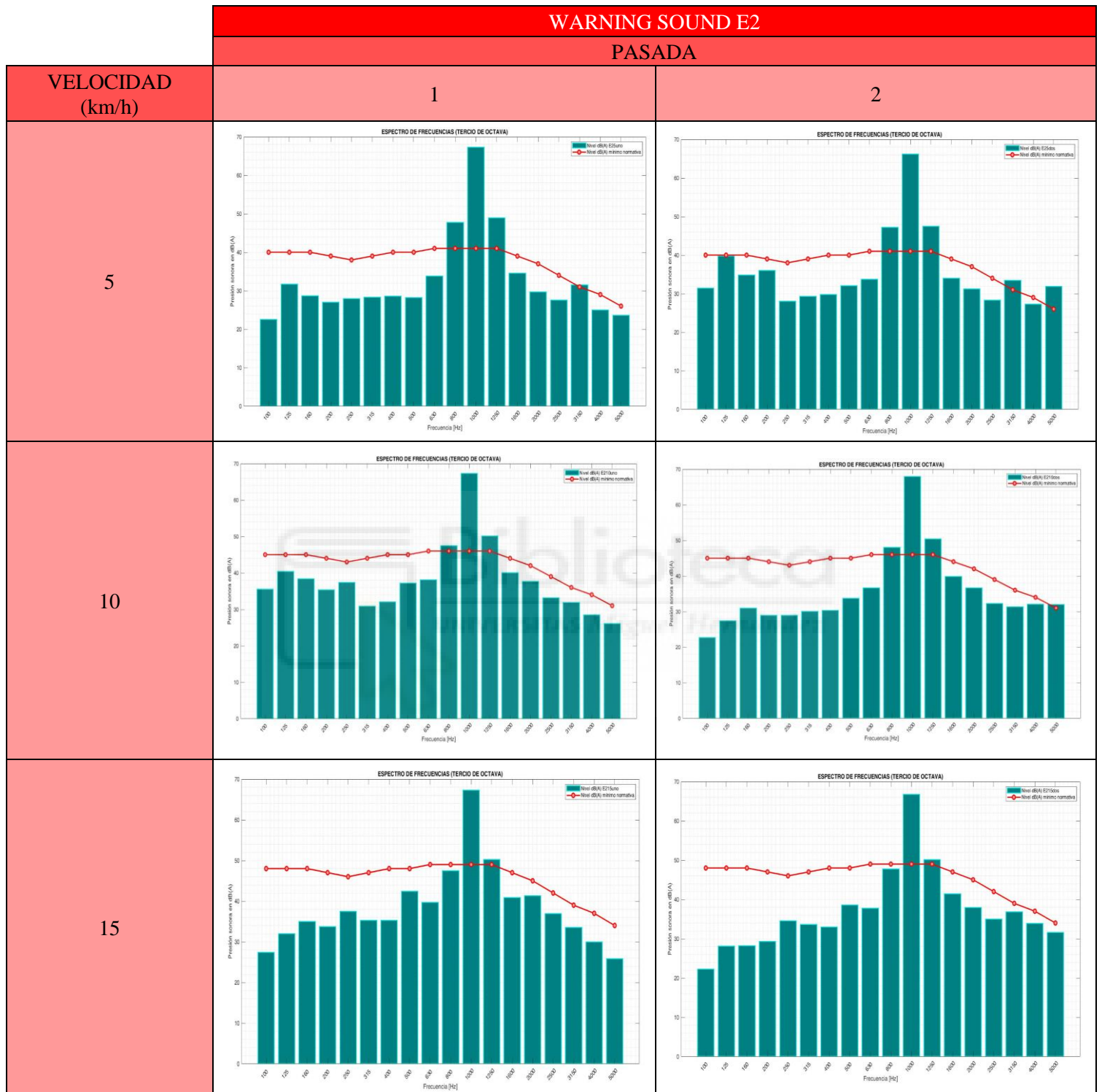


Tabla 4.04. Espectros de frecuencia para velocidades de 5-15 km/h para E2.



WARNING SOUND E2		
PASADA		
VELOCIDAD (km/h)	1	2
20	<p>ESPECTRO DE FRECUENCIAS (TERCIO DE OCTAVA)</p> <p>Gráfico de barras y líneas que muestra el espectro de frecuencias (tercio de octava) para una velocidad de 20 km/h en la posición 1. El eje vertical representa la presión sonora en dB(A) (0-70) y el eje horizontal la frecuencia en Hz (100-5000). Se comparan el nivel dB(A) E200ms (barras verdes) y el nivel dB(A) métrico normalizado (línea roja con círculos). El nivel dB(A) E200ms muestra un pico de aproximadamente 70 dB(A) a 1000 Hz.</p>	<p>ESPECTRO DE FRECUENCIAS (TERCIO DE OCTAVA)</p> <p>Gráfico de barras y líneas que muestra el espectro de frecuencias (tercio de octava) para una velocidad de 20 km/h en la posición 2. El eje vertical representa la presión sonora en dB(A) (0-70) y el eje horizontal la frecuencia en Hz (100-5000). Se comparan el nivel dB(A) E200ms (barras verdes) y el nivel dB(A) métrico normalizado (línea roja con círculos). El nivel dB(A) E200ms muestra un pico de aproximadamente 68 dB(A) a 1000 Hz.</p>
25		<p>ESPECTRO DE FRECUENCIAS (TERCIO DE OCTAVA)</p> <p>Gráfico de barras y líneas que muestra el espectro de frecuencias (tercio de octava) para una velocidad de 25 km/h en la posición 2. El eje vertical representa la presión sonora en dB(A) (0-70) y el eje horizontal la frecuencia en Hz (100-5000). Se comparan el nivel dB(A) E200ms (barras verdes) y el nivel dB(A) métrico normalizado (línea roja con círculos). El nivel dB(A) E200ms muestra un pico de aproximadamente 68 dB(A) a 1000 Hz.</p>
30	<p>ESPECTRO DE FRECUENCIAS (TERCIO DE OCTAVA)</p> <p>Gráfico de barras y líneas que muestra el espectro de frecuencias (tercio de octava) para una velocidad de 30 km/h en la posición 1. El eje vertical representa la presión sonora en dB(A) (0-70) y el eje horizontal la frecuencia en Hz (100-5000). Se comparan el nivel dB(A) E200ms (barras verdes) y el nivel dB(A) métrico normalizado (línea roja con círculos). El nivel dB(A) E200ms muestra un pico de aproximadamente 68 dB(A) a 1000 Hz.</p>	<p>ESPECTRO DE FRECUENCIAS (TERCIO DE OCTAVA)</p> <p>Gráfico de barras y líneas que muestra el espectro de frecuencias (tercio de octava) para una velocidad de 30 km/h en la posición 2. El eje vertical representa la presión sonora en dB(A) (0-70) y el eje horizontal la frecuencia en Hz (100-5000). Se comparan el nivel dB(A) E200ms (barras verdes) y el nivel dB(A) métrico normalizado (línea roja con círculos). El nivel dB(A) E200ms muestra un pico de aproximadamente 68 dB(A) a 1000 Hz.</p>

Tabla 4.05. Espectros de frecuencia para velocidades de 20-30 km/h para E2.

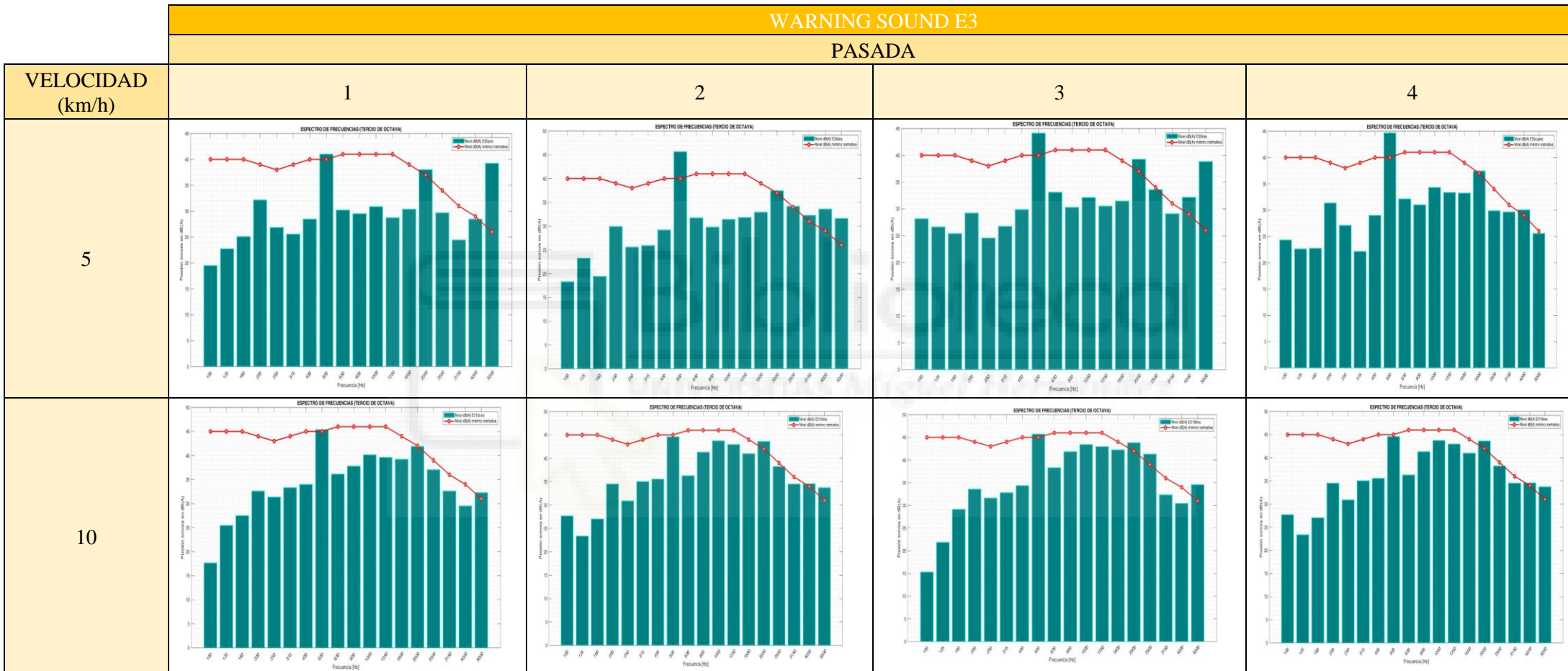


Tabla 4.06. Espectros de frecuencia para velocidades de 5-10 km/h para E3.

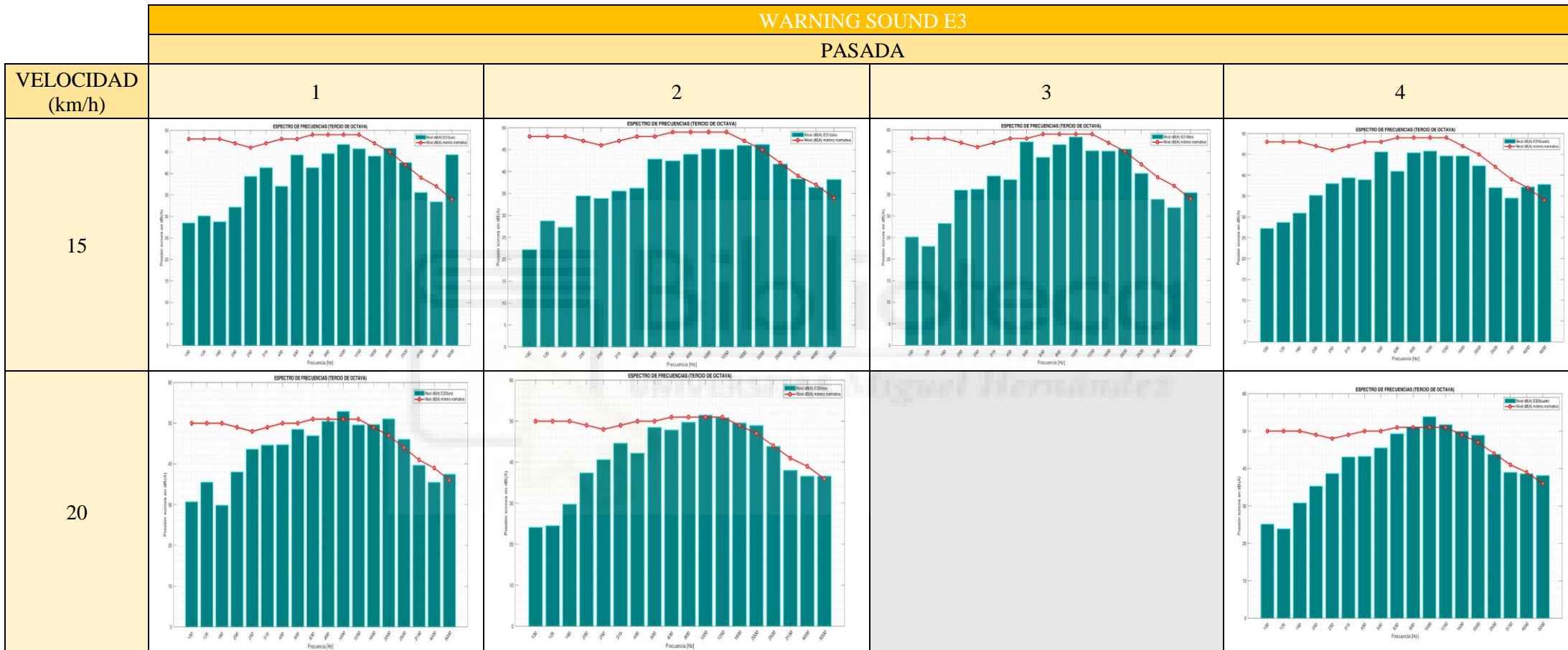


Tabla 4.07. Espectros de frecuencia para velocidades de 15-20 km/h para E3.

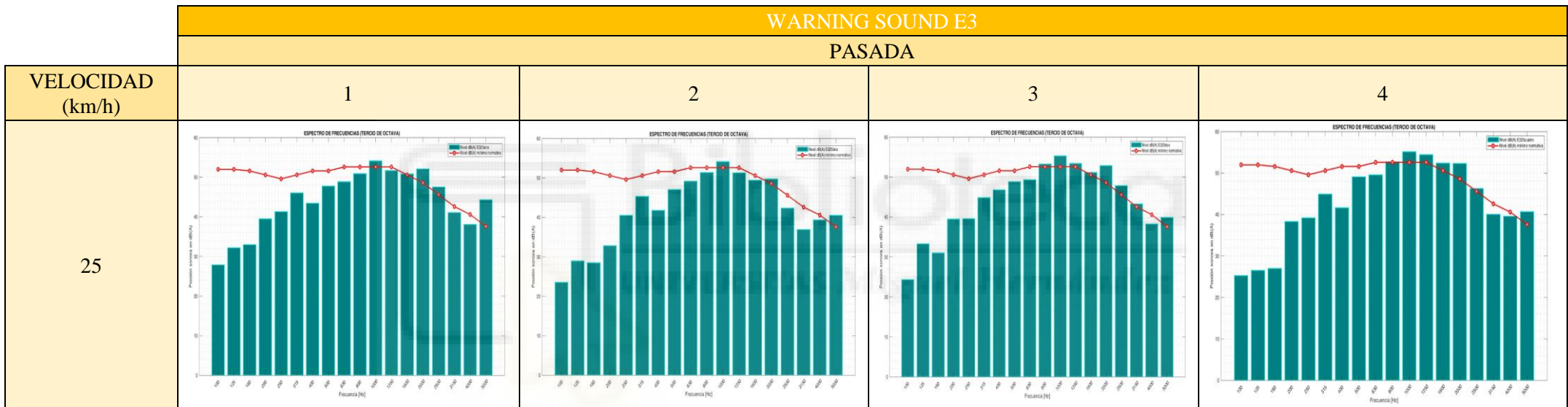


Tabla 4.08. Espectros de frecuencia para velocidad de 25 km/h para E3.



Tabla 4.09. Espectros de frecuencia para velocidades de 5-10 km/h para E4.

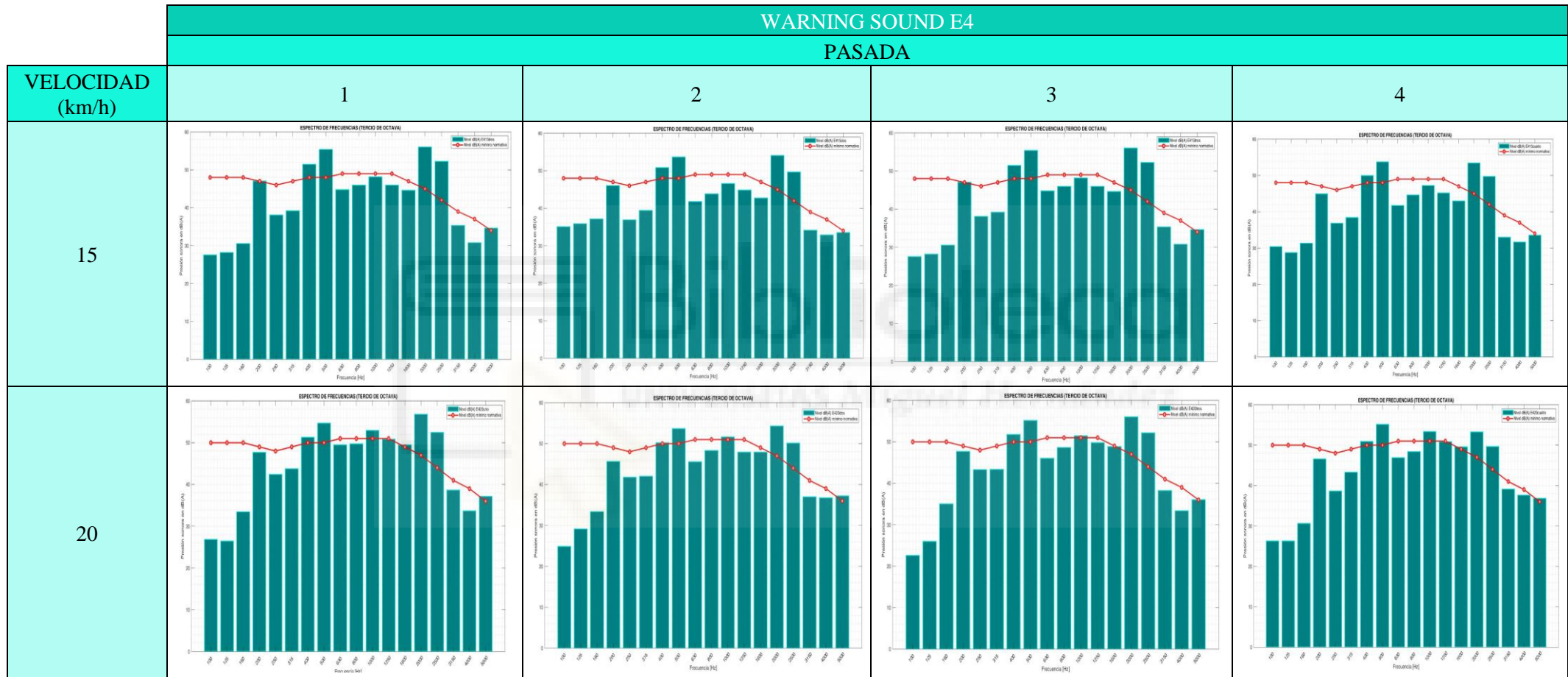


Tabla 4.10. Espectros de frecuencia para velocidades de 15-20 km/h para E4.

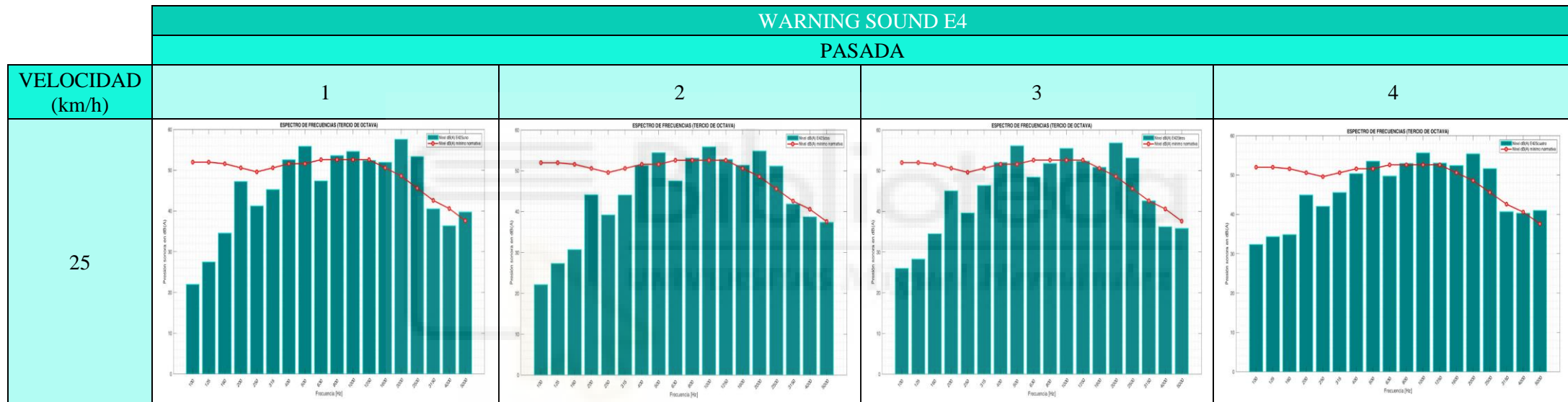


Tabla 4.11. Espectros de frecuencia para velocidad de 25 km/h para E4.

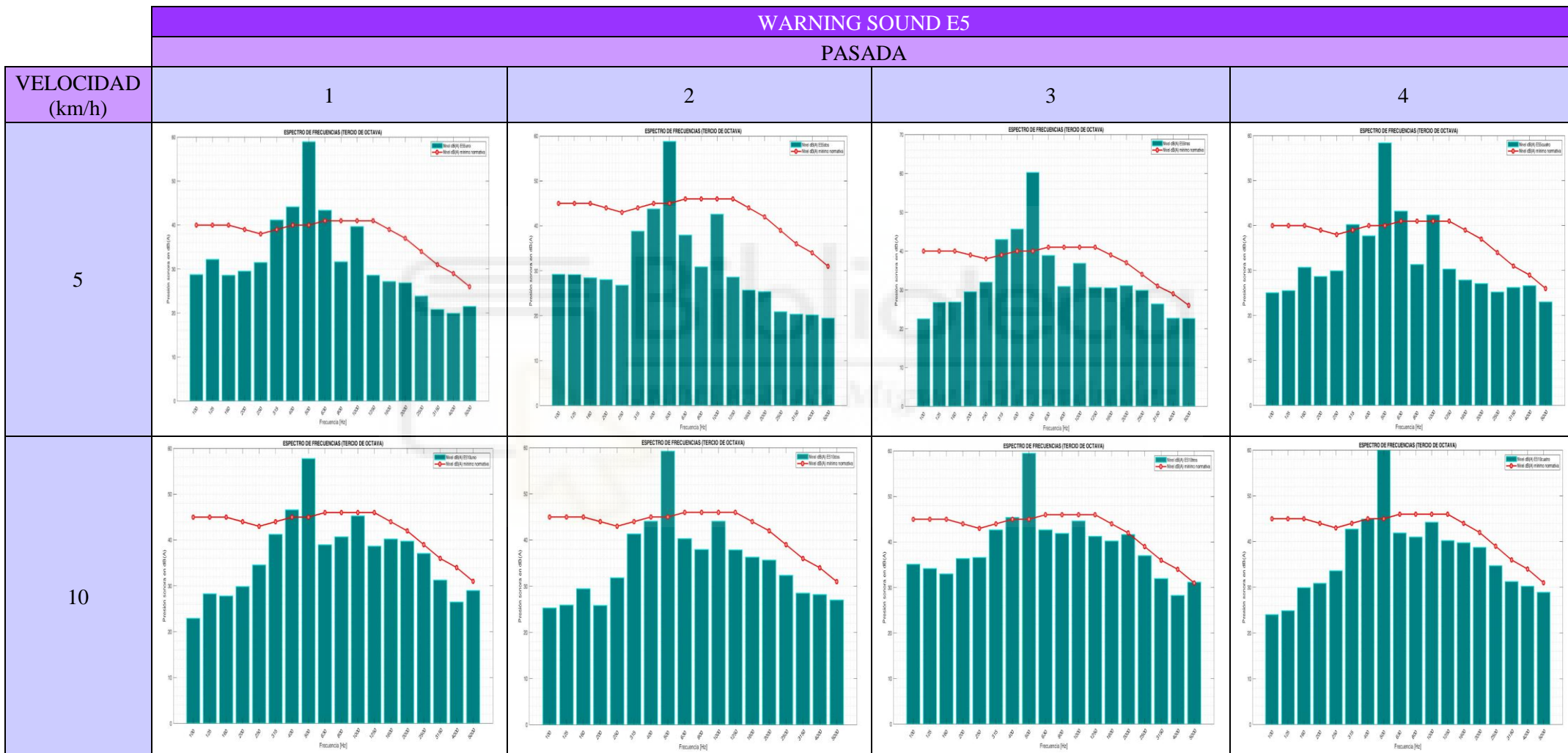


Tabla 4.12. Espectros de frecuencia para velocidades de 5-10 km/h para E5.



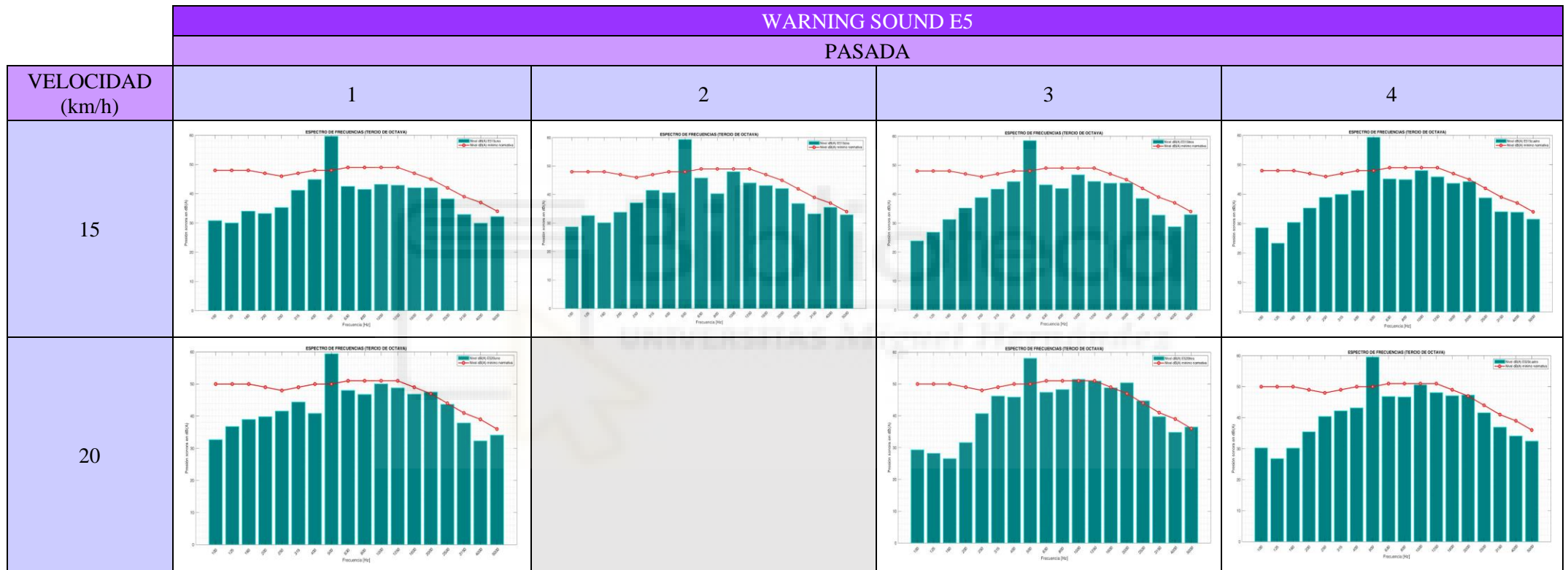


Tabla 4.13. Espectros de frecuencia para velocidades de 15-20 km/h para E5.

		WARNING SOUND E5			
		PASADA			
VELOCIDAD (km/h)		1	2	3	4
25					

Tabla 4.14. Espectros de frecuencia para velocidad 25 km/h para E5.

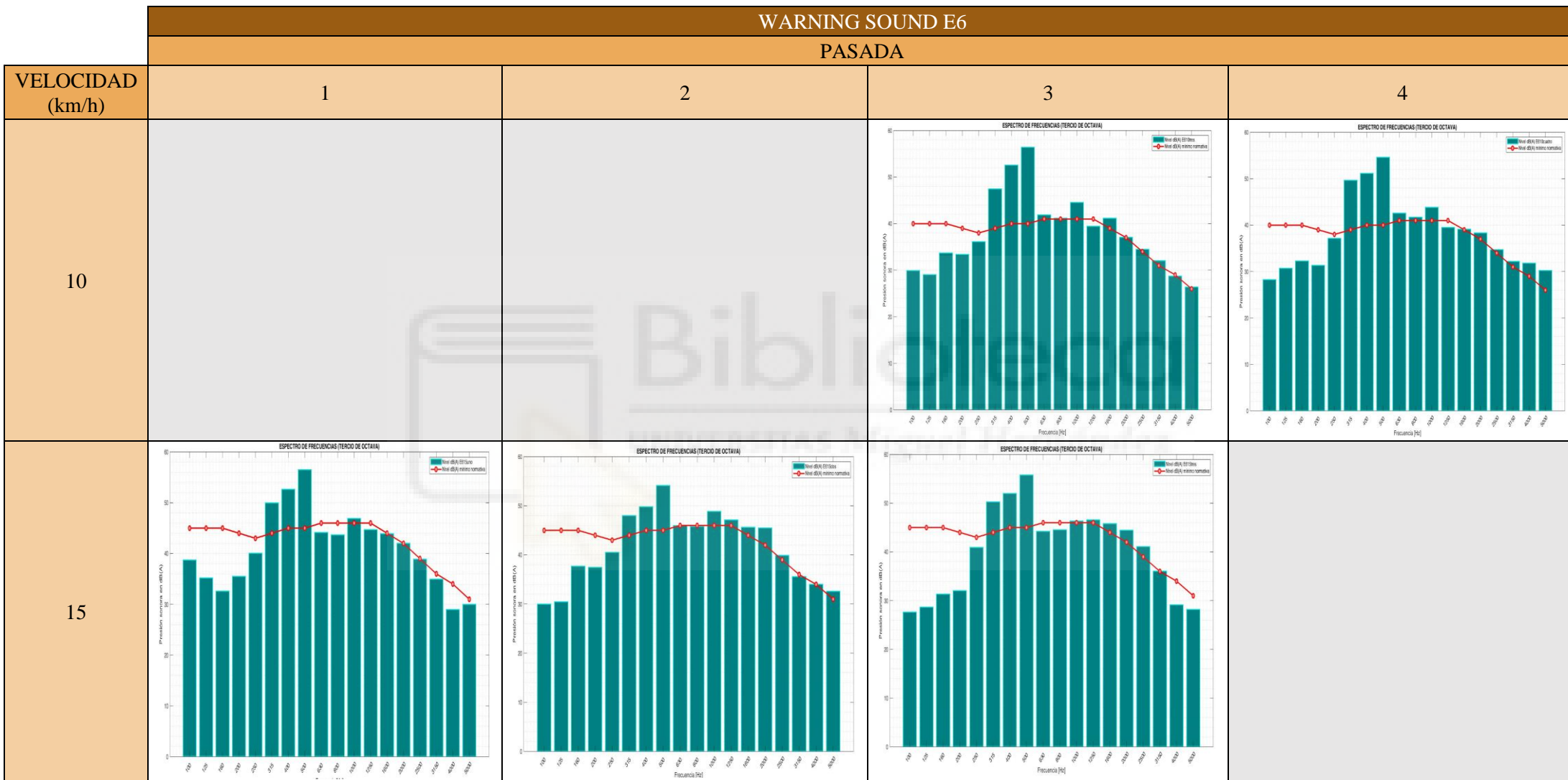


Tabla 4.15. Espectros de frecuencia para velocidades de 10-15 km/h para E6.

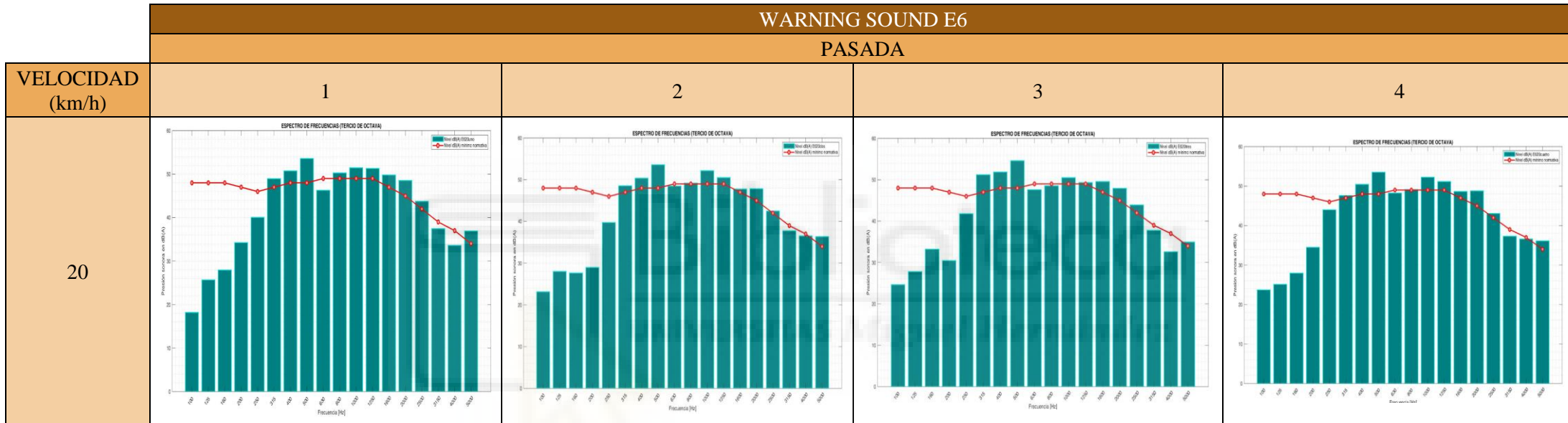


Tabla 4.16. Espectros de frecuencia para velocidad 20 km/h para E6.

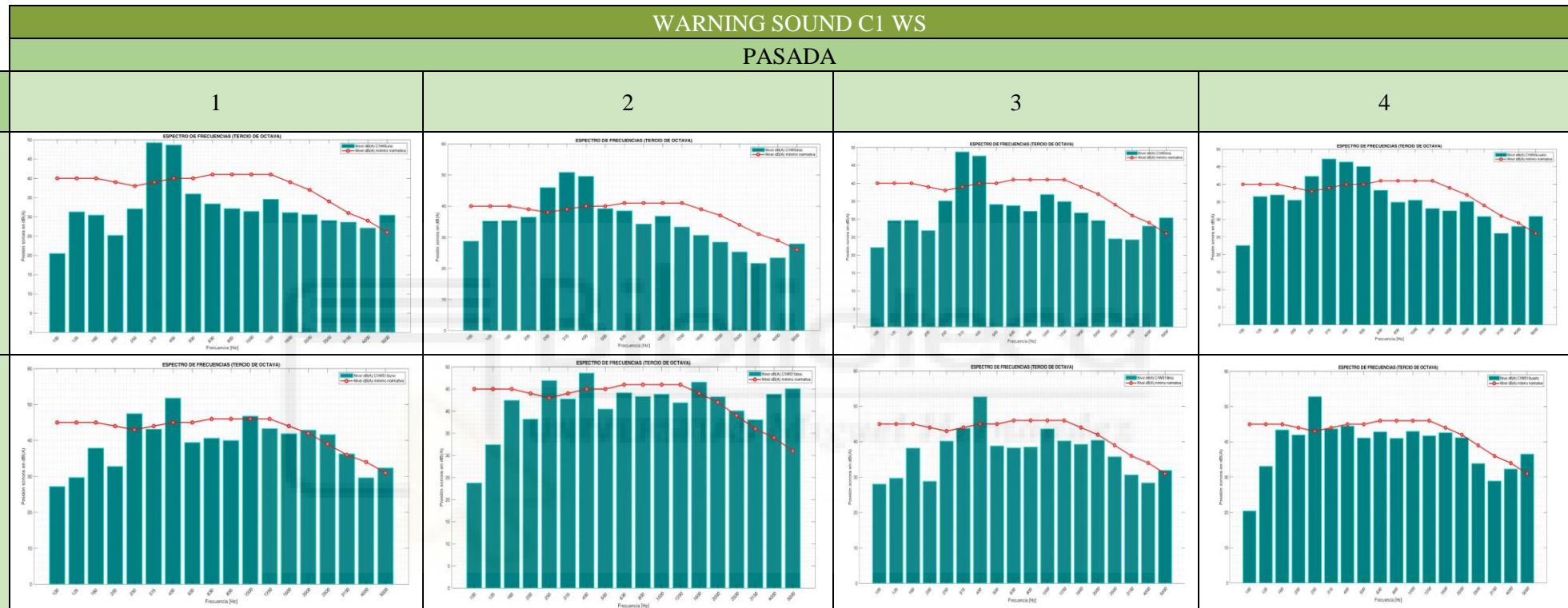


Tabla 4.17. Espectros de frecuencia para velocidades de 5-10 km/h para C1 WS.

WARNING SOUND C1 WS

PASADA

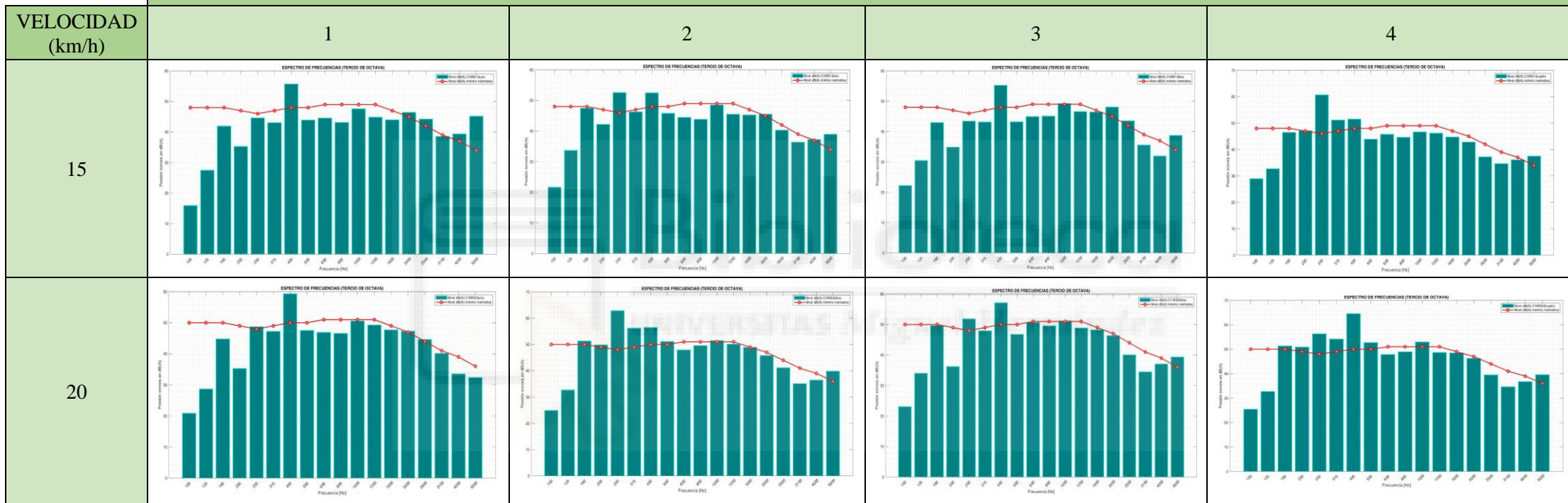


Tabla 4.18. Espectros de frecuencia para velocidades de 15-20 km/h para C1 WS.

WARNING SOUND C1 WS				
PASADA				
VELOCIDAD (km/h)	1	2	3	4
25				

Tabla 4.19. Espectros de frecuencia para velocidad de 25 km/h para C1 WS.

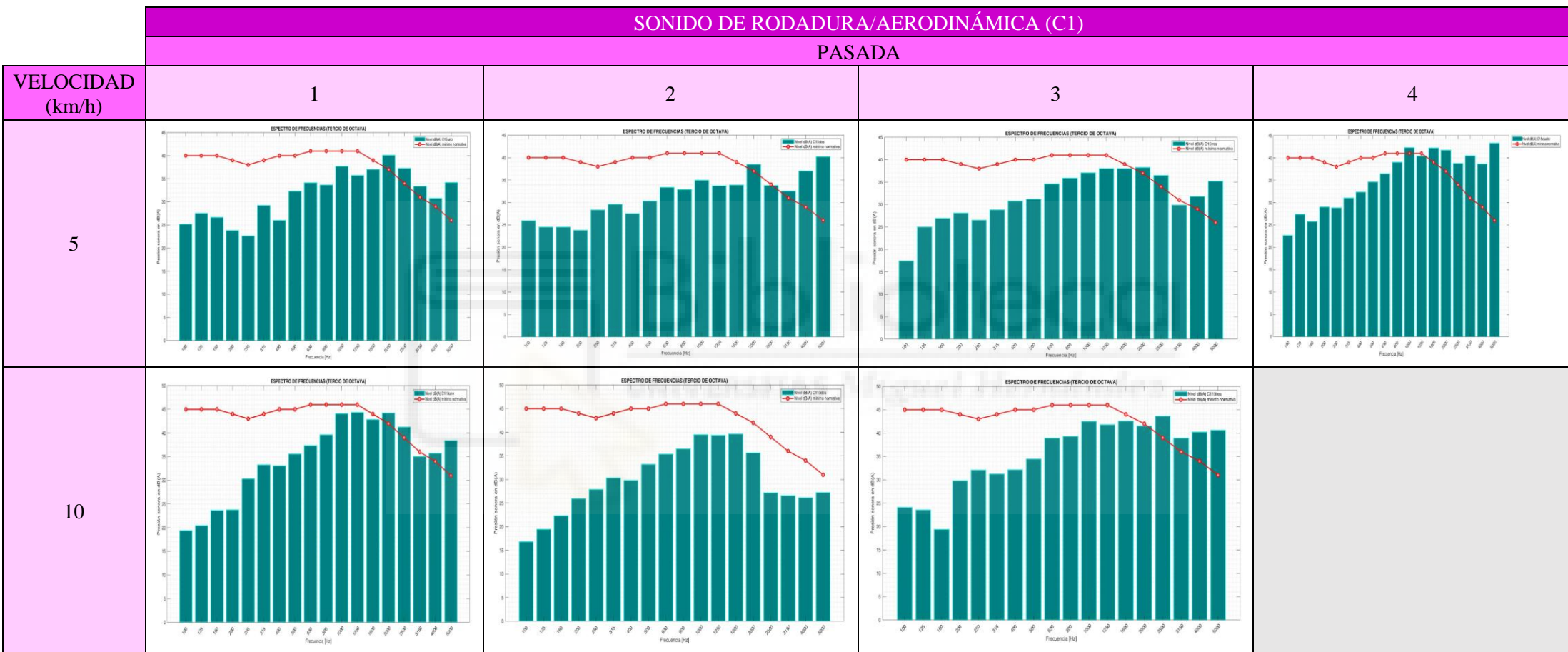


Tabla 4.20. Espectros de frecuencia para velocidades de 5-10 km/h para C1.



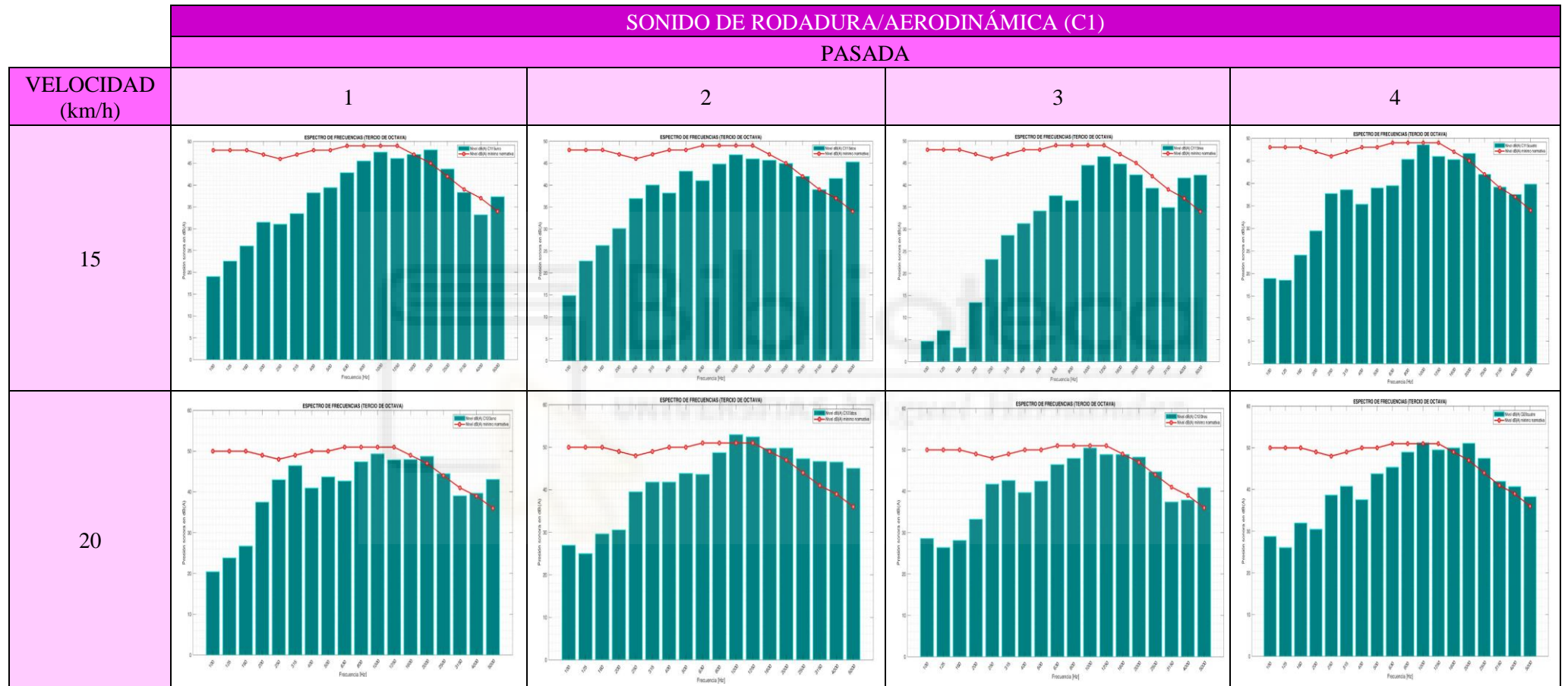


Tabla 4.21. Espectros de frecuencia para velocidades de 15-20 km/h para E6.

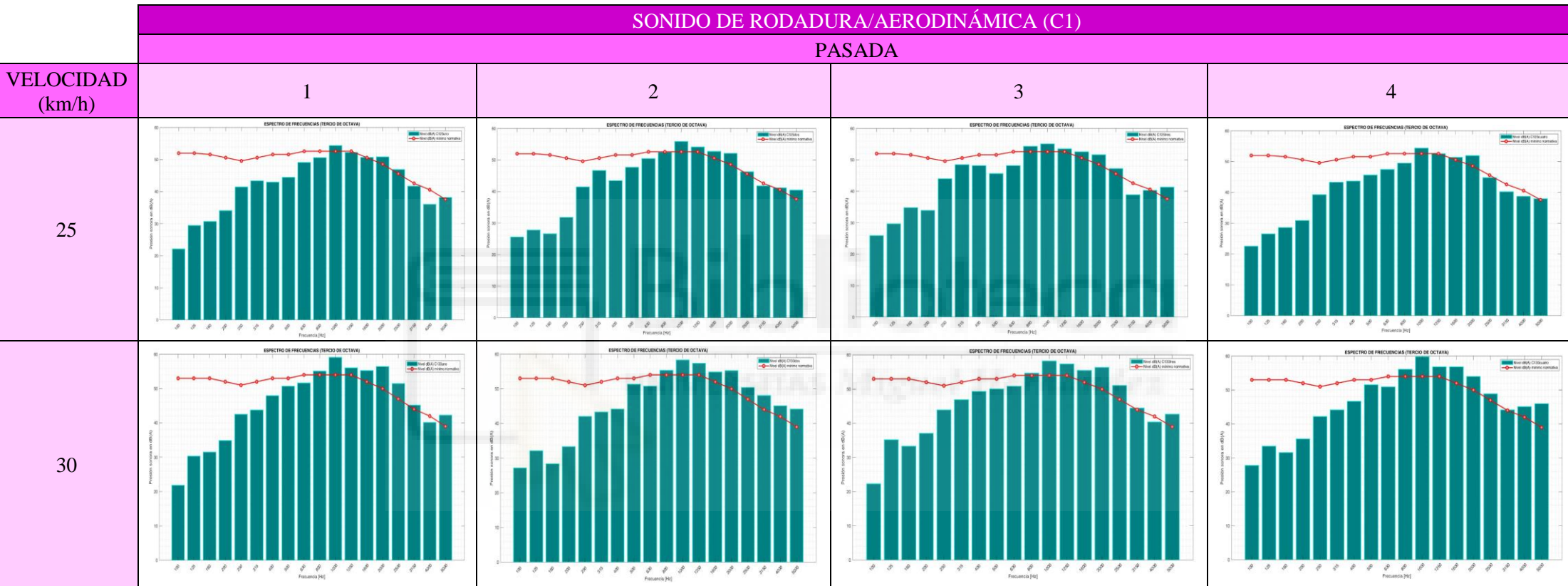


Tabla 4.22. Espectros de frecuencia para velocidades de 25-30 km/h para C1.

**4.5.1.2. DIRECTIVIDAD VERTICAL**

		WARNING SOUND E1	
		PASADA	
VELOCIDAD (km/h)		1	2
5			
10			
15			

Tabla 4.23. Directividad vertical para velocidades de 5-15 km/h para E1.

WARNING SOUND E1		
PASADA		
VELOCIDAD (km/h)	1	2
20		
25		
30		

Tabla 4.24. Directividad vertical para velocidades de 20-30 km/h para E1.

		WARNING SOUND E2	
		PASADA	
VELOCIDAD (km/h)		1	2
5		<p>DIRECTIVIDAD (DI)</p>	<p>DIRECTIVIDAD (DI)</p>
10		<p>DIRECTIVIDAD (DI)</p>	<p>DIRECTIVIDAD (DI)</p>
15		<p>DIRECTIVIDAD (DI)</p>	<p>DIRECTIVIDAD (DI)</p>

Tabla 4.25. Directividad vertical para velocidades de 5-15 km/h para E2.

		WARNING SOUND E2	
		PASADA	
VELOCIDAD (km/h)		1	2
20		<p>DIRECTIVIDAD (DI)</p>	<p>DIRECTIVIDAD (DI)</p>
25			<p>DIRECTIVIDAD (DI)</p>
30		<p>DIRECTIVIDAD (DI)</p>	<p>DIRECTIVIDAD (DI)</p>

Tabla 4.26. Directividad vertical para velocidades de 20-30 km/h para E2.

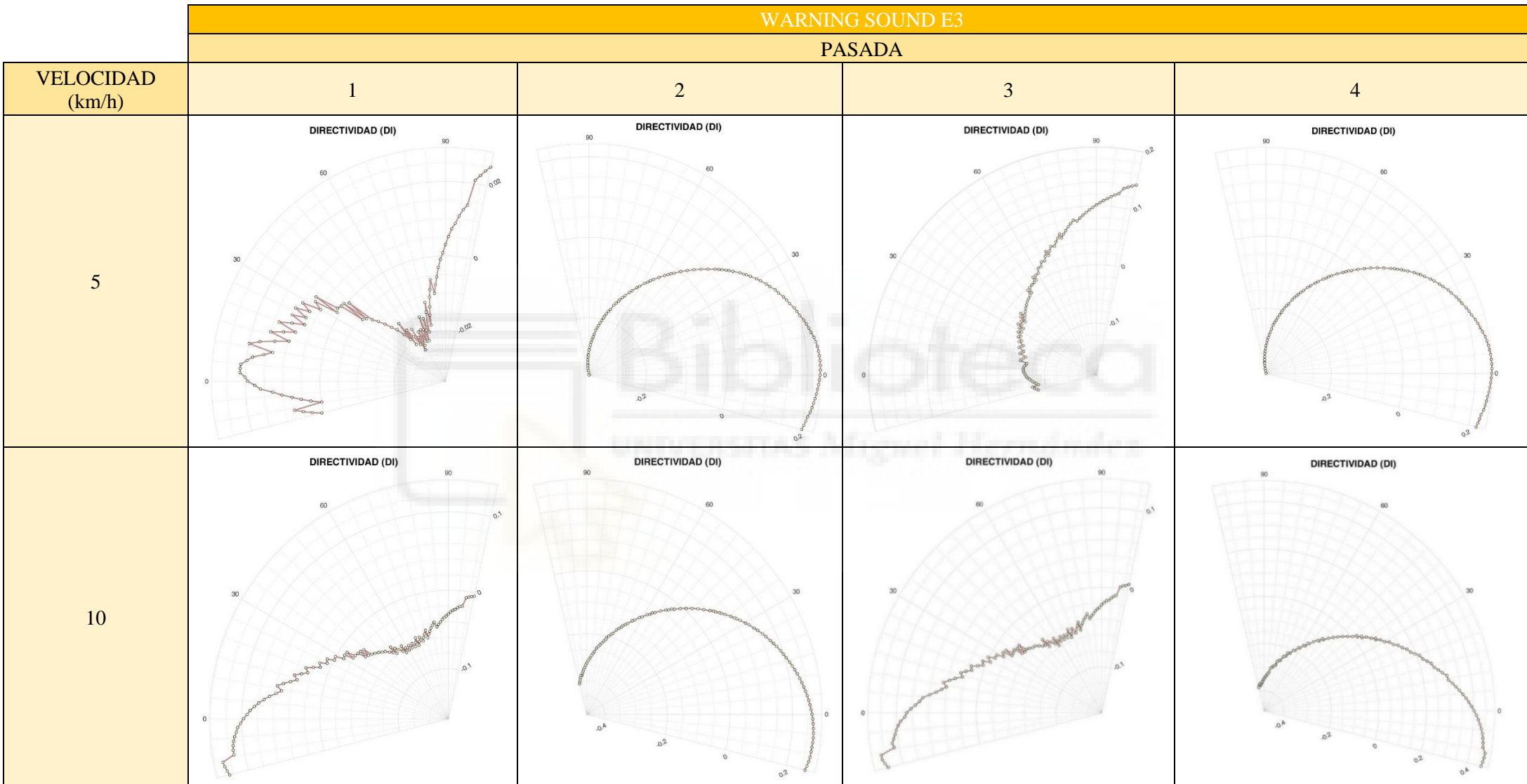


Tabla 4.27. Directividad vertical para velocidades de 5-10 km/h para E3.

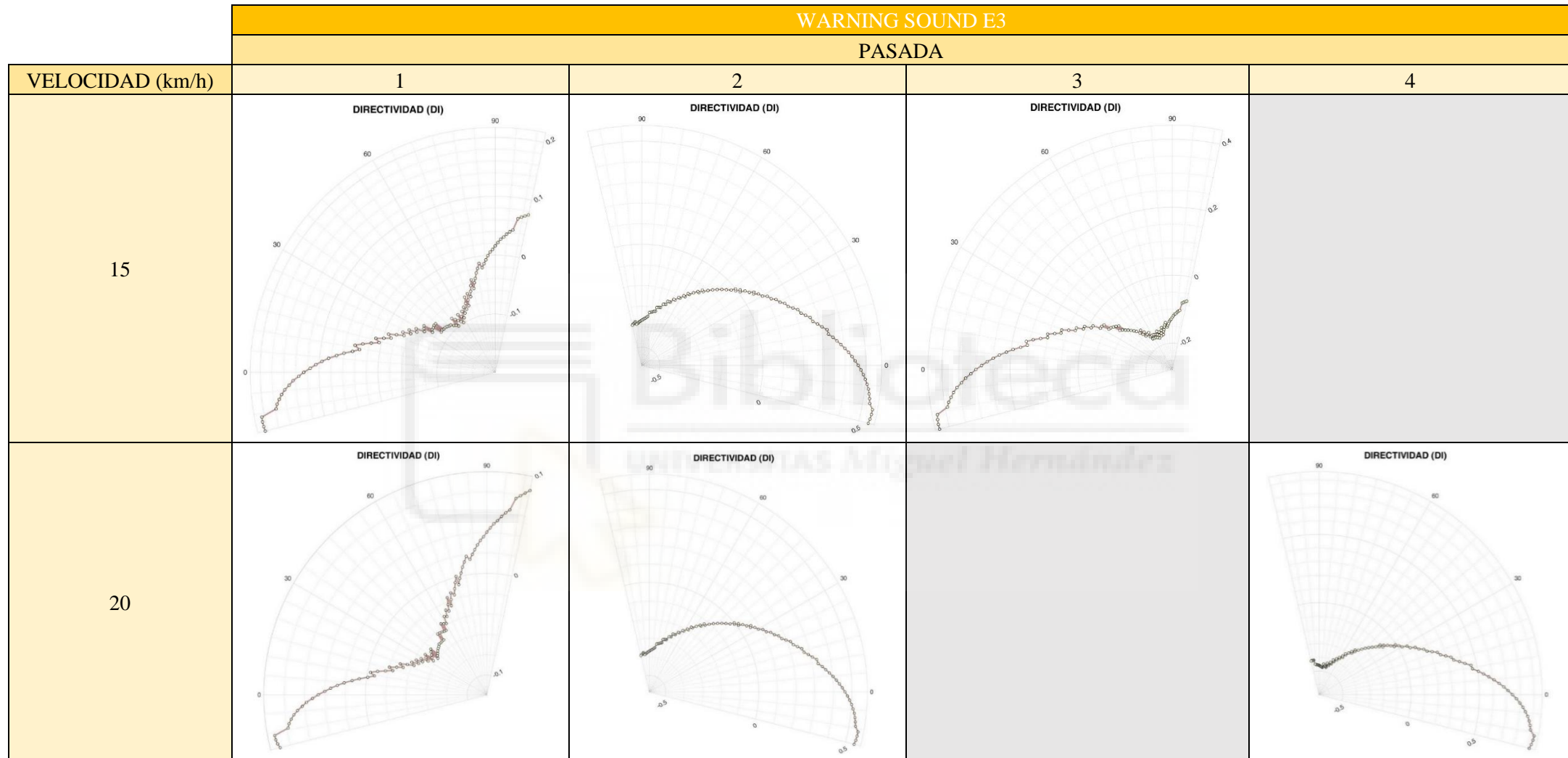


Tabla 4.28. Directividad vertical para velocidades de 15-20 km/h para E3.



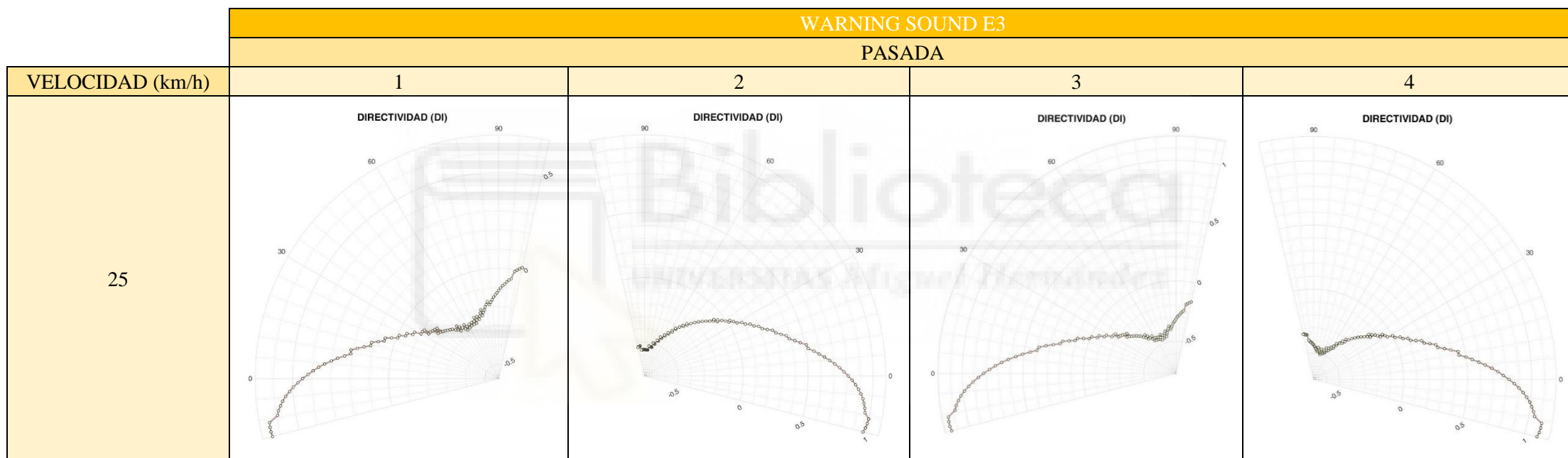


Tabla 4.29. Directividad vertical para velocidad de 25 km/h para E3.

WARNING SOUND E4				
PASADA				
VELOCIDAD (km/h)	1	2	3	4
5				
10				

Tabla 4.30. Directividad vertical para velocidades de 5-10 km/h para E4.

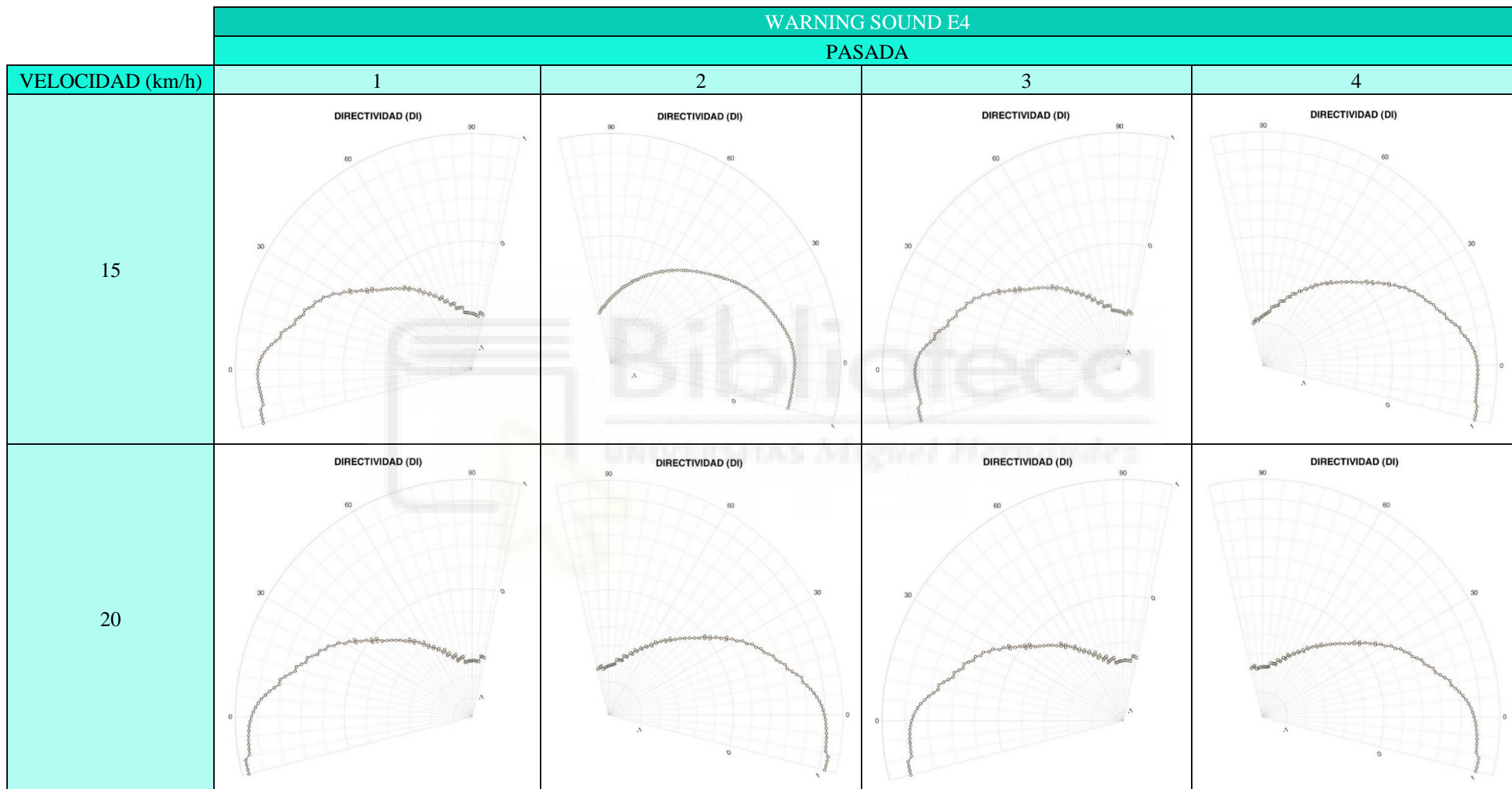


Tabla 4.31. Directividad vertical para velocidades de 15-20 km/h para E4.

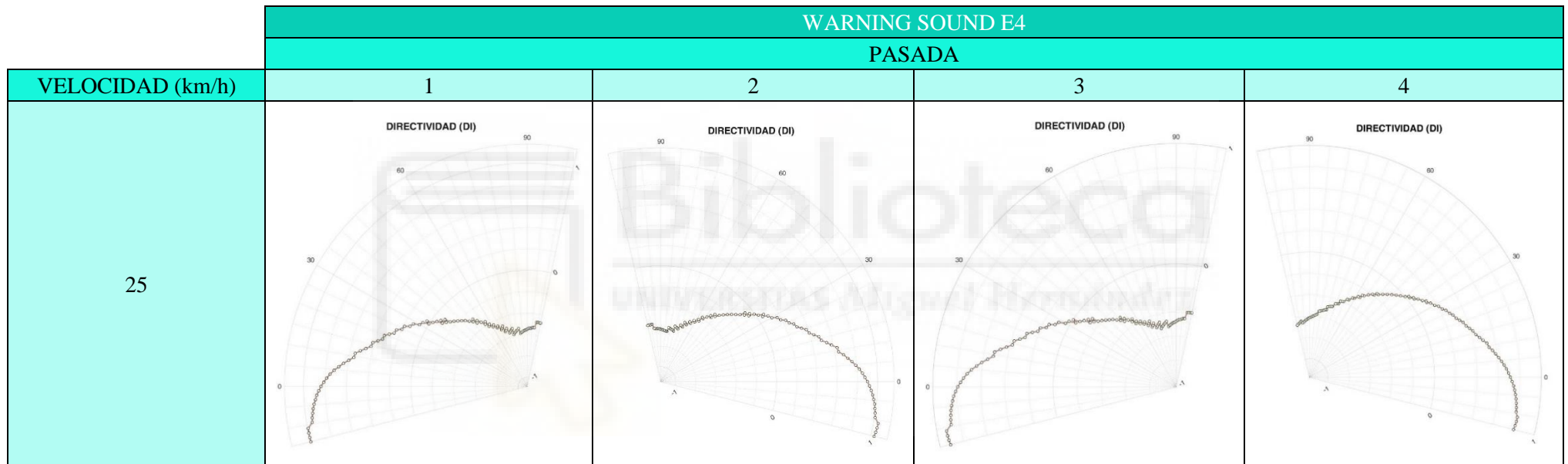


Tabla 4.32. Directividad vertical para velocidad de 25 km/h para E4.

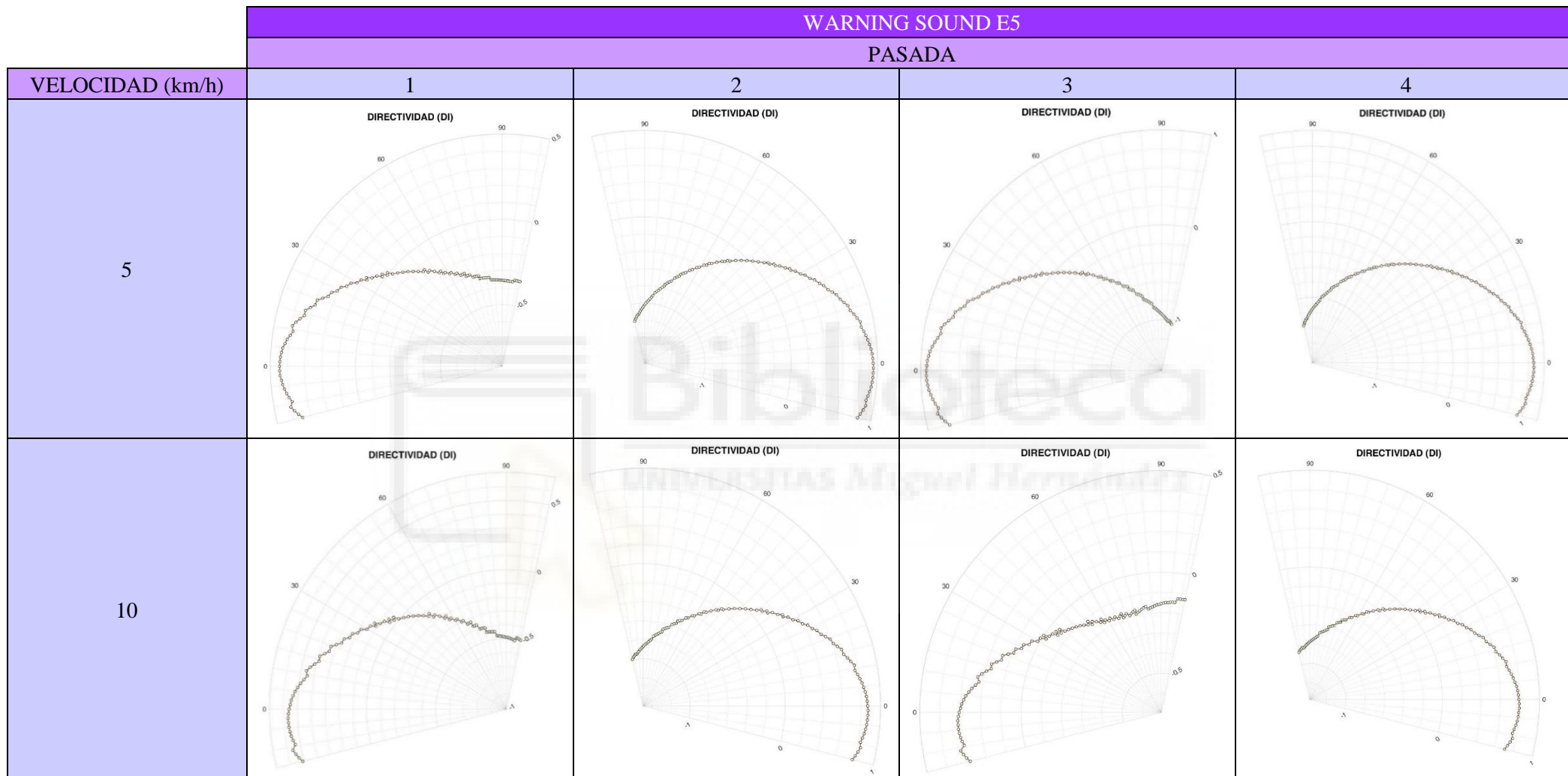


Tabla 4.33. Directividad vertical para velocidades de 5-10 km/h para E5.

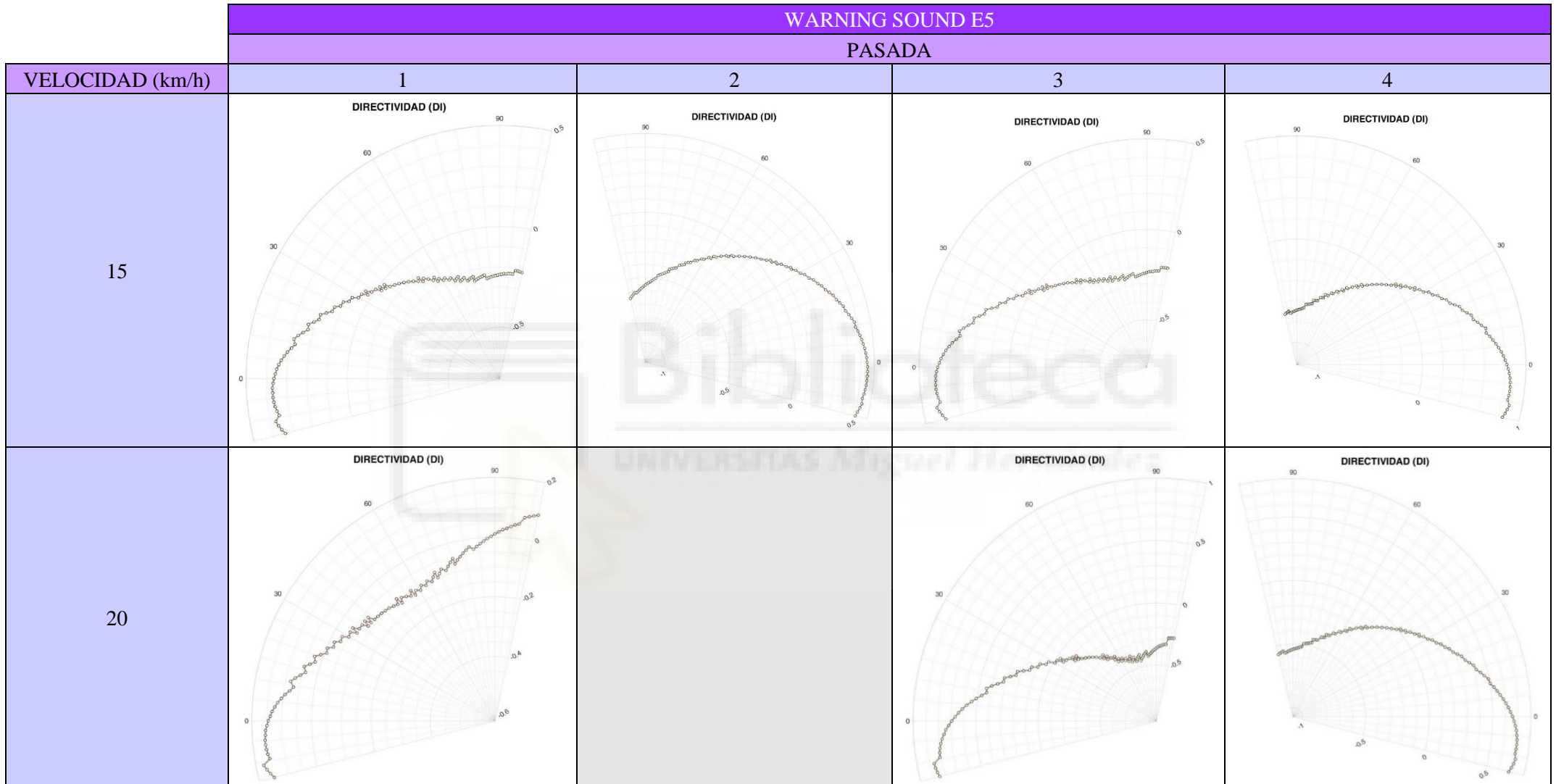


Tabla 4.34. Directividad vertical para velocidades de 15-20 km/h para E5.

WARNING SOUND E5				
PASADA				
VELOCIDAD (km/h)	1	2	3	4
25		Biblioteca UNIVERSITAS Miguel Hernández	Biblioteca UNIVERSITAS Miguel Hernández	

Tabla 4.35. Directividad vertical para velocidad de 25 km/h para E5.

WARNING SOUND E6				
PASADA				
VELOCIDAD (km/h)	1	2	3	4
10				
15				

Tabla 4.36. Directividad vertical para velocidades de 10-15 km/h para E6.



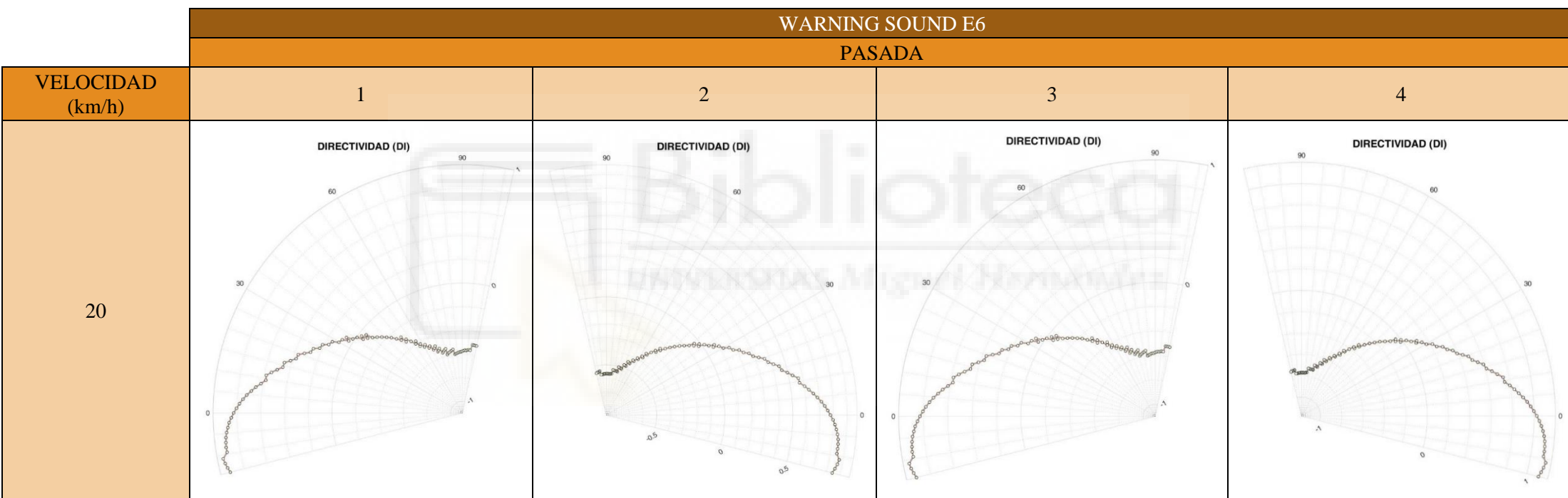


Tabla 4.37. Directividad vertical para velocidad de 20 km/h para E6.

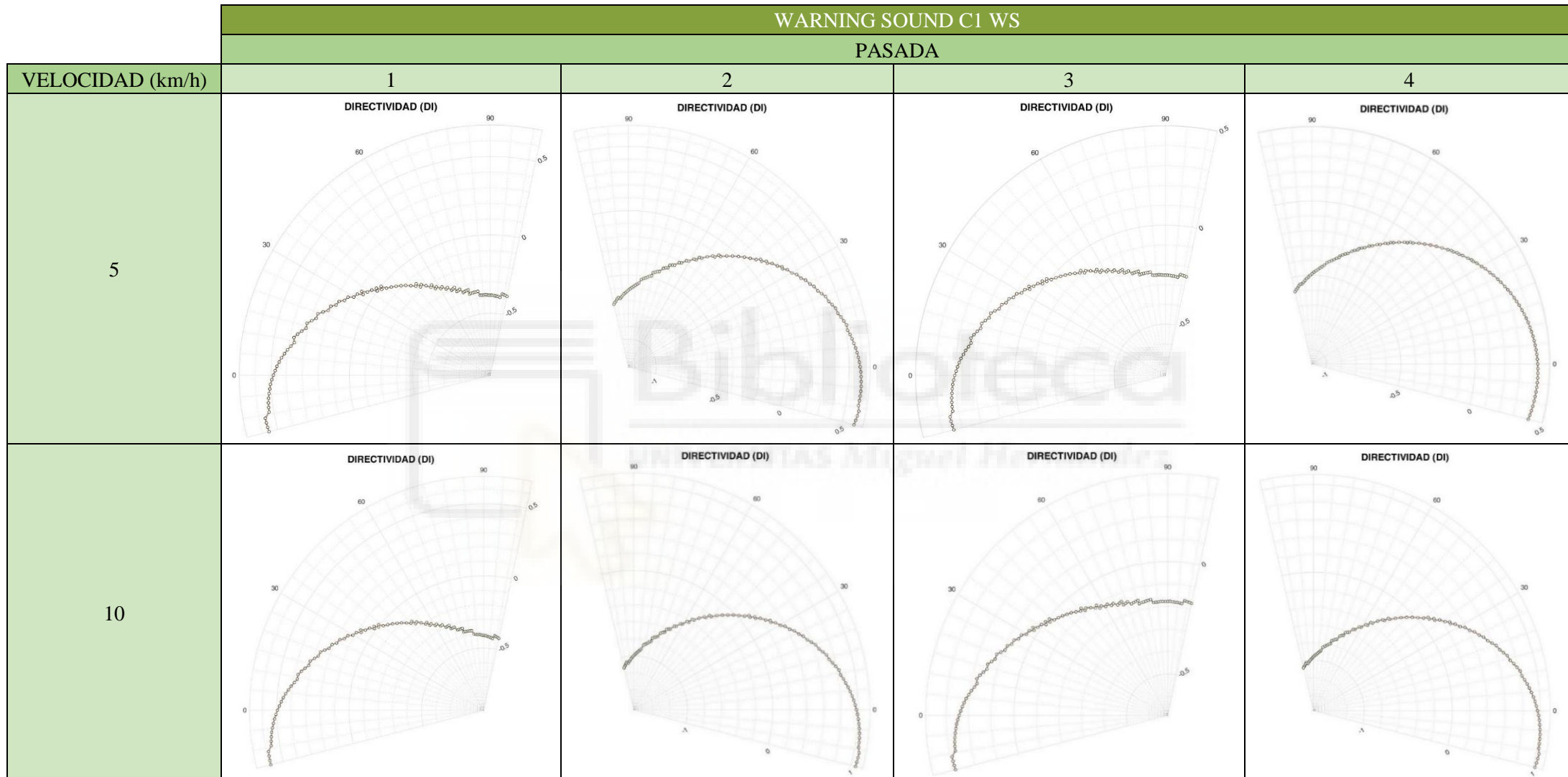


Tabla 4.38. Directividad vertical para velocidades de 5-10 km/h para C1 WS.

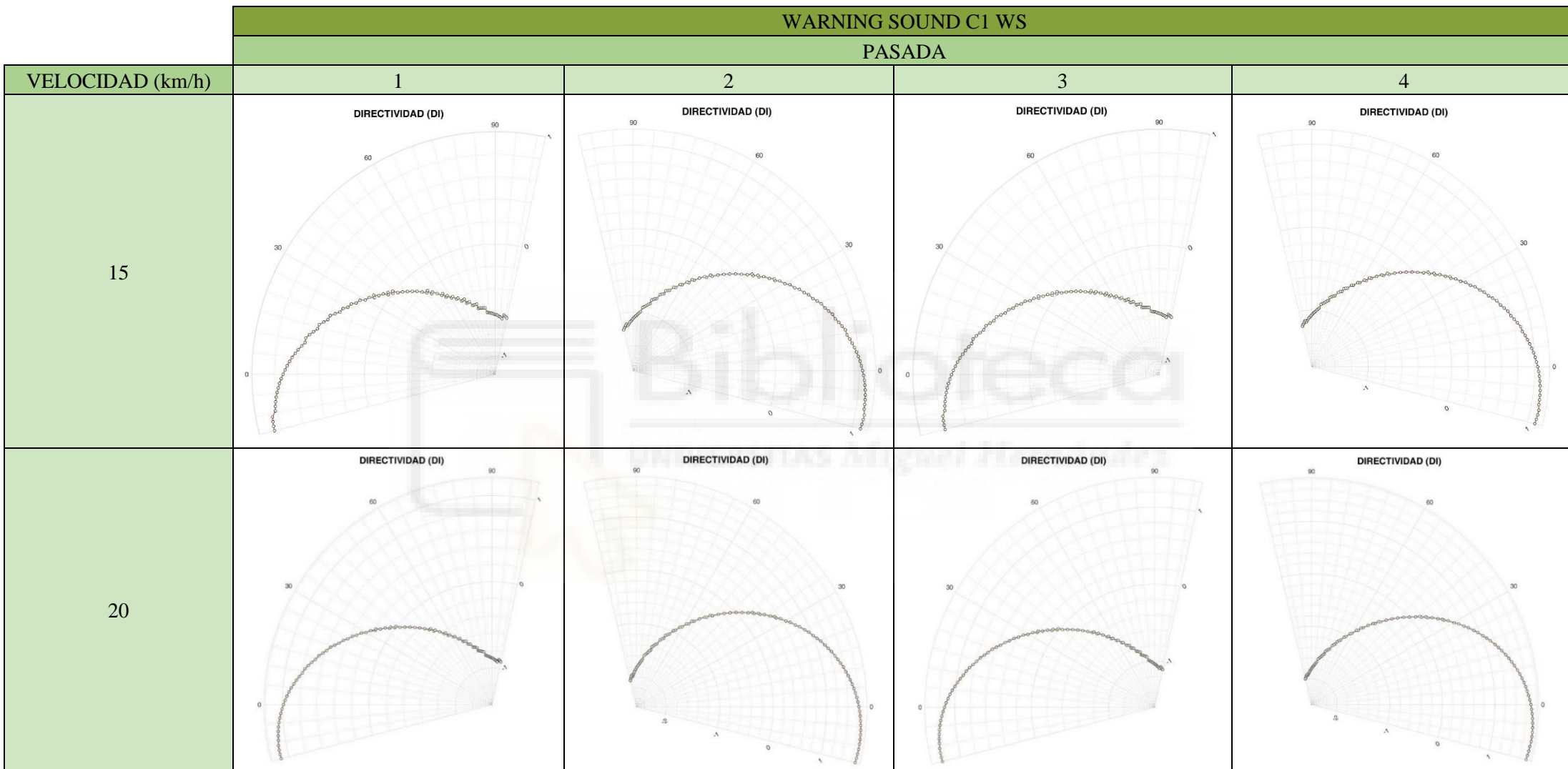


Tabla 4.39. Directividad vertical para velocidades de 15-20 km/h para C1 WS.

WARNING SOUND C1 WS				
PASADA				
VELOCIDAD (km/h)	1	2	3	4
25				

Tabla 4.40. Directividad vertical para velocidad de 25 km/h para C1 WS.

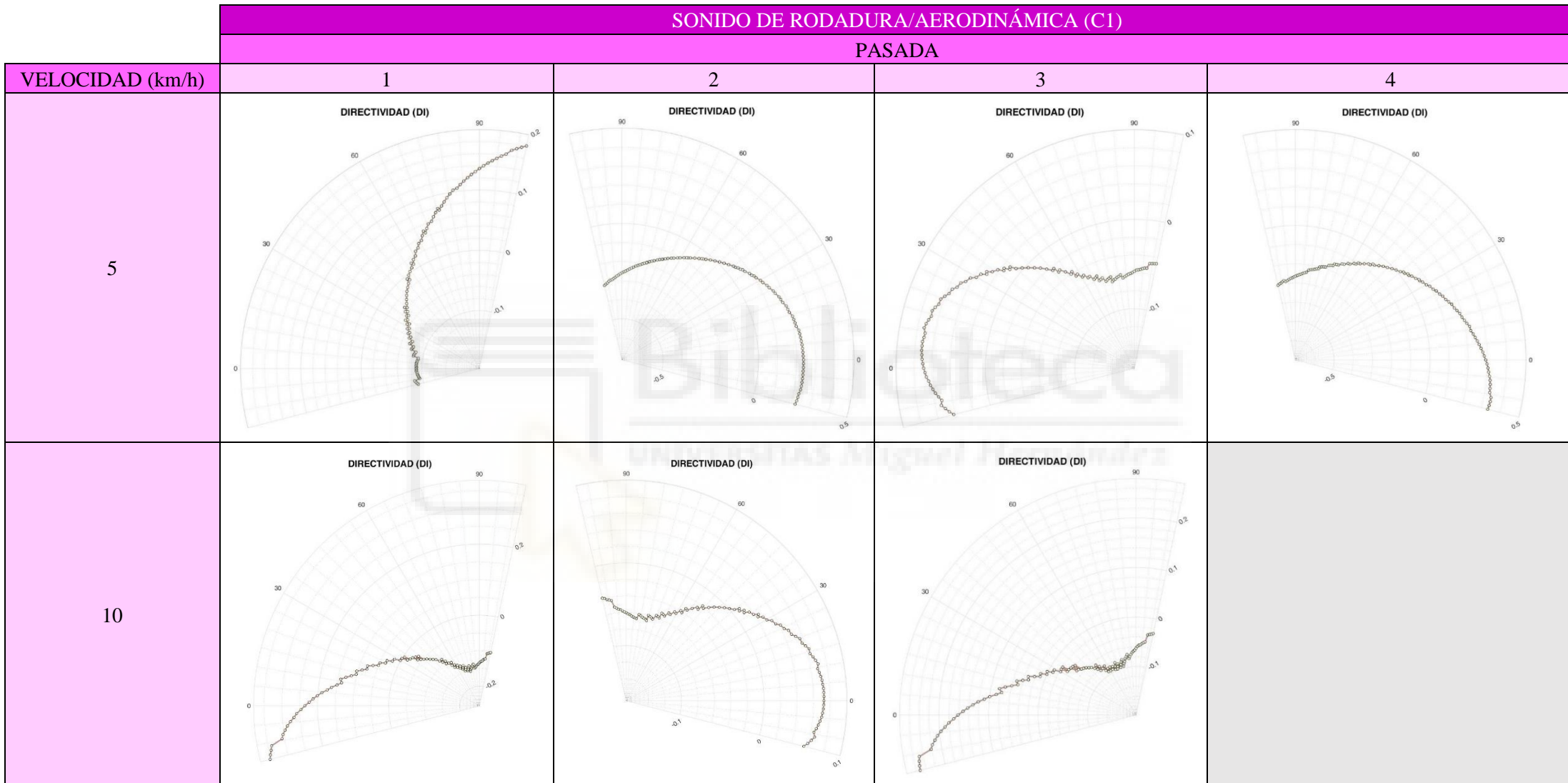


Tabla 4.41. Directividad vertical para velocidades de 5-10 km/h para C1.

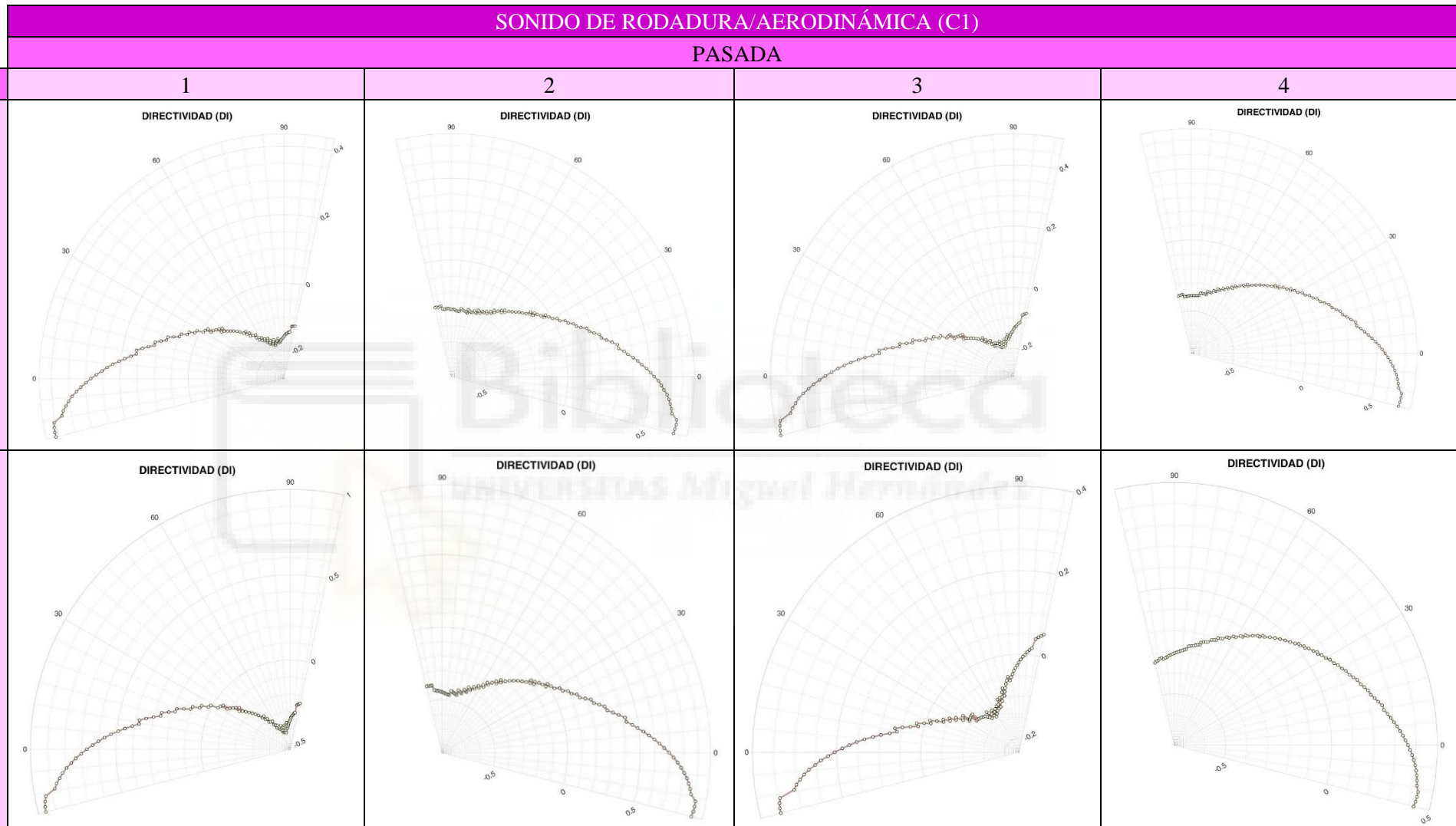


Tabla 4.42. Directividad vertical para velocidades de 15-20 km/h para C1.

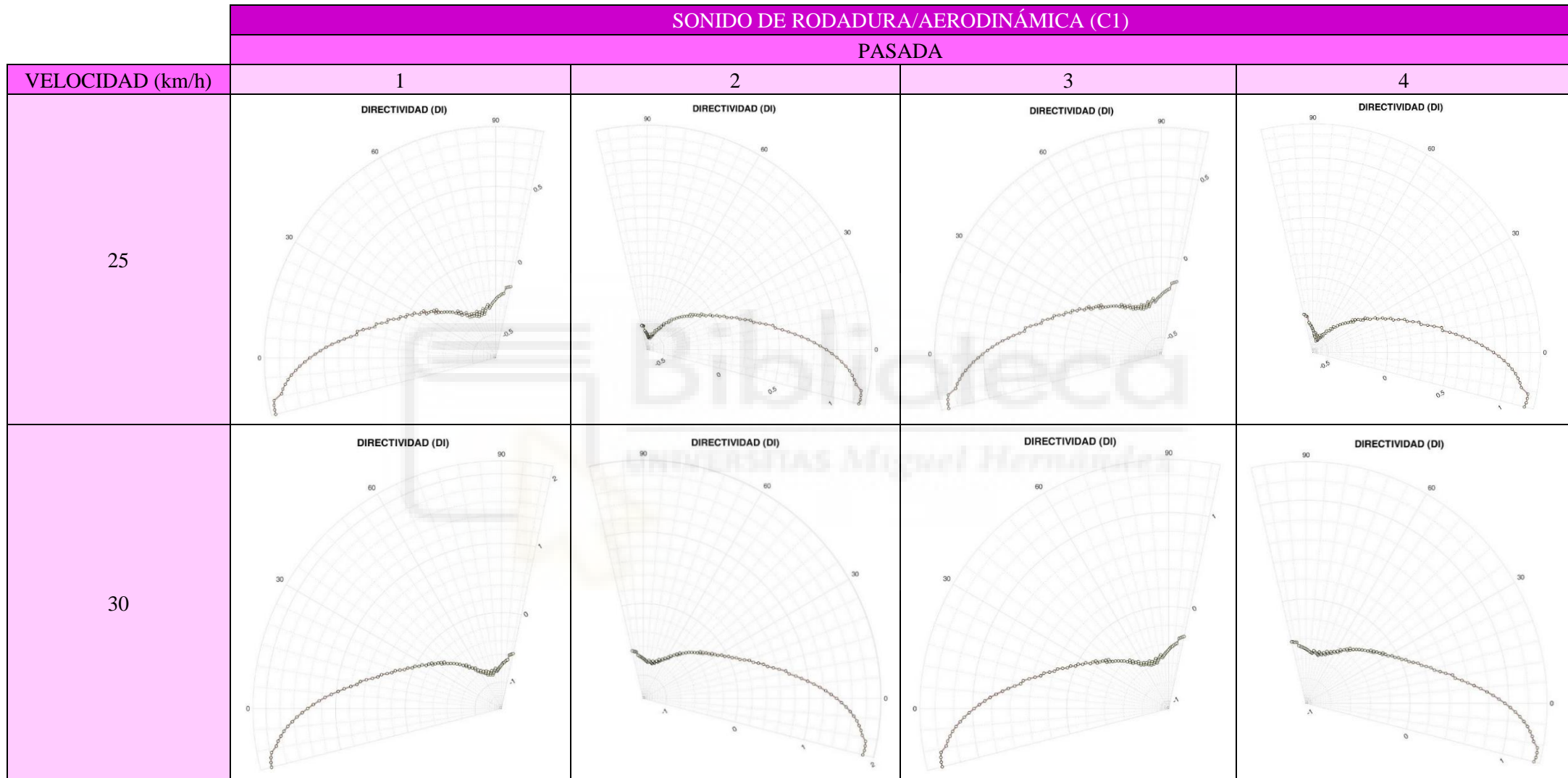


Tabla 4.43. Directividad vertical para velocidades de 25-30 km/h para C1.

## **4.5.2. DISCUSIÓN**

A través de este apartado, se extraerá información acerca de los resultados dispuestos a lo largo del apartado 4.5.1.

### **4.5.2.1. NIVEL SONORO CONTINUO EQUIVALENTE MÁXIMO**

En este apartado, se analizarán los resultados del apartado 4.5.1.1 que se correspondían con el cálculo del nivel sonoro equivalente máximo y el espectro de frecuencia del intervalo de presiones sonoras cuyo extremo superior coincidía con este valor.

#### **4.5.2.1.1. GRÁFICA GLOBAL**

En este punto se desarrollará una explicación acerca de la gráfica donde se reflejan las tendencias de cada uno de los sonidos de advertencia que se estudiaron en los ensayos. Para la obtención de las tendencias, se tomó en cuenta el nivel sonoro equivalente máximo de cada una de las mediciones que se hicieron para cada señal de advertencia, asociándolas con la velocidad a la que se realizaron. Dicho índice de presión sonora se presentó con una cifra decimal.

El punto 3.5 del Anexo 3 del Reglamento nº138 del CEPE [28], indica una serie de disposiciones referidas a la forma en la que deben declararse los resultados. Este establece que, para cada velocidad, se debe escoger el valor más bajo de las dos medias de ambos lados. Tras esto, se toma la media con menor valor y se redondea al entero más cercano.

Esto quiere decir que, con la información de todas las mediciones de un sonido de advertencia, se hace una división por velocidades. Una vez completada esta tarea, se toma cada una de las velocidades por separado y se clasifican las pasadas según su sentido de marcha. Seguidamente, se toman los resultados de cada sentido de marcha y se hace la media con ellos. Finalmente se comparan ambas medias, se toma la menor de ellas y se redondea su valor al entero próximo.

En los resultados expuestos en el apartado 4.5.1.1.1, queda claramente reflejado que no se siguió la normativa. Esto se debe a que, en muchas ocasiones, los resultados obtenidos para distintas señales de advertencia, a través de la forma expuesta en la normativa, se superponían pudiendo provocar confusión. Además, el empleo de mi propia forma de representación, permite obtener unas ecuaciones para las tendencias de las señales de advertencia más ajustadas. Estas ecuaciones se presentarán posteriormente.

Lo primero que hay que destacar, es que los requerimientos mínimos de presión sonora que deben cumplir los sonidos de advertencia en función de la velocidad, quedan representados en el



gráfico a través de unos marcadores amarillos con línea discontinua en color púrpura. Como bien queda reflejado, solamente una toma de datos no cumple con los requerimientos mínimos. Esta medición se corresponde con la velocidad 10 km/h del sonido emitido por el vehículo sin emplear ningún sonido de advertencia (C1). Sin embargo, esto es lógico, puesto que no se está analizando un sonido de advertencia en sí. En cuanto a los requerimientos máximos, ninguna pasada excede el límite permitido.

Antes de comenzar con el análisis de las líneas de tendencia y de los efectos que generan los cambios de velocidad en los sonidos de advertencia, se va a proceder a determinar la dispersión de los datos que se produce en cada velocidad. Para definir esto, lo que se va a calcular es el rango, que consiste en calcular la diferencia entre el valor máximo y mínimo de cada división de celeridad. El siguiente cuadro muestra cuánto vale el rango para cada caso.

	Velocidad (km/h)					
	5	10	15	20	20	30
Rango	19.7	20.8	12.9	11.5	8.7	5.5

Tabla 4.44. Medidas de dispersión de nivel sonoro continuo equivalente para cada velocidad.

Como era de esperar, a mayor velocidad de circulación en el ensayo, el rango es menor y, por lo tanto, los datos se encuentran más agrupados. El único instante donde se produce un ligero aumento en el rango, se produce en el paso de una velocidad de 5 km/h a una velocidad de 10 km/h. Esto es consecuencia del dato del sonido C1 que no cumple con el mínimo establecido en normativa. No obstante, la tendencia general del rango es disminuir.

Seguidamente, se va a llevar a cabo un análisis sobre la tendencia de los sonidos de advertencia estudiados en los ensayos. Pero antes de proceder con el estudio, se incluye a continuación las ecuaciones de las líneas de tendencia de cada señal de advertencia junto con la tendencia del sonido del coche sin emisión de sonido de alerta.

Sonido de advertencia	Ecuación de la línea de tendencia
E1	$L_{A\text{fasteq}\text{m}\acute{\text{a}}\text{x}}(v) = 14.3021\log_{10}(v) + 40.668$
E2	$L_{A\text{fasteq}\text{m}\acute{\text{a}}\text{x}}(v) = 2.2381\log_{10}(v) + 65.197$
E3	$L_{A\text{fasteq}\text{m}\acute{\text{a}}\text{x}}(v) = 18.4576\log_{10}(v) + 34.897$
E4	$L_{A\text{fasteq}\text{m}\acute{\text{a}}\text{x}}(v) = 4.6937\log_{10}(v) + 56.297$
E5	$L_{A\text{fasteq}\text{m}\acute{\text{a}}\text{x}}(v) = 3.9172\log_{10}(v) + 56.854$
E6	$L_{A\text{fasteq}\text{m}\acute{\text{a}}\text{x}}(v) = 5.0754\log_{10}(v) + 54.341$
C1 WS	$L_{A\text{fasteq}\text{m}\acute{\text{a}}\text{x}}(v) = 17.5892\log_{10}(v) + 40.357$
Sonido de rodadura/aerodinámica	Ecuación de la línea de tendencia
C1	$L_{A\text{fasteq}\text{m}\acute{\text{a}}\text{x}}(v) = 20.5213\log_{10}(v) + 32.588$

Tabla 4.45. Ecuaciones para las tendencias de cada sonido de advertencia y el sonido del coche sin emisión de señal de alerta.

Una vez definidas las líneas de tendencia, se procede a desarrollar el análisis. Para ello, se dividen los sonidos de alerta en distintos grupos. La agrupación de las señales de advertencia se realiza por similitud en la forma y ecuación de la línea de tendencia. Por otro lado, el sonido de rodadura/aerodinámica conformará otro grupo. En total, se generan cinco grupos

- El primer grupo incluye el sonido de advertencia E2.
- El segundo grupo engloba el sonido de advertencia C1 WS.
- El tercer grupo está formado por los sonidos de advertencia E4, E5 y E6.
- El cuarto grupo se designa a través de los sonidos de advertencia E1 y E3.
- El quinto grupo queda definido por el sonido de rodadura/aerodinámica C1. Esta agrupación se usará para determinar la mejora en la emisión como consecuencia de utilizar cada sonido de advertencia.

El primer grupo contiene únicamente la señal de advertencia E2. La línea de tendencia de este sonido de advertencia experimenta un crecimiento muy leve de nivel de presión sonora. El nivel de presión sonora pasa de, aproximadamente, 66.5 dB(A) a 5 km/h a un nivel de, más o menos, 68.5 dB(A) a una velocidad de 30 km/h. Como se puede ver, el cambio es poco significativo (2 dB(A)).

El segundo grupo, que está compuesto solamente por el sonido de advertencia C1 WS, funciona justamente de forma contraria. La línea de tendencia generada a través de sus mediciones, experimenta un crecimiento claramente relevante. De este modo, se pasa de valores que se encuentran en un intervalo de 52.5-55.5 dB(A) a un intervalo de 64.5-68 dB(A). En el mejor de los casos el cambio de nivel de presión sonora llega a ser de 15.5 dB(A).

Las tendencias del tercer grupo, se asemejan a la del sonido de advertencia E2. Esto significa que el nivel de presión aumenta ligeramente a medida que aumenta la velocidad. Si se observan los niveles asociados a una velocidad de 5 km/h y se simplifican, el nivel al inicio es de aproximadamente 60 dB(A). Si se realiza lo mismo con los valores asociados a la última velocidad, el nivel aproximado es de 64 dB(A). El rango sería, más o menos, 4 dB(A).

Por último, el cuarto y quinto grupo presenta similitud con la señal de advertencia C1 WS. Sus tendencias son parecidas a las que presenta este sonido. Por lo tanto, la variación de nivel de presión sonora experimentada con estos sonidos de advertencia es considerable. Echando un vistazo a la gráfica se puede considerar que, para la velocidad inicial (haciendo una estimación

bastante abrupta), se obtendría un nivel de presión sonora de 50 dB(A). Tomando la última velocidad y estimando el nivel de presión sonora asociada, se obtiene un valor de 64.5 dB(A). El rango que se obtendría sería de 14.5 dB(A), que es un aumento considerable.

Una vez que se conoce como se comporta cada sonido de advertencia en función de la velocidad, conviene determinar el grado de intervención en la percepción del vehículo.

Si se observa la gráfica, se puede ver que la mayoría de puntos a altas velocidades (25-30 km/h) se encuentran agrupados en una zona. La siguiente figura muestra eso.

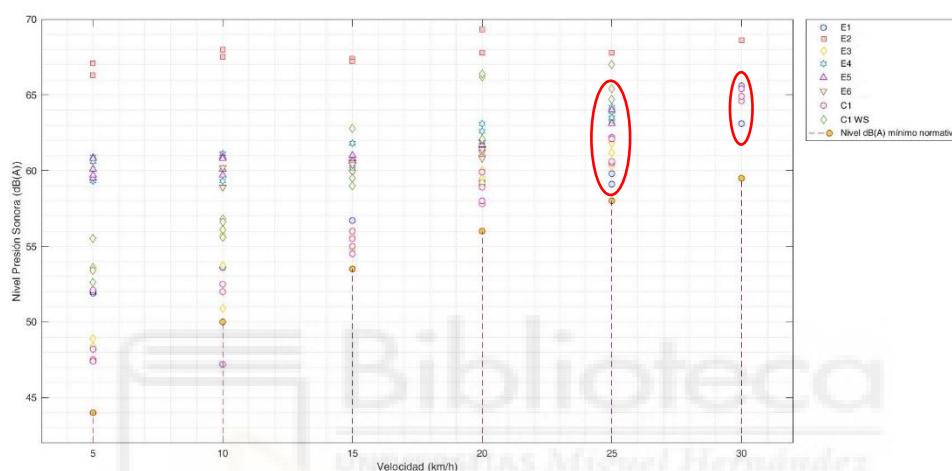


Figura 4.91. Zonas de agrupación de datos en la gráfica donde se representan las tendencias.

Además, esto también queda reflejado en el cálculo del rango puesto que, si se consulta los valores de este para dichas velocidades en la tabla 4.44, se obtienen valores bajos respecto a velocidades más pequeñas. Es decir, existe agrupación de datos. Además, la diferencia de nivel de presión sonora existente entre ambas agrupaciones es pequeña (aproximadamente 2.2 dB(A)).

Estas zonas se corresponden con el momento en el que el sonido que se percibiría, vendría mayormente determinado por el sonido generado a través de la rodadura y de la acción aerodinámica del vehículo.

Por lo tanto, se puede afirmar varias cosas:

1. El sonido de advertencia E2, posee una línea de tendencia que refleja un ligero aumento entre velocidades. Sin embargo, en comparación con el resto de señales, este genera un aumento significativo en el sonido del coche a cualquier velocidad. Esto provoca un desplazamiento de su tendencia hacia la parte superior del gráfico, quedando separada del resto y sobrepasando por mucho el sonido de rodadura/aerodinámica (C1).

2. El sonido de advertencia C1 WS, genera un aumento claro en el nivel de presión sonora desde la velocidad inicial a la final. Este sonido de advertencia se encuentra en todo momento por encima del sonido C1, mejorando la emisión en todos los tramos (incluso a velocidades altas, donde la rodadura y aerodinámica predomina).
3. El conjunto de sonidos perteneciente al tercer grupo posee líneas de tendencia donde el aumento del nivel de presión sonora, entre velocidades, es pequeño. Es decir, al principio de la marcha, el nivel es similar al sonido emitido por el vehículo cuando este proviene, de forma predominante, del sonido de rodadura y aerodinámico. Sin embargo, la diferencia con C1 es evidentemente notable en el rango de velocidades bajas/media, siendo prácticamente nula, esta diferencia, cuando se alcanzan los 25-30 km/h.
4. El cuarto grupo, que se comporta como C1 WS, produce un cambio significativo en el nivel de presión sonora al aumentar la velocidad. No obstante, estas señales de advertencia son las que más cerca se encuentran a C1 (sobre todo el sonido E3). Esto implica que la aportación que hacen estas señales al sonido de rodadura/aerodinámica es pequeña. A bajas/medias velocidades las señales de advertencia de este grupo emiten por encima de C1. Al alcanzar la velocidad de 15 km/h para E3 y 20 km/h para E1 el sonido percibido provendría exclusivamente de la rodadura y aerodinámica.

Con toda la información extraída, se pueden asegurar varias cosas. En primer lugar, aquellos sonidos que presentan líneas de tendencia como los del grupo 1 y 3, presentan cierto problema. La emisión que generan estos sonidos de advertencia es buena a bajas velocidades. Incluso con E2 el nivel energético es tan alto, que enmascara el sonido del conjunto rodadura/aerodinámica. A pesar de ello, estas señales de advertencia pueden generar confusión en los usuarios de la vía, no reconociendo adecuadamente la velocidad real del vehículo. Además, la contaminación acústica que producirían sería considerable, sobre todo, E2.

En segundo lugar, se encuentran los grupos 2 y 4, cuyas tendencias los hace más adecuadas para ser empleados como sonidos de advertencia. Esto se sustenta también en el hecho de que el sonido C1 WS, es la señal de advertencia que los ingenieros de Renault diseñaron para utilizar en su vehículo. Por lo tanto, es un sonido creado por expertos del ámbito, así que su funcionamiento puede considerarse como una referencia. Aunque las líneas de tendencia de los sonidos del grupo 4 se asemejan a la de C1 WS y cumplen con la normativa, el nivel que se consigue a través de estos es menor que el resto, e incluso llegan a quedar enmascarados por la rodadura/aerodinámica en tramos de velocidad medios/altos. Quizá sea conveniente aumentar la intensidad con la que se emiten los mismos.

#### 4.5.2.1.2. ESPECTRO DE FRECUENCIAS

A lo largo de este apartado, se analizarán la información que proporcionan los espectros de frecuencias representados en el apartado 4.5.1.1.2 de este trabajo. Para este estudio, se observará cada sonido de advertencia por separado, discerniendo acerca de varios puntos. Estos puntos consistirán en: cumplimiento de los requerimientos mínimos establecidos en normativa, tendencia general de los espectros de frecuencia en el sonido de advertencia y examen de los efectos, según el espectro de frecuencias, que producen las señales de advertencia.

##### a) Sonido de advertencia E1.

El sonido de advertencia E1 cumple de forma general las disposiciones en cuanto a las bandas en tercios de octava establecidas en la normativa. Es cierto que existen tres casos donde no se cumplen los requisitos, pero estos se encuentran en una posición muy cercana a cumplirla. El primer incumplimiento se debe a que existen dos bandas que cumplen los requerimientos, pero ninguna se encuentra por debajo o es igual que la banda de 1600 Hz. El otro caso es porque sí que existe esa banda por debajo o igual a la de 1600 Hz, pero una segunda banda no cumple, por poco, la normativa. Aunque no todos los casos cumplen con la normativa, se puede considerar que no es un problema grave.

Este sonido de advertencia, comienza presentando la banda de 500 Hz destacando por encima del resto, pudiendo considerarla como una componente tonal. Para 15 km/h, sigue existiendo cierta diferencia de esta banda con el resto, pero se puede considerar que la distribución es uniforme. Cuando se alcanzan los 20 km/h, todos los niveles de presión se igualan, adoptando una distribución de banda ancha. La homogeneidad se mantiene en todas las mediciones de las velocidades posteriores.

El nivel de presión sonora para la banda predominante, es de aproximadamente 50 dB(A) en la primera velocidad de marcha y acaba en un nivel cercano a 59 dB(A) a los 30 km/h. El aumento que experimenta la banda es de alrededor de 9 dB(A). Se puede decir que los niveles de presión sonora son normales.

##### b) Sonido de advertencia E2.

El sonido de advertencia E2 cumple, en todos los espectros de frecuencia obtenidos, los requerimientos del reglamento.

En este caso, se presenta una clara predominancia de la banda de 1000 Hz a lo largo de todas las mediciones que se realizaron. No obstante, cabe destacar que las bandas de 800 Hz y 1250 Hz presentan cierta relevancia a bajas velocidades, desapareciendo cuando el vehículo circula por encima de 20 km/h. Este es un ejemplo de componente tonal que se produce a cualquier velocidad.

El nivel energético para la banda de 1000 Hz, es de aproximadamente 69 dB(A) para cualquier velocidad. El nivel es considerablemente elevado.

c) Sonido de advertencia E3.

De los 16 espectros de frecuencia que se obtuvieron con esta señal de advertencia, 6 casos no cumplen con la normativa. Destaca la velocidad de 10 km/h, donde dos de los casos no cumplen, aunque por poco. Sin embargo, lo más llamativo se produce a la velocidad de 15 km/h. Cuando se conduce a esta velocidad, ningún espectro de frecuencia cumple con las exigencias mínimas. Esto indica que en el intervalo de velocidades medio-bajo el sonido de advertencia presenta problemas.

Este sonido de advertencia, presenta una predominancia de la banda de 500 Hz, junto con otras como la de 2000 Hz o la de 5000 Hz. Esto se produce de forma notable a 5 km/h. Claramente, lo que se presenta a esa velocidad, son componentes tonales.

Cuando se alcanzan los 10 km/h, los niveles de presión sonora comienzan a homogeneizarse, aunque las bandas anteriormente enumeradas destacan en cierta medida. Para velocidades superiores a esta, el espectro de frecuencia adquiere de forma evidente un comportamiento de banda ancha.

La banda predominante, presenta un nivel de presión sonora, cercano a 45 dB(A) en la primera velocidad, pasando a un valor de aproximadamente 56 dB(A). El aumento de energía sonora, es alrededor de 9 dB(A). De nuevo, el nivel de presión sonora presente es normal.

d) Sonido de advertencia E4.

Este sonido de advertencia cumple, en todo momento, las exigencias mínimas referidas a las bandas de tercios de octava.

A excepción de los casos que se realizaron por encima de 15 km/h, este sonido de advertencia contiene 5 bandas que destacan por encima del resto. Esto quiere decir, que existe

tonalidad en los espectros de frecuencia de este sonido de advertencia a velocidades bajas/medias. Estas bandas son las de 200 Hz, 420 Hz, 500 Hz, 2000 Hz y 2500 Hz.

Tras alcanzar los 20 km/h, aunque es cierto que las bandas anteriores siguen teniendo cierta notoriedad, el comportamiento que predomina, en el resto de velocidades, es el de banda ancha donde la distribución de las bandas se caracteriza por ser homogéneo.

Los niveles de presión sonora que presentan las bandas con mayor energía se encuentran en un intervalo de 50-60 dB(A) en la primera velocidad de marcha. Este intervalo asciende ligeramente a 52-60 dB(A) para la última velocidad. Los niveles que se presentan son usuales.

e) Sonido de advertencia E5.

La señal de advertencia E5, presenta una serie de rarezas. Primeramente, para una velocidad de 5 km/h, la normativa se cumple para tres de las cuatro mediciones (aunque por poco cumplen las cuatro tomas de datos). Para 10 km/h, ocurre exactamente lo mismo que en 5 km/h. Pero al llegar a 15 km/h, ninguna medición de las realizadas cumple con los requerimientos mínimos. Finalmente, para 20 km/h y 25 km/h, las exigencias se vuelven a cumplir.

Este sonido de advertencia presenta un componente tonal, cuya diferencia con el resto de bandas disminuye hasta que se alcanza una velocidad de marcha de 25 km/h, donde se produce una cierta homogeneización en el nivel de presión sonora. La banda que predomina a lo largo de las mediciones es la de 500 Hz, aunque a 5 km/h destacan también otras.

La banda de 500 Hz presenta un nivel de presión sonora, en todas las mediciones, de cerca de 60 dB(A). El nivel energético es ligeramente elevado.

f) Sonido de advertencia E6.

En E6, no se presenta ningún inconveniente en los espectros de frecuencia obtenidos. La normativa se cumple perfectamente.

La predominancia de las bandas de 315 Hz, 400 Hz y 500 Hz, para la velocidad de 10 km/h, denota un cierto carácter tonal. No obstante, a 15 km/h estas bandas son superiores al resto, pero podría comenzar a considerarse que la distribución es homogénea. Para la última velocidad de circulación del vehículo, se puede asegurar que el comportamiento es de banda ancha.

Al inicio de la marcha, la banda de 500 Hz presenta un nivel de presión sonora que oscila entre 54 dB(A) y los 56 dB(A). Cuando el vehículo alcanza la última velocidad a la que se ensayó

ese sonido de advertencia, los niveles de la banda no cambiaron. El nivel de presión sonora presente a lo largo del ensayo es usual.

g) Sonido de advertencia C1 WS.

Todas las pasadas de C1 WS cumplen las disposiciones exigidas en normativa.

El sonido de alerta C1 WS, presenta componentes tonales en las bandas de 315 Hz y 400 Hz, en la mayoría de las mediciones realizadas a 5 km/h (aunque en la misma velocidad hay otras bandas que también destacan). Tras esto, a pesar de que hay ciertos casos donde hay alguna banda que destaca por encima del resto (sobre todo la de 400 Hz), la distribución predominante es la de banda ancha.

En la primera velocidad, las bandas predominantes presentan un nivel energético cercano a 49 dB(A), llegando a los 62 dB(A) en la última velocidad. Aunque los niveles de presión sonora al principio son normales, en las velocidades superiores el nivel energético es elevado. Además, el aumento del nivel sonoro en las bandas predominantes es considerable.

h) Sonido del vehículo sin emisión de sonido de advertencia (C1).

El sonido C1, es incapaz de cumplir con la normativa para velocidades bajas/medias. No es hasta que el vehículo circula a 20 km/h, cuando aparecen mediciones donde se cumplen todos los requisitos de la normativa (solamente dos mediciones de cuatro). Tras esto, para 25 km/h y 30 km/h, las exigencias se cumplen sobradamente. Esto no es extraño, puesto que este sonido no presenta emisión de ningún tipo de sonido de advertencia. El hecho de que cumpla la normativa a altas velocidades, se debe a que, en estas, el efecto sonoro de la rodadura y la aerodinámica del vehículo es mucho mayor que a bajas.

Todos los espectros de frecuencias de esta señal de advertencia, presentan en todo momento una composición en forma de banda ancha. Algo que tampoco es raro, puesto que el sonido de rodadura/aerodinámica tiene dicha forma. Algo excepcional que se produce a lo largo de las mediciones, es que se produce en varios casos una concentración de energía sonora en bandas de frecuencia altas, como en la medición número 4 de la velocidad 5km/h. Esto es algo que no había ocurrido hasta el momento ya que, en los sonidos de alerta, la energía suele concentrarse en la zona intermedia del espectro de frecuencias.

Los niveles de presión sonora de las bandas con mayor energía se encuentran, a 5 km/h, en todos los casos por debajo de 45 dB(A). Sin embargo, cuando el coche alcanza la última velocidad estudiada, el límite superior de dichas bandas es de 60 dB(A). El aumento del nivel de presión



sonora que se producen en las bandas de los espectros de frecuencia de este sonido de advertencia es relevante.

Se debe realizar una aclaración sobre la forma en la que se distribuye el nivel energético en los espectros de frecuencias a medida que la velocidad aumenta. Como se puede observar, en muchas ocasiones, las señales de advertencia presentan, en mayor o menor medida, cierto nivel de tonalidad a velocidades bajas/medias. Sin embargo, cuando se alcanzan velocidades elevadas (25-30 km/h) se produce, en la mayoría de los casos, una transformación en la distribución de la energía, cambiando hacia un comportamiento de banda ancha. Esto es consecuencia de la predominancia del sonido de rodadura/aerodinámica en la emisión.

Por otro lado, cabe destacar el comportamiento tonal que presentan la mayoría de los vehículos a bajas velocidades. La presencia de componentes tonales en los espectros de frecuencia implica un aumento de la detectabilidad por parte de los usuarios de la vía. No obstante, esto también lleva asociado un aumento en la posibilidad de que el sonido sea percibido como molesto. Un caso especial es la señal de alerta E2, que presenta tonalidad a lo largo de todas las velocidades.

Si a esto se añade el hecho de que existen niveles de presión sonora elevados (como el caso de E2), el efecto anterior se intensifica. Esto puede suponer, más que una solución, un problema serio de contaminación acústica y molestia que derivará en problemas de salud. No obstante, hay casos como C1 WS (sonido de advertencia diseñado por Renault), donde primeramente se tiene tonalidad a baja velocidad y después se tiene un comportamiento de banda ancha. Además, en este caso se produce un aumento significativo en el nivel energético de las bandas a medida que aumenta la velocidad produciendo, probablemente, una mayor detectabilidad.

Finalmente, cabe recalcar que los niveles de presión sonora de las bandas predominantes, siguen dos patrones a la hora de incrementar la velocidad. Estos dos patrones son: aumento del nivel de presión sonora o estancamiento del nivel sonoro. Se puede observar que estos patrones están relacionados con la forma que tienen las tendencias de cada sonido de advertencia. Esto quiere decir que, en el caso de que la línea de tendencia no sufra un aumento significativo (presente prácticamente una tendencia lineal), el nivel de la banda no aumentará al incrementar la velocidad. En contraposición, si la tendencia sufre un aumento considerable, el nivel energético ascenderá, con mayor fuerza cuanto mayor incremento realice la tendencia de la señal de alerta analizada.

#### 4.5.2.2. DIRECTIVIDAD VERTICAL

En este apartado, se llevará a cabo un estudio de los resultados de directividad vertical del apartado 4.5.1.2. Primeramente, se analizará la forma de emisión de cada uno de los sonidos de advertencia. Al mismo tiempo, se buscarán tendencias entre todas las mediciones de un mismo sonido de advertencia. Finalmente se intentará encontrar una correlación entre todas las pasadas a una misma velocidad independientemente del sonido de advertencia que se emitiera.

Como se ha expuesto en el anterior párrafo, en primer lugar, se procede a determinar la directividad en cada sonido de advertencia. También se buscarán correlaciones que se produzcan entre las distintas tomas de datos de cada uno de ellos. Esto se hará realizando un estudio yendo uno por uno.

##### a) Sonido de advertencia E1.

Esta señal de advertencia presenta una emisión en forma de arco, aunque existen algunas excepciones. Estas excepciones pueden ser de dos tipos. El primer tipo de excepción presenta una geometría de emisión que presenta un mínimo intermedio, generando dos arcos de menor tamaño. El segundo tipo de excepción presenta una emisión en forma de arco, pero que no cumple la forma de emisión general.

Para los casos que no son una excepción, la emisión se distribuye de la siguiente forma. Para ángulos superiores de  $90^\circ$  hasta el rango de ángulos  $55^\circ$ - $35^\circ$ , según el caso, la emisión es inferior a la que realizaría una fuente omnidireccional. Tras este rango, el índice de directividad pasa a ser positivo.

Las velocidades 5 km/h y 10 km/h de las mediciones con código número 1, presentan una directividad que sigue la segunda excepción definida anteriormente. Como ya se ha dicho, esta sigue una forma de arco, pero presenta una distribución distinta a la forma general. Aquí, los ángulos que se encuentran entre el límite superior del sector polar y los ángulos  $40^\circ$  o  $60^\circ$ , respectivamente, presentan un índice de directividad superior a 0. Finalmente, los ángulos que se presentan entre los ángulos anteriores y el límite inferior del sector polar, adquieren valores inferiores a 0.

Por otro lado, el primer tipo de excepción sigue otra distribución. Las mediciones número 1 con velocidad inferiores a 20 km/h, presentan, en primer lugar, valores de índice de directividad positivos desde ángulos mayores de  $90^\circ$  hasta aproximadamente los  $85^\circ$ , momento en el que el índice de directividad alcanza el 0 y continúa descendiendo hasta que se alcanza un ángulo de entre  $45^\circ$  y  $50^\circ$ , momento en el que comienza la ascensión alcanzando nuevamente valores de

índice de directividad positivos. Para las mediciones con el mismo valor asignado realizadas a velocidades de 25 km/h y 30 km/h, las emisiones comienzan prácticamente con valores muy cercanos a 0 llegando a alcanzar el mínimo índice de directividad a los 60°, aumentando este, seguidamente, hasta alcanzar su máximo aproximadamente en -15° aproximadamente.

Las divisiones que se utilizan en el sector polar, son iguales a la unidad. No obstante, la tercera velocidad de la medición 1, presenta valores de índice de directividad mayores a -0.1 y menores a 0.1. Esto indica que la diferencia respecto a una emisión omnidireccional no es especialmente relevante.

Lo que se produce en esta señal de advertencia, a medida que aumenta la velocidad, son varias cosas:

- En las mediciones que tienen asignadas el número 1, se genera un mínimo que produce una disminución significativa en la emisión. Además, para las velocidades de 25 km/h y 30 km/h, lo que se produce es una reducción de gran envergadura de los valores del índice de directividad del arco que queda a la derecha del pico.
- Las mediciones definidas con número 2, se produce una reducción progresiva del ángulo donde el índice de directividad se hace 0, pasando de casi 55° a 35°. A su vez, para una velocidad de 30 km/h, en el límite izquierdo del arco, se genera una ligera variación del arco.
- El rango de valores del índice de directividad para todos los casos aumenta. Esto implica que, las partes positivas emiten cada vez más que la fuente omnidireccional. Por otro lado, las zonas con valores negativos emitirán progresivamente menos.

#### b) Sonido de advertencia E2.

El segundo sonido de advertencia establece una emisión con una forma aproximada de dipolo en todo momento.

El índice de directividad presenta valores negativos desde ángulos con valor superior a 90° hasta un ángulo oscilante entre 10° y 15°. A lo largo de todo este recorrido, el índice realiza varios cambios. Primeramente, este aumenta acercándose al 0, llegando incluso, en la medición 1 velocidad 10 km/h, a sobrepasarlo cuando se tiene un ángulo de aproximadamente 45°. A continuación, entre 45° y 20° el índice de directividad disminuye, generándose el primer arco. Seguidamente, el índice de directividad vuelve a aumentar, consiguiendo sobrepasar y alcanzar el máximo valor de emisión casi al alcanzar el límite inferior del sector polar.

Los límites del sector polar donde se representa índice de directividad, superan la unidad. Esto significa que los índices de directividad representados son elevados. Por esto, se puede afirmar que la diferencia en la emisión que produce este sonido de advertencia respecto a una emisión omnidireccional es muy relevante.

El aumento de la velocidad para esta señal de advertencia produce una serie de efectos:

- El ángulo donde se produce el máximo del arco de menor tamaño, va disminuyendo de  $60^\circ$  a  $45^\circ$  aproximadamente.
- El rango de valores del índice de directividad disminuye. La traducción de esto es que tanto los valores negativos como los positivos se encuentran cada vez más cerca de la emisión omnidireccional.

### c) Sonido de advertencia E3.

Este sonido de advertencia genera una directividad muy similar a la que se genera en el E1. La forma más extendida dentro de este sonido es el arco o el arco con variación en el extremo, que aparecen en velocidades altas. A su vez, se presentan otros casos donde la distribución genera un mínimo que forma dos arcos. Además, también existen ciertos casos donde, aunque la forma de la emisión es un arco, no se sigue la distribución de los casos normales.

Para la forma de emisión predominante, la emisión comienza siendo menor que la fuente omnidireccional desde el límite superior del sector polar hasta un ángulo cuyos valores oscilan entre  $50^\circ$  y  $30^\circ$ , según el caso. Tras el ángulo anterior, el índice adopta valores positivos hasta que se alcanza el máximo en el límite inferior del sector polar o en sus proximidades. Si el arco presenta la variación en el extremo, antes de comenzar la ascensión al máximo, se produce una disminución del índice alcanzando un mínimo en un ángulo que oscila entre  $70^\circ$  y  $60^\circ$ .

Para los casos que presentan una geometría de doble arco con mínimo intermedio, estos comienzan presentando una emisión que se encuentra por encima de la fuente omnidireccional. El índice de directividad desciende, llegando al mínimo, que se produce entre los  $60^\circ$  y los  $40^\circ$  según la medición y tiene valor negativo. Tras esto, la emisión aumenta sobrepasando el 0 y alcanzando el máximo alrededor del límite inferior. A pesar de que esto es cierto para la mayoría de los casos, en la medición 1 de la velocidad 5 km/h, el índice de directividad no aumenta de forma progresiva, si no que realiza picos muy abruptos donde la emisión varía constantemente, alcanzando el máximo a los  $10^\circ$ .

El último caso que no se corresponde a la geometría predominante, es un arco donde la distribución de la directividad no se corresponde con la explicada anteriormente. Esto se produce en la medición 3 de la velocidad 5 km/h. Aquí, se produce una disminución de los valores del índice de directividad. El sector comienza presentando valores de emisión por encima de la fuente omnidireccional desde su límite superior hasta los  $55^\circ$ , momento donde se alcanza el 0. El valor del índice de directividad continúa descendiendo hasta el límite inferior. El máximo nivel de emisión se alcanza en el límite superior.

Las divisiones límite que presenta el sector polar, son en todo momento iguales o menores a la unidad. Esto implica que la directividad de este sonido de advertencia no presenta una diferencia con respecto a la fuente omnidireccional muy exagerada.

Los efectos producidos en la directividad de esta señal de advertencia como consecuencia de un aumento de velocidad, son:

- El rango de valores de índice de directividad aumenta, produciendo un alejamiento de los valores de emisión reales respecto la emisión omnidireccional.
- Descenso del ángulo donde se produce el cambio de signo del índice de directividad. La traducción de esto es un achatamiento del arco cercano al límite inferior del sector polar.
- Aparición de casos con mínimo intermedio, produciendo casos de doble arco. Esto a su vez lleva asociada un aumento del valor del índice de directividad en ángulos elevados (desde  $75^\circ$  a ángulos mayores de  $90^\circ$ ). También se producen casos de geometría de arco con ligera variación en el extremo.

#### d) Sonido de advertencia E4.

La señal de advertencia en cuarta posición, presenta una directividad con distribución en forma de arco irregular en la mayoría de los casos.

La distribución de los valores del índice de directividad, comienza con la presencia de datos negativos, que se mantiene desde el ángulo máximo analizado hasta, generalmente, los  $40^\circ$ , aunque puede reducirse hasta los  $30^\circ$ . Tras alcanzar este ángulo, se produce un cambio en la emisión, que pasa a ser superior que la fuente omnidireccional.

Los límites del sector polar donde se representa el índice de directividad oscilan entre valores de  $-1$  y  $1$ , en la mayoría de los casos (existen casos donde el índice de directividad sobrepasa dichos límites). Esto implica que la diferencia respecto a una emisión omnidireccional es considerable.

El aumento de la velocidad para este sonido de advertencia produce una serie de efectos:

- La forma del arco irregular va cambiando hasta generar un arco con apariencia más circunferencial. Además, se genera una ligera variación en la forma del arco en su extremo para las velocidades de 20 km/h y 25 km/h.
- El rango de valores de índice de directividad disminuye. El significado de esto es que, tanto los valores negativos y positivos, se encuentran cada vez más cerca de la emisión omnidireccional.
- El paso de valores de índice de directividad de positivos a negativos se produce, cada vez en un ángulo menor. Al principio este ángulo es de  $40^\circ$  y al final es de aproximadamente  $35^\circ$ . Esto desemboca en un achatamiento del arco.

e) Sonido de advertencia E5.

El quinto sonido de advertencia emite en forma de arco, tanto con una forma más o menos circunferencial como con variación en el extremo. Sin embargo, a medida que la velocidad aumenta, se generan varios casos particulares.

Los casos normales se identifican por presentar la siguiente distribución. El valor del índice de directividad comienza encontrándose por debajo de 0 desde el límite superior del sector polar hasta un ángulo que oscila entre  $45^\circ$  y  $30^\circ$ . Posteriormente, el índice de directividad continúa ascendiendo, pero con valores positivos, hasta alcanzar el máximo en una zona cercana o igual al límite del sector polar. Hay que recalcar que las mediciones que cuentan con una geometría de arco con ligera variación en el extremo, siguen la misma tendencia explicada anteriormente, con la diferencia de que existe una disminución del valor de la emisión antes de comenzar la ascensión al 0.

Los casos excepcionales ocurren en la primera medición de las velocidades de 20 km/h y 25 km/h. El primero tiene forma de línea con arcos en los extremos, mientras que el segundo, tiene forma de doble arco con mínimo intermedio. Ambas geometrías siguen una distribución parecida. El índice de directividad comienza con valores positivos cercanos al 0. Este desciende, primeramente, alcanzando un mínimo de emisión a  $60^\circ$ . Tras este mínimo, se produce el incremento de los valores del índice, hasta alcanzar el máximo casi en el límite inferior del sector polar.

Los valores de índice de directividad que se presentan en esta señal de advertencia, se encuentran en todo momento por debajo de 1 y por encima de -1. Incluso hay casos donde los

valores se encuentran por debajo de 0.5 y por encima de -0.5. Esto implica que la diferencia respecto a una emisión omnidireccional no es extraordinaria.

El incremento de la velocidad para esta señal de advertencia produce una serie de efectos:

- Aparición de ligera variación en el extremo del arco e incluso desarrollo de mínimo intermedio que provoca división en dos arcos.
- Disminución del ángulo donde se produce el cambio de signo del índice de directividad. Esto implica el achatamiento del arco.

f) Sonido de advertencia E6.

Esta señal de advertencia genera una directividad en forma de arco con ligera variación en uno de los extremos, aunque dos casos se presentan como arcos perfectos.

En primer lugar, la emisión presenta valores negativos, cuando el ángulo analizado se encuentra entre  $35^\circ$  y  $40^\circ$  según el caso. Tras posicionarse el valor del índice de directividad en 0, cuando se alcanza el ángulo anterior, este índice comenzará a aumentar hasta el límite inferior del sector polar.

Las divisiones límites que se usan en el sector polar son iguales a la unidad o menores a esta. Por ello, se puede afirmar que la diferencia de la emisión real con respecto a una omnidireccional es ligeramente importante.

Al aumentar la velocidad mientras se emplea el sonido de advertencia E6, se produce un solo efecto:

- El rango de valores de índice de directividad aumenta, produciendo un alejamiento de los valores de emisión reales de la emisión omnidireccional.

g) Sonido de advertencia C1 WS.

El último sonido de advertencia emite en forma de arco.

El índice de directividad comienza presentando valores negativos para ángulos que se encuentren por debajo del límite superior del sector polar hasta un rango de  $55^\circ$  y  $40^\circ$ , en función de la medición. El siguiente tramo va desde el ángulo anterior hasta el límite inferior del sector polar, que consta de valores de índice de directividad positivos. Aproximadamente en el extremo inferior se produce el máximo de emisión.

Las divisiones del sector polar, son inicialmente menores a la unidad. Posteriormente, las divisiones se igualan a la unidad. En última instancia, estas superan a la unidad. Esto significa, que a medida que la velocidad aumenta, la directividad con respecto a una emisión omnidireccional va variando de no muy importante a ciertamente importante.

Incrementar la velocidad produce, en la directividad de esta señal de advertencia el siguiente efecto:

- El rango de valores de índice de directividad aumenta. La traducción de esto es que tanto los valores negativos y positivos se alejan más de la emisión omnidireccional con cada aumento.

#### h) Sonido del vehículo sin emisión de sonido de advertencia (C1).

C1, realiza una emisión cuya forma generalmente se corresponde a un arco con una ligera variación en el extremo. Sin embargo, hay casos que no se corresponden exactamente a esta geometría, siendo por ejemplo un arco o doble arco con mínimo intermedio.

En primer lugar, se definirá la distribución de los valores del índice de directividad para los casos que siguen la geometría predominante. El índice, comienza con valores menores a 0, para un rango de ángulos que va desde  $45^\circ$  a  $20^\circ$ , según la medición. Antes de llegar a 0, primeramente, se produce un descenso del índice hasta llegar a un mínimo. Tras esto, comienza el incremento del valor del índice. El aumento hace que este alcance el 0 y continúe creciendo hasta alcanzar el máximo aproximadamente en el límite del sector polar.

La manera en la que se produce la emisión en las mediciones 2 y 4 de la velocidad 5 km/h y la cuarta medición de la velocidad 20 km/h tiene forma de arco normal. Su directividad es muy similar a la expuesta en el párrafo anterior. La única modificación que se debe tener en cuenta es que, al no existir la variación en el extremo, el índice de directividad solamente aumenta.

Para la medición 1 de la velocidad 5 km/h, la directividad también tiene forma de arco. Sin embargo, los índices de directividad positivos se encuentran desde el límite superior del sector polar, hasta el ángulo de  $50^\circ$ . Tras este ángulo y hasta el límite inferior, el valor del índice es menor a 0. El valor máximo de emisión se alcanza en el límite superior.

Por último, se debe analizar la toma de datos 3 de la velocidad 25 km/h. Esta presenta una emisión con forma excepcional, donde se generan dos arcos con un mínimo intermedio que los separa. Aquí la emisión comienza siendo positiva, pero muy cercana a 0. El índice de directividad desciende, disminuyendo su valor por debajo del 0, hasta que llega a  $60^\circ$ , donde se alcanza el



mínimo. Justamente después, comienza a aumentar recuperando la positividad de sus valores y alcanzando el valor máximo de emisión en el límite inferior.

El sector polar presenta divisiones cuyos valores poseen una gran variabilidad. Los sectores polares a bajas velocidades presentan divisiones que son menores a la unidad. A medida que la velocidad aumenta, las divisiones incrementan su valor, llegando incluso a sobrepasar la unidad. Esto implica que, aunque al principio la diferencia con respecto a una emisión omnidireccional es ínfima, cuando se alcanzan velocidades mayores, la diferencia comienza a cobrar relevancia. Esto se debe al efecto de la rodadura y a la aerodinámica del vehículo que genera una emisión considerable a altas velocidades, cosa que a bajas no ocurre.

El incremento de la velocidad usando este sonido de advertencia, genera los siguientes efectos:

- El rango de valores de índice de directividad aumenta, produciendo un alejamiento de los valores de emisión reales de la emisión omnidireccional.
- Achatamiento del arco, producido por el continuo descenso del ángulo donde se produce el cambio de signo del índice de directividad.

Cabe destacar, antes de realizar una síntesis de las geometrías de emisión que se producen en cada uno de los sonidos de advertencia, que las geometrías denominadas como doble arco con mínimo intermedio y arco con ligera variación en el extremo, son similares. Esto quiere decir, que la primera geometría es una exageración de la segunda. Por lo tanto, la presencia de ambas en un mismo sonido de advertencia no es sinónimo de error.

Una vez definidas cada una las formas de emisión de cada uno de los sonidos de advertencia, y de las tendencias que se producen al aumentar la velocidad de circulación del vehículo empleándolos, se realiza una síntesis de geometrías.

- Sonido de advertencia E1. La manera en la que este sonido emite, oscila entre diversas formas, produciéndose una mayor diferencia a bajas/medias velocidades. Para obtener una mayor homogeneización de resultados, habría convenido tomar más datos. Las geometrías generadas a través de la representación del índice de directividad difieren entre arco y doble arco con mínimo intermedio, llegando a altas velocidades a ser de arco con variación en el extremo.
- Sonido de advertencia E2. Todos los casos estudiados presentaban una forma de emisión similar a un dipolo.

- Sonido de advertencia E3. Caso similar a E1, donde se tiene alta variabilidad en cuanto a la geometría de directividad. No obstante, el hecho de haber realizado una mayor cantidad de pasadas, permite asegurar que la forma de emisión va desde arco o arco con ligera variación en el extremo (bajas velocidades), a arco con ligera variación en el extremo o doble arco con mínimo intermedio (medias/altas velocidades).
- Sonido de advertencia E4. La geometría de emisión a bajas velocidades se presenta como un arco irregular, pasando a homogeneizarse y a variar en el extremo a medida que la velocidad aumenta.
- Sonido de advertencia E5. La quinta señal de advertencia estudiada presenta, mayormente, una geometría de arco o arco con ligera variación en el extremo. No obstante, también hay algún caso aislado de doble arco con mínimo intermedio.
- Sonido de advertencia E6. Este caso presenta, de forma general, una geometría de arco con variación en el extremo.
- Sonido de advertencia C1 WS. Aquí, la emisión es en todo momento en forma de arco.
- Sonido C1. La directividad, en este caso, tiene forma de arco con ligera variación en el extremo en la mayoría de las mediciones.

Hasta ahora, toda la información detallada ha estado referida a cada sonido de advertencia de forma aislada, sin buscar tendencias que relacionen los sonidos de advertencia entre sí. A pesar de que, debido a la disparidad de geometrías y de tendencias que se producen entre sonidos de advertencia, e incluso dentro de los mismos, se consiguió establecer una tendencia. Esta tendencia se refiere al hecho de que, en la mayoría de los sonidos de advertencia, se produce un achatamiento de la geometría de emisión. Esto es un efecto favorable, ya que el sonido de advertencia conseguiría alertar a los usuarios de la vía, reduciendo las emisiones hacia zonas elevadas (plantas de edificios).

Por último, comentar que, sonidos de advertencia como E2 donde los valores de directividad son elevados, son muy interesantes debido a la alta capacidad que tendrían de alertar a los usuarios de la vía. No obstante, este sonido de advertencia tiene dos puntos flacos. El primero es la disminución progresiva de los valores de índice de directividad a medida que avanza la velocidad. Si analizamos el comportamiento de la señal de advertencia que los ingenieros de Renault diseñaron para su coche (C1 WS), se puede ver que la emisión se aleja de una directividad omnidireccional a medida que el coche aumenta de velocidad. Esto se debe a que cuanto más rápido circula el coche, se debe alertar con mayor énfasis al usuario para evitar accidentes.

La segunda flaqueza, es el arco menor que presentan sus geometrías de emisión que pueden producir contaminación acústica en zonas elevadas como edificios. A su vez, el elevado valor de la emisión con respecto a una fuente omnidireccional, intensificaría el efecto de dicha contaminación.

## 5. CONCLUSIONES

Las dificultades que presenta el vehículo eléctrico actual para ser percibido por los usuarios de la vía, han hecho que se propongan diversas medidas para abordar un problema que, año tras año, se cobra nuevas víctimas.

Una de las propuestas que mayor éxito ha presentado es la introducción de sonidos dentro del coche, para ser emitidos por medio de los sistemas SAAV. De esta manera, diversas investigaciones y proyectos, llevados a cabo tanto por empresas particulares especializadas en el sector como investigadores/docentes, han intentado buscar respuestas a diversas preguntas que se plantean. Preguntas como ¿qué sonido es el mejor?, ¿qué manera de emisión es la más acertada? o ¿cómo disponer los altavoces en el vehículo para optimizar la percepción?

Aunque cada investigación ha llegado a diversas conclusiones, ninguna de ellas ha conseguido proporcionar una verdad absoluta. Por ello, este trabajo surgió con el propósito de proporcionar un granito de arena para encontrar una solución óptima del problema. Aunque, quizá esto sea algo utópico que nunca pueda llegar a suceder como consecuencia de la subjetividad con la que el ser humano percibe a través de sus sentidos.

A través del primer se definieron las señales de alerta a instalar en el vehículo eléctrico. Estas señales provenían de distintas fuentes y presentaban claras diferencias entre ellas. De esta forma se conseguía a través de un estudio, analizar sonidos muy variados.

En el segundo apartado de este trabajo, se realizaron los ensayos pass-by. En estos se ejecutaron las mediciones de las señales de audio que emitió el vehículo. Las señales dependían del tipo de pasada realizada, la velocidad de la misma y el tipo de señal de alerta empleada. A su vez, la cámara acústica almacenó datos correspondientes al mapa sonoro alrededor del vehículo.

Tras esto, se desarrollaron una serie de programas que permitiesen manipular las señales recogidas a través de los ensayos. Con estos se logró determinar, entre otros parámetros, el nivel sonoro equivalente máximo de cada una de las señales almacenadas y el espectro de frecuencias resultante al tomar un intervalo de presión sonora donde el límite superior es el nivel sonoro equivalente máximo.

Seguidamente, se tomaron los datos relacionados con el mapa sonoro alrededor del vehículo, y se realizaron programas que permitiesen establecer resultados acerca de la directividad vertical producida en el plano de circulación del vehículo.

Una vez obtenida la información anterior, se procedió a analizarla. En primer lugar, a calcularon las tendencias que seguía cada uno de los sonidos de advertencia al aumentar la velocidad. A su vez, estas tendencias se representaron en una gráfica, comparando así todos los casos estudiados. Se llegó a la conclusión de que tendencias con ligero aumento en la tendencia e incluso desplazadas hacia zonas superiores del gráfico, podían presentar problemas de contaminación acústica y de confusión de los usuarios de la vía. Por otro lado, los sonidos de advertencia que presentaban una tendencia de aumento progresivo, podían solventar de forma efectiva el problema de la detectabilidad del vehículo, aunque se recomendó el aumento de la intensidad de emisión de los sonidos de advertencia del grupo cuatro para equipararse a la señal de alerta C1 WS.

Seguidamente, se analizaron los espectros de frecuencias de cada una de las mediciones. Se observó la presencia de componentes tonales, a bajas/medias velocidades, en prácticamente todas las emisiones con sonidos de advertencia. Sin embargo, cuando las velocidades alcanzaban rangos altos, los espectros de frecuencia adoptaban de forma completa un comportamiento de banda ancha (aunque a velocidades medias este comportamiento ya se hacía, más o menos, evidente).

A pesar de que lo normal es lo anterior, hay casos como E2, que presentan tonalidad a cualquier velocidad. Esto implica una mejor detectabilidad del vehículo, aunque también puede aparecer cierta molestia en los usuarios. Esto, sumado al hecho de que hay casos como E2, donde el nivel energético de la banda predominante es elevado, supone aún más un aumento de la posibilidad de detección y de molestia.

Finalmente, se estudiaron los resultados del índice de directividad de cada una de las mediciones. Se observó que había gran disparidad en las formas de emisión y las tendencias que se producían al aumentar la velocidad del vehículo emitiendo cada sonido de advertencia. No obstante, se concluyó que una de las posibles tendencias generales (sufrida por todas las señales de alerta), era el achatamiento del arco que se genera cercano al límite inferior del sector polar. Esto es positivo, puesto que cuando el sonido del vehículo aumente, la contaminación acústica recibida en edificios disminuirá. A su vez, se determinó que casos como E2, donde la diferencia de resultados del índice de directividad, comparándola con una fuente omnidireccional, era elevada mejorarían la detectabilidad del vehículo. No obstante, tanto la elevada diferencia como su geometría de emisión, generarían una mayor contaminación acústica en los alrededores.

## 5.1. TRABAJOS FUTUROS

En este trabajo se han desarrollado un conjunto de programas en Matlab que permitiesen extraer, transformar y obtener información a partir de las señales obtenidas en los ensayos pass-by. Los parámetros calculados fueron el nivel sonoro equivalente máximo y el índice de directividad.

Las mediciones que se realizaron a lo largo de la ejecución de los ensayos, se basaron en Pasadas Largas y Cortas. A pesar de que los programas que desarrollé obtuvieron resultados para ambos tipos de pasadas, a la hora de discutirlos, solamente se analizaron las Pasadas Cortas. Por ello, una futura tarea podría ser el análisis de estos datos.

Por otro lado, los programas creados permiten generar otras salidas de información, tales como la señal de presión sonora a lo largo del tiempo o la señal de nivel sonoro equivalente en función del tiempo. Con estos datos, junto con los obtenidos en este trabajo, lo que se puede llevar a cabo es un análisis de otros parámetros como, por ejemplo, la modulación que pueda presentar las señales. Con esta información, se podría definir que sonidos de advertencia presentarían un mayor grado de detectabilidad, cuales se percibirían como agradables/molestos o cuales generarían mayor contaminación acústica.

Finalmente, en los archivos con extensión .txt y en los archivos .dat, se presenta información acerca de los espectros de frecuencias que se recopilaban a lo largo de los ensayos. Otra tarea a realizar, podría ser el estudio de la distribución de la energía sonora en función de la frecuencia, obteniendo así un mayor conocimiento del comportamiento de las señales de alerta. A su vez, se podría hacer algo similar que en la tarea anterior: determinar detectabilidad, molestia y el grado de contaminación acústica.

## 6. REFERENCIAS

### 6.1. BIBLIOGRAFÍA

[1] Mariano-Merino J, Muñoz-Repiso L. La percepción acústica: Física de la audición. Dpto. de Didáctica de las Ciencias Experimentales de la Universidad de Valladolid, España. Revista de Ciencias, junio 2013

[2] Pérez-Vega C. Comprensión de audio y radio digital. Dpto. de Ingeniería de Comunicaciones de la Universidad de Cantabria, España. Presentación gráfica, octubre 2011.

[3] Martín-Domingo A. Apuntes de Acústica. Dpto. de Estructuras y Física de Edificación de la Universidad Politécnica de Madrid, España. Libro versión 2.1, marzo 2014.

[4] Norma UNE-ISO 226:2013. Ruido y sus efectos sobre las personas. Normalización Española, 18 de septiembre de 2013.

[5] Ministerio para la Transición Ecológica y el Reto Demográfico. Conceptos básicos del ruido ambiental. Dpto. de Calidad y Evaluación ambiental, España, 2005.

[6] Real Decreto 1367/2007. Zonificación acústica, objetivos de calidad y emisiones acústicas. Ministerio de la Presidencia, España, 23 de octubre de 2007.

[7] Manton-Harrys Cyril. Manual de Medidas Acústicas y Control del Ruido (Tercera Edición). Madrid, España, McGraw-Hill/Interamericana de España, 1995. Capítulos 5 y 11.

[8] Alonso-Olaya P, Beltrán-Vargas K, Torres-Romero J. Protocolo de evaluación y diagnóstico de emisión e inmisión de ruido en industrias. VI Congreso Iberoamericano de Acústica, Argentina, noviembre 2008.

[9] García-Sanz S. Diseño de arrays de micrófonos para aplicaciones de vigilancia, 10-13. Trabajo de Fin de Grado. Escuela Politécnica Superior de la Universidad de Alcalá, España, 2016.

[10] Alloza-Frutos P, Böck M. Aplicaciones de localización de fuentes de ruido en acústica de edificios. XI Congreso Iberoamericano de Acústica, España, 2018.

- [11] Madeira R, Alloza P, Liaño J, García G, Lamotte L, Álava Ingenieros. Micrófono direccional basado en beamforming en tiempo real. 45º Congreso español de acústica, España, 2014.
- [12] Glaeser K, Marx T, Schmidt E. Sound detection of electric vehicles by blind or visually impaired persons. *Internoise*, 2012.
- [13] Altinsoy E. The detectability of conventional, hybrid and electric vehicles sounds by sighted, visually impaired and blind pedestrians. *Internoise*, Innsbruck, 2013.
- [14] MLIT. Japanese Activities on Approaching Vehicle Audible System for HEVs and EVs. Japón, 4 de mayo 2010.
- [15] NHTSA. Minimum sound requirements for hybrid and electric vehicles-Draft environmental assessment. EEUU, Enero 2013.
- [16] Wogalter M, Raymond L, Nyeste P. On the hazard of quiet vehicles to pedestrians and drivers, *Applied ergonomics*, 45(5), 2013
- [17] Fleury S, Jamet É, Roussarie V, Bosc L, Chamard J. Effect of additional warning sounds on pedestrians' detection of electric vehicles: An ecological approach. *Accident Analysis & Prevention*, 97, 2016.
- [18] Misdariis N, Cera A, Levallois E, Locqueteau C. Do electric cars have to make noise? An emblematic opportunity for designing sounds and soundscapes. Paper Presented at *Acoustics*, 2012.
- [19] Misdariis N, Gruson A, Susini P. Detectability study of warning signals in urban background noises: a first step for designing the sound of electric vehicles. *Proceedings of Meetings on Acoustics (ICA2013)*, Canada, 2013.
- [20] Emerson R.W, Naghshineh K, Hapeman J, Wiener W. A pilot study of pedestrians with visual impairments detecting traffic gaps and surges containing hybrid vehicles. *Transp. Res. Part F. Traffic Psychol. Behav.* 14 (2), 117–127, 2011.

- [21] Sun R, Zhuang X, Wu C, Zhao G, Zhang K. The estimation of vehicle speed and stopping distance by pedestrians crossing streets in a naturalistic traffic environment. *Transp. Res. Part F*, 30, 97–106, 2015.
- [22] Chamard J-C, Roussarie V, 2012. Design of electric or hybrid vehicle alert sound system for pedestrian. *Proceedings of Acoustics, Nantes*, 1691–1695, 2012.
- [23] Parizet E, Ellermeier W, Robart R. Auditory warnings for electric vehicles: detectability in normal-vision and visually-impaired listeners. *Applied Acoustic*, 86, 50–58, 2014.
- [24] Van der Rots R, Berkhoff A. Directional loudspeaker arrays for acoustic warning systems with minimised noise pollution. *Applied Acoustics*, 89, 2014.
- [25] Poveda-Martinez P, Peral-Orts R, Campillo-Davo N, Nescolarde-Selva J, Lloret-Climent M, Ramis-Soriano J. Study of the effectiveness of electric vehicle warning sounds depending on the urban environment. *Applied Acoustics*, 116, 2017.
- [26] Steinbach L, Altinsoy M.E. Influence of an artificially produced stationary sound of electrically powered vehicles on the safety of visually impaired pedestrians. *Applied Acoustics*, 165:107290, 2020
- [27] Bodden M, Belschner T. Legal requirements and design of exterior sounds for electric vehicles. 18. *Internationales Stuttgarter Symposium*, 111-122, 2018.
- [28] Reglamento CEPE nº138, Disposiciones uniformes relativas a la homologación de vehículos silenciosos de transporte por carretera en lo relativo a su audibilidad reducida. *Diario oficial de la Unión Europea*, 13 de enero de 2017.
- [29] Norma ISO 10844:2014. Specification of test tracks for measuring noise emitted by road vehicles and their tyres. *International Organization for Standardization*, mayo 2014.
- [30] Norma CEI 61672-1:2013. Sound level meters (Part 1). Specifications. *International Electrotechnical Commission*, septiembre 2013.



[31] Norma CEI 60942-2003. Sound calibrators. Specifications. International Electrotechnical Commission, enero 2003.

[32] Facchin-Olavarría A. Estadística Básica Aplicada (Capítulo 2). Programa Básico de Formación del Corredor Inmobiliario.

## 6.2. WEBGRAFÍA

[I] Noticias Parlamento Europeo. Emisiones de CO<sub>2</sub> de los coches: hechos y cifras (infografía). Disponible en:

<https://www.europarl.europa.eu/news/es/headlines/society/20190313STO31218/emisiones-de-co2-de-los-coches-hechos-y-cifras-infografia>

[II] Movilidad eléctrica. Artículos Técnicos. Los peajes en Cataluña ya no son gratuitos para los coches eléctricos. Disponible en:

<https://movilidadelectrica.com/peajes-cataluna-coches-electricos/>

[III] Noticias Seat. Acceso a Madrid Central y ZBE Barcelona. Disponible en:

<https://www.seat.es/coches-seat/green-zones.html>

[IV] Blog de Telecomunicaciones. El nivel de presión sonora. Disponible en:

<http://tecnicoteleco.blogspot.com/2015/10/el-nivel-de-presion-sonora-spl.html>

[V] Miyara F. Biblioteca de información acústica. Disponible en:

<https://www.fceia.unr.edu.ar/acustica/biblio/niveles.htm>

[VI] NTi audio. Fast, Slow, Impulse Time Weighting. What do they mean? Disponible en:

<https://www.nti-audio.com/en/support/know-how/fast-slow-impulse-time-weighting-what-do-they-mean>

[VII] Yamaha. Which types of Microphone Are Used with Professional Audio systems?

Disponible en:

<https://asia-latinamericamea.yamaha.com/en/products/contents/proaudio/musicianspa/equipments/microphone.html>

[VIII] The Acoustic Camera. Arrays de micrófonos. Disponible en:

<https://www.acoustic-camera.com/en/products/microphone-arrays.html>

[IX] Escuelaing. Conceptos Generales del Ruido. Disponible en:

[https://www.escuelaing.edu.co/uploads/laboratorios/116\\_2\\_ruido.pdf](https://www.escuelaing.edu.co/uploads/laboratorios/116_2_ruido.pdf)

[X] Matlab. Measure sound pressure level of audio signal. Disponible en:

<https://es.mathworks.com/help/audio/ref/splmeter-system-object.html>

[XI] Teleco Pedro. Características técnicas de los micrófonos. Disponible en:

<https://sites.google.com/site/dfgm1demayopedro/home/telecomunicaciones-1/megafonia-y-sonorizacion/03-equipos-y-transductores-electroacusticos-el-microfono/3-2-caracteristicas-tecnicas-de-los-microfonos>

[XII] Departamento de Salud Pública, Medio Ambiente y Determinantes Sociales de la Salud de la OMS. Reducción de los riesgos sanitarios mundiales mediante la mitigación de los efectos de los contaminantes climáticos de vida corta. Disponible en:

[https://www.who.int/phe/health\\_topics/outdoorair/climate-reducing-health-risks-faq/es/](https://www.who.int/phe/health_topics/outdoorair/climate-reducing-health-risks-faq/es/)

[XIII] Parliament of UK. Automated and Electric Vehicles Bill. Disponible en:

<https://publications.parliament.uk/pa/cm201719/cmpublic/automated/memo/aevb25.htm>

[XIV] Lawrence Rosenblum. Hybrid Cars Are Harder to Hear. Disponible en:

[https://newsroom.ucr.edu/news\\_item.html?action=page&id=1803](https://newsroom.ucr.edu/news_item.html?action=page&id=1803)

[XV] Revista DGT. Coches eléctricos y también ruidosos. Disponible en:

<http://revista.dgt.es/es/motor/noticias/2019/07JULIO/0703-Coche-electrico-ruido.shtml#.XxDJJSgzZPY>

[XVI] Enhanced Vehicle Acoustics. External sounds to quietly-operating hybrid, electric and fuel cell vehicles. Disponible en:

<https://web.archive.org/web/20110716132014/http://evacoust.startlogic.com/product.html>

[XVII] Brigade. Quiet Vehicle Sounder/QVS. Disponible en:

<https://brigade-electronics.com/es/productos/alarmas-de-aviso-para-conduccion-marcha-atras/sistema-de-sonido-para-vehiculos-silenciosos/>

[XVIII] Kendrion. Sound Solutions for Exterior Sound. Disponible en:

<https://www.kendrion.com/en/products/sound-systems/sound-solutions-for-exterior-sound>

[XIX] Hella. Sound warning system designs for the vehicles with silent engine for PEV (pure electric vehicle), HEV (hybrid electric vehicle) and FCV (fuel cells vehicles). Disponible en:

[https://www.hella.com/e-cars/assets/media/1577\\_KI\\_AVAS%20System\\_HELLA\\_EN.pdf](https://www.hella.com/e-cars/assets/media/1577_KI_AVAS%20System_HELLA_EN.pdf)

[XX] AutomotiveIQ. Proposed Sounds for Electric Vehicles. Disponible en:

<https://www.automotive-iq.com/powertrain/articles/proposed-sounds-electric-vehicles#:~:text=Early%20examples%20include%20GM's%20'Safe,emit%20warning%20chirps%20to%20pedestrians.>

[XXI] Autoblog. Hyundai BlueOn electric vehicle unveiled in South Korea. Disponible en:

<https://www.autoblog.com/2010/09/09/hyundai-blueon-electric-vehicle-unveiled-in-south-korea/1/>

[XXII] Just-auto. Nissan makes Leaf noisier at low speed. Disponible en:

[https://www.just-auto.com/news/nissan-makes-leaf-noisier-at-low-speed\\_id104731.aspx](https://www.just-auto.com/news/nissan-makes-leaf-noisier-at-low-speed_id104731.aspx)

[XXIII] Toyota. TMC to Sell Approaching Vehicle Audible System for Prius. Disponible en:

<http://www2.toyota.co.jp/en/news/10/08/0824.html>

[XXIV] NissanPartsDeal. Speaker-Vehicle Sound For Pedestrians. Disponible en:

[https://www.nissanpartsdeal.com/parts-list/2012-nissan-leaf/body\\_electrical/speaker.html?PNC=281H0](https://www.nissanpartsdeal.com/parts-list/2012-nissan-leaf/body_electrical/speaker.html?PNC=281H0)

[XXV] Media.Jaguar. El sonido del Jaguar I-PACE protege a los peatones. Disponible en:

<https://media.jaguar.com/es-es/news/2018/10/el-sonido-del-jaguar-i-pace-protege-los-peatones>

[XXVI] Inta. Pistas de Prueba. Disponible en:

<https://www.inta.es/pistas/es/pistas/>

[XXVII] The Acoustic Camera. Ring48 AC Pro microphone array. Disponible en:

<https://www.acoustic-camera.com/en/products/microphone-arrays/ring48-ac-pro.html#:~:text=Ring48%20AC%20Pro%20%7C%20Acoustic%20Camera&text=This%20ring%20array%20is%20a,fault%20detection%20and%20quality%20control.>

[XXVIII] Baumer. LX VisualApplets cameras. Disponible en:

<https://www.baumer.com/ch/en/product-overview/industrial-cameras-image-processing/industrial-cameras/lx-visualapplets-cameras/c/14053>

[XXIX] Baumer. Sistema de adquisición de datos mcdRec. Disponible en:

<https://www.acoustic-camera.com/es/productos/adquisicion-de-datos/sistema-de-adquisicion-de-datos-mcdrec.html>

[XXX] PCE. Anemómetro termohigrómetro PCE-THA 10. Disponible en:

<https://www.pce-iberica.es/medidor-detalles-tecnicos/instrumento-de-aire/anemometro-pce-thb10.htm>

[XXXI] Renault. Características nuevas del Renault Zoe. Disponible en:

[https://cdn.group.renault.com/ren/es/modelos/nuevo-zoe/caracteristicas\\_tecnicas\\_nuevo\\_zoe\\_.pdf](https://cdn.group.renault.com/ren/es/modelos/nuevo-zoe/caracteristicas_tecnicas_nuevo_zoe_.pdf)

[XXXII] Amazon. Keling A8 Silver Portable Mini Wireless Bluetooth Speaker Outdoor Audio TF Card Mobile Phone Hands-free Call Subwoofer. Disponible en:

<https://www.amazon.ca/Portable-Wireless-Bluetooth-Hands-free-Subwoofer/dp/B01LD79X10>

[XXXIII] MathWorks. A-weighting Filter with Matlab. Hristo Zhivomirov, 17 de diciembre 2018

<https://es.mathworks.com/matlabcentral/fileexchange/46819-a-weighting-filter-with-matlab>

[XXXIV] MathWorks. Octave. Cristophe Couvreur, 29 de diciembre 1997. Disponible en:

<https://es.mathworks.com/matlabcentral/fileexchange/69-octave>

## ANEXO I. PROGRAMAS PARA ANÁLISIS DE NIVEL SONORO EQUIVALENTE MÁXIMO

### Programa extractor de datos e130uno

```
ACOUSTIC CAMERA IN MATLAB
curDirMFile = pwd
filename = fullfile(curDirMFile, [mfilename '.dat'])

% open/read file
fid = fopen(filename, 'r');

% read parameter
nNbrOfChannels = fread(fid, 1, 'long')
nNbrOfChannelsPlusPins = fread(fid, 1, 'long')

nNbrOfSamplesPerChannel = fread(fid, nNbrOfChannels, 'long')
nNbrOfDigitalPinsPerChannel = fread(fid, nNbrOfChannels, 'long');
nChannelQuantization = fread(fid, nNbrOfChannels, 'long')

nTimeFactors = fread(fid, nNbrOfChannels, 'double');
nMapFactors = fread(fid, nNbrOfChannels, 'double');

% legend names
ChannelNameLengthes = fread(fid, nNbrOfChannelsPlusPins, 'long');
IsDigitalChannelName = fread(fid, nNbrOfChannelsPlusPins, 'long');

ChNames = cell(nNbrOfChannels, 1);
i_PlusPins = 0;
nDigital = 0;
nAnalog = 0;
for i = 1:nNbrOfChannels
    % digital
    if nNbrOfDigitalPinsPerChannel(i) > 0
        for j = 1:nNbrOfDigitalPinsPerChannel(i)
            i_PlusPins = i_PlusPins + 1;
            nDigital = nDigital + 1;
            ChNames{i}{j}(1:ChannelNameLengthes(i_PlusPins)) = char(fread(fid, ChannelNameLengthes(i_PlusPins), 'char'));
            ChannelNames(i_PlusPins, 1:ChannelNameLengthes(i_PlusPins)) = ChNames{i}{j}(1:ChannelNameLengthes(i_PlusPins));
            DigitalChannelNames(nDigital, 1:ChannelNameLengthes(i_PlusPins)) = ChNames{i}{j}(1:ChannelNameLengthes(i_PlusPins));
        end
    % analog
    else
        i_PlusPins = i_PlusPins + 1;
        nAnalog = nAnalog + 1;
        ChNames{i}(1:ChannelNameLengthes(i_PlusPins)) = char(fread(fid, ChannelNameLengthes(i_PlusPins), 'char'));
        ChannelNames(i_PlusPins, 1:ChannelNameLengthes(i_PlusPins)) = ChNames{i}(1:ChannelNameLengthes(i_PlusPins));
        AnalogChannelNames(nAnalog, 1:ChannelNameLengthes(i_PlusPins)) = ChNames{i}(1:ChannelNameLengthes(i_PlusPins));
    end
end
ChannelNames

% camera (cam) position vector in local array coordinate system (x,y,z), [m]
dCameraPositionVector = fread(fid, 3, 'double')
% camera (cam) view vector in local array coordinate system (x,y,z), [m]
dCameraViewVector = fread(fid, 3, 'double');
% camera (cam) up vector in local array coordinate system (x,y,z), [m]
dCameraUpVector = fread(fid, 3, 'double');
% camera opening angle, [Grad]
dCameraOpeningAngle = fread(fid, 1, 'double')
% Focus distance to measure object (Object), [m]
dFocus = fread(fid, 1, 'double')

% sensor (mic) positions in local array coordinate system (x,y,z), [m]
for i = 1:nNbrOfChannels
    dMicPosition(i, 1:3) = zeros(1, 3, 'double');
end
for i = 1:nNbrOfChannels
    % (x,y,z) of each microfon
    dMicPosition(i, 1:3) = fread(fid, 3, 'double');
end

'data reading ... please wait'

% read data/samples
nChannels = cell(nNbrOfChannels, 1);
for i = 1:nNbrOfChannels
    % digital
    if nNbrOfDigitalPinsPerChannel(i) > 0
        for j = 1:nNbrOfDigitalPinsPerChannel(i)
```

```
nChannels{i}{j} = {zeros(1,nNbrOfSamplesPerChannel(i), 'int8')};
end
% analog
else
% 'int16' data
if nChannelQuantization(i) == 16
nChannels{i} = {zeros(1,nNbrOfSamplesPerChannel(i), 'int16')};
% 'int32' data
elseif nChannelQuantization(i) == 32
nChannels{i} = {zeros(1,nNbrOfSamplesPerChannel(i), 'int32')};
end
end
end

for i = 1:nNbrOfChannels
% digital
if nNbrOfDigitalPinsPerChannel(i) > 0
for j = 1:nNbrOfDigitalPinsPerChannel(i)
nChannels{i}{j} = fread(fid, nNbrOfSamplesPerChannel(i), 'ubit1=>int8');
fread(fid, 16-mod(nNbrOfSamplesPerChannel(i),16), 'ubit1=>int8');
end
% analog
else
% 'int16' data
if nChannelQuantization(i) == 16
nChannels{i} = fread(fid, nNbrOfSamplesPerChannel(i), 'int16');
% 'int32' data
elseif nChannelQuantization(i) == 32
nChannels{i} = fread(fid, nNbrOfSamplesPerChannel(i), 'int32');
end
end
end

% close file
status = fclose(fid);
'... data read'

OBTENCIÓN DE VALORES
for i = 1:nNbrOfChannels
% analog
if nNbrOfDigitalPinsPerChannel(i) == 0
FirstSample(i) = 1;
LastSample(i) = nNbrOfSamplesPerChannel(i);
% digital
else
FirstSample(i) = 1;
LastSample(i) = nNbrOfSamplesPerChannel(i);
%LastSample(i) = nNbrOfSamplesPerChannel(i)/16;
end
end
for i = 1:nNbrOfChannels
% analog
if nNbrOfDigitalPinsPerChannel(i) == 0
xanalog{i} = nTimeFactors(i) * (FirstSample(i)-1>LastSample(i)-1);
% xanalog = nTimeFactors(i) * (FirstSample(i)-1>LastSample(i)-1);
%break;
end
end
for i = 1:nNbrOfChannels
% digital
if nNbrOfDigitalPinsPerChannel(i) > 0
xdigital = nTimeFactors(i) * (FirstSample(i)-1>LastSample(i)-1);
break;
end
end

microphone=1;
for i = 1:microphone
nFactor = nMapFactors(i);
if nNbrOfDigitalPinsPerChannel(i) == 0
data=nFactor*nChannels{i};
%plot(xanalog, nFactor*nChannels{i}(FirstSample(i):LastSample(i)));
%plot(nChannels{i}(FirstSample(i):LastSample(i)));
end
end

'finished'
```

## Programa operaciones

### DEFINICIÓN DE VARIABLES.

```
fs=192000; %Frecuencia de muestreo.  
dataA=filterA(data, fs); %Filtrado de la señal en A.  
signalA=dataA.^2; %Elevamos la señal al cuadrado.  
tau=0.125; %Intervalo de tiempo a aplicar el salto a nuestra señal.  
p0=2*10^(-5); %Presión de referencia.  
tf=length(data)/fs; %Cálculo del instante final de tiempo.
```

### FILTRADO DE LA SEÑAL EN FAST.

```
num=1/(tau*fs); %Numerador del salto a aplicar.  
den=[1 -exp(-1/(tau*fs))]; %Denominador del salto a aplicar.  
signalfA=filter(num,den,signalA); %Filtrado fast de la señal con ponderación A.
```

### TRANSFORMACIÓN DE LA SEÑAL A dB(A).

```
signaldB=10*log10(signalfA/p0^2); %Transformación de la señal en A a dB(A).
```

### EJECUCIÓN DEL ARCHIVO QUE DETERMINA ANÓMALOS Y MÁXIMOS.

```
run('..\Tarea 1\anomalos.m') %Ejecución del script con nombre "anómalos".
```

### REPRESENTACIÓN DE LOS DATOS.

```
% GRÁFICA SIN PONDERACIÓN EN PA:
```

```
figure %Representación del gráfico en una ventana independiente.  
plot(time,data,'color',[72,61,139]/255) %Representación de los datos de la señal en pascales sin ninguna ponderación.  
xlim([0 tf]) %Determinación de los límites del eje X para la gráfica de la señal en pascales sin ninguna ponderación.  
grid minor %Colocación cuadrícula en la gráfica.  
xlabel('Tiempo [s]'); %Colocación de etiqueta en el eje X para la gráfica de la señal en pascales sin ninguna ponderación.  
ylabel('Presión sonora [Pa]'); %Colocación de etiqueta en el eje Y para la gráfica de la señal en pascales sin ninguna ponderación.  
title('SEÑAL EN PASCALES') %Colocación de título para la gráfica de la señal en pascales sin ninguna ponderación.
```

```
% GRÁFICA CON PONDERACIÓN TEMPORAL FAST Y PONDERACIÓN DE FRECUENCIA A:
```

```
figure %Representación del gráfico en una ventana independiente.  
plot(time,signalfA,'o','MarkerIndices',posmaxfA,'MarkerSize',8,'MarkerEdgeColor',[139,0,0]/255,'MarkerFaceColor',[255,99,71]/255)  
%Colocación de marcador en punto máximo para gráfico de la señal con ponderación temporal fast y ponderación de frecuencia A.  
hold on %Mantenimiento del marcador en la ventana para realizar más operaciones sin machacar lo anteriormente realizado.  
plot(time,signalfA,'color',[0,128,128]/255) %Representación de los datos de la señal con ponderación temporal fast y ponderación de frecuencia A.  
hold off %Eliminación del mantenimiento para realizar operaciones.
```

```
xlim([0 tf]) %Definición de los límites del eje X para la señal ponderada en A y fast.  
grid minor %Colocación cuadrícula en la gráfica.  
xlabel('Tiempo [s]'); %Colocación de etiqueta del eje X para el gráfico de la señal ponderada en A y fast.  
ylabel('Presión sonora [Pa^2(A)]'); %Colocación de etiqueta del eje Y para el gráfico de la señal ponderada en A y fast.  
title('SEÑAL PONDERACION TEMPORAL FAST Y PONDERACIÓN DE FRECUENCIA A') %Colocación de título para el gráfico de la señal ponderada en A y fast.
```

```
sac1=num2str(maxfA,'%10.3e'); %Transformación de datos numéricos del máximo valor numérico de la señal ponderada en A y fast a array de caracteres.  
sac2=num2str(tmaxfA,4); %Transformación de datos numéricos de tiempo de ese máximo valor de la señal ponderada a array de caracteres.  
txt1=['Lp^2 = ',sac1,' Pa^2']; %Definición del texto para marcador estableciendo valor en X.  
txto1=text(tmaxfA+0.07,tmaxfA+0.03*10^-3,txt1); %Colocación de texto en marcador del máximo valor de la señal ponderada en A y fast.  
txt2=['t = ',sac2,' s']; %Definición del texto para marcador estableciendo valor en Y.  
texto2=text(tmaxfA+0.07,tmaxfA+0.01*10^-3,txt2); %Colocación de texto en marcador del valor de tiempo del máximo valor de la señal ponderada en A y fast.
```

```
% GRÁFICA CON PONDERACIÓN TEMPORAL FAST Y dB(A):
```

```
figure %Representación del gráfico en una ventana independiente.  
plot(time,signaldB,'o','MarkerIndices',posdBAmx,'MarkerSize',8,'MarkerEdgeColor',[70,130,180]/255,'MarkerFaceColor',[135,206,235]/255)  
%Colocación de marcador en punto máximo para gráfico de la señal con ponderación temporal fast y dB(A).  
hold on %Mantenimiento del marcador en la ventana para realizar más operaciones sin machacar lo anteriormente realizado.  
plot(time,signaldB,'color',[139,0,0]/255) %Representación de los datos de la señal con ponderación temporal fast y dB(A).  
hold off %Eliminación del mantenimiento para realizar operaciones.
```

```
xlim([0 tf]) %Definición de los límites del eje X para la señal ponderada en fast y dB(A).  
grid minor %Colocación cuadrícula en la gráfica.  
xlabel('Tiempo [s]'); %Colocación de etiqueta del eje X para el gráfico de la señal ponderada en fast y dB(A).  
ylabel('Presión sonora [dB(A)]'); %Colocación de etiqueta del eje Y para el gráfico de la señal ponderada en fast y dB(A).  
title('SEÑAL PONDERACIÓN TEMPORAL FAST Y dB(A)') %Colocación de título para el gráfico de la señal ponderada en fast y dB(A).
```

```
sac3=num2str(dBAmx,3); %Transformación de datos numéricos del máximo valor numérico de la señal ponderada en fast y dB(A) a array de caracteres.  
sac4=num2str(tmaxdBA,4); %Transformación de datos numéricos de tiempo de ese máximo valor de la señal ponderada a array de caracteres.  
txt3=['LAeq = ',sac3,' dB(A)']; %Definición del texto para marcador estableciendo valor en X.  
texto3=text(tmaxdBA+0.08,dBAmx+1.1,txt3); %Colocación de texto en marcador del máximo valor de la señal ponderada en fast y dB(A).  
txt4=['t = ',sac4,' s']; %Definición del texto para marcador estableciendo valor en Y.  
texto4=text(tmaxdBA+0.08,dBAmx+0.3,txt4); %Colocación de texto en marcador del máximo valor de la señal ponderada en fast y dB(A).
```

### EJECUCIÓN DEL ARCHIVO QUE GENERA EL ESPECTRO DE FRECUENCIA DEFINIDO.

```
run('..\Tarea 1\espfrec.m') %Ejecución del script con nombre "anómalos".
```

## Programa anomalos

### DEFINICIÓN DE VARIABLES.

```
chauenetmuestras=[5*10^5; 5*10^6; 5*10^7]; %Número de muestras para coeficientes de Chauvenet.
chauenetcoef=[4.89; 5.33; 5.73]; %Coeficientes de chauvenet en funcion de número de muestras.
longsignalfA=length(signalfA); %Cálculo de la longitud de la señal para interpolar.
chauvuse=interp1(chauenetmuestras,chauenetcoef,longsignalfA); %Interpolación para sacar el coeficiente de Chauvenet que necesitamos.
med=mean(signalfA); %Cálculo de la media.
desvest=std(signalfA); %Cálculo de la desviación estándar.
signalfAa=signalfA; %Definición de una nueva variable de señal en fast para no machacar la señal (si no, el ploteo no se haría bien).
signalfAamax=max(signalfA); %Cálculo del valor máximo de la señal en fast.
distmed=abs(signalfAamax-med); %Cálculo de la distancia a la media del valor máximo de la señal en fast.
coeficiente=distmed/desvest; %Cálculo del coeficiente a comparar con Chauvenet.
```

### DETERMINACIÓN DE PRESENCIA ANÓMALOS.

```
if coeficiente>chauvuse %Primera condición donde se definen operaciones en caso de que coeficiente calculado sea mayor que el coeficiente de Chauvenet.
```

```
display('Toma de datos presenta datos anomalos') %Escribir por pantalla la presencia de datos anomalos.
```

```
while coeficiente>chauvuse %Bucle que repite la detección y eliminación de anomalos.
```

```
postimax=find(signalfAa==signalfAamax); %Determinación del punto en el que se encuentra el máximo.
timax=time(postimax); %Determinación del tiempo donde se da dicho máximo.
timenor=timax-0.125; %Parte de la señal por detrás a quitar.
postimenor=find(timenor==time); %Posición del punto de la señal por detrás a quitar.
timayor=timax+0.05; %Parte de la señal por delante a quitar.
postimayor=find(timayor==time); %Posición del punto de la señal por delante a quitar.
vectpos=[postimenor:1:postimayor]; %Vector de posiciones a quitar.
signalfAa(vectpos)=med; %Transformación de valores de posiciones a quitar de la señal fast al valor de media.
signalfAamax=max(signalfAa); %Cálculo del valor máximo de la señal en fast.
distmed=abs(signalfAamax-med); %Cálculo de la distancia a la media del valor máximo de la señal en fast.
coeficiente=distmed/desvest; %Cálculo del coeficiente a comparar con Chauvenet.
```

```
end %Cierre de bucle.
```

```
signaldBAa=10*log10(signalfAa/p0^2); %Cálculo de la señal modificada en dB(A).
```

```
maxfA=max(signalfAa); %Obtención del valor máximo de la señal fast modificada.
posmaxfA=find(signalfA==maxfA); %Posición en la que se encuentra ese máximo en la señal sin modificar.
tmaxfA=time(posmaxfA); %Obtención del tiempo para ese instante.
tmaxfAr=round(tmaxfA,2); %Redondeo a 2 decimales el instante de tiempo donde se da el valor fast máximo.
```

```
dBAmax=max(signaldBAa); %Obtención del valor dB(A) máximo en la señal modificada.
posdBAmax=find(signaldBAa==dBAmax); %Posición en la que se encuentra dicho dB(A) máximo en la señal sin modificar.
tmaxdBAr=round(tmaxdBAr,2); %Redondeo del instante de tiempo donde se da ese valor máximo a 2 decimales.
```

```
else %Segunda condición donde se definen operaciones en caso de que coeficiente calculado sea menor que el coeficiente de Chauvenet.
```

```
maxfA=max(signalfA); %Obtención del valor máximo de la señal fast sin modificar.
posmaxfA=find(signalfA==maxfA); %Posición en la que se encuentra ese máximo en la señal sin modificar.
tmaxfA=time(posmaxfA); %Obtención del tiempo para ese instante.
tmaxfAr=round(tmaxfA,2); %Redondeo a 2 decimales el instante de tiempo donde se da el valor fast máximo.
```

```
dBAmax=max(signaldBA); %Obtención del valor dB(A) máximo.
posdBAmax=find(signaldBA==dBAmax); %Posición en la que se encuentra dicho dB(A) máximo.
tmaxdBAr=round(tmaxdBAr,2); %Redondeo del instante de tiempo donde se da ese valor máximo a 2 decimales.
```

```
end %Cierre de condiciones.
```

## Programa espfrec

### DEFINICIÓN DE VARIABLES A EMPLEAR.

```
timemenor=tmaxdBAr-0.125; %Cálculo del menor valor del intervalo de tiempo a usar para determinar el espectro de frecuencia.
postmenor=find(time==timemenor); %Determinación de la posición de dicho instante en el vector de tiempo.
samplesuse=[postmenor:1:posdBAmax]; %Determinación del vector con las posiciones a emplear del vector de presión sonora en Pa.
samplesPa=data(samplesuse); %Determinación del vector presión sonora en Pa a utilizar.
```

### OBTENCIÓN DE LOS VALORES EN dB(A).

```
[B,A]=adsgn(fs); %Configura el filtro en dB(A).
samplesPafiltr=filter(B,A,samplesPa); %Filtra la señal acústica.
[P,F]=filtbank(samplesPafiltr,fs,[]); %En P calcula el nivel de presión de las bandas F.
vectfrec=1:length(F); %Vector de apoyo para hacer equidistante los valores del eje X.
```

### GRÁFICO DEL ESPECTRO DE FRECUENCIA EN TERCIOS DE OCTAVA.



```

figure % Representación del gráfico en una ventana independiente.
bar(vectfrec,P,'FaceColor',[0 .5 .5],'EdgeColor',[0 .9 .9], 'LineWidth', 1.5) % Plotea con barras el nivel de presión.
hold on %Mantenimiento de la gráfica anterior en la ventana para realizar más operaciones sin machacar lo anteriormente realizado.

if (v==1) %Primera condición donde se definen operaciones para una velocidad de 5 km/h.

    nivnorm=[40 40 40 39 38 39 40 40 41 41 41 39 37 34 31 29 26]; %Definición de los niveles de presión sonora mínimos en dB(A) según
    normativa para 5 km/h.
    plot(vectfrec,nivnorm,'-ro','LineWidth',2,'MarkerSize',7,'MarkerFaceColor',[1 .6 .6],'MarkerEdgeColor','r') %Representación de los niveles
    anteriormente definidos.

else if (v==2) %Segunda condición donde se definen operaciones para una velocidad de 10 km/h.

    nivnorm=[45 45 45 44 43 44 45 45 46 46 46 46 44 42 39 36 34 31]; %Definición de los niveles de presión sonora mínimos en dB(A) según
    normativa para 10 km/h.
    plot(vectfrec,nivnorm,'-ro','LineWidth',2,'MarkerSize',7,'MarkerFaceColor',[1 .6 .6],'MarkerEdgeColor','r') %Representación de los niveles
    anteriormente definidos.

else if (v==3) %Tercera condición donde se definen operaciones para una velocidad de 15 km/h.

    nivnorm=[48 48 48 47 46 47 48 48 49 49 49 49 47 45 42 39 37 34]; %Definición de los niveles de presión sonora mínimos en dB(A) según
    normativa para 15 km/h.
    plot(vectfrec,nivnorm,'-ro','LineWidth',2,'MarkerSize',7,'MarkerFaceColor',[1 .6 .6],'MarkerEdgeColor','r') %Representación de los niveles
    anteriormente definidos.

else if (v==4) %Cuarta condición donde se definen operaciones para una velocidad de 20 km/h.

    nivnorm=[50 50 50 49 48 49 50 50 51 51 51 51 49 47 44 41 39 36]; %Definición de los niveles de presión sonora mínimos en dB(A) según
    normativa para 20 km/h.
    plot(vectfrec,nivnorm,'-ro','LineWidth',2,'MarkerSize',7,'MarkerFaceColor',[1 .6 .6],'MarkerEdgeColor','r') %Representación de los niveles
    anteriormente definidos.

else if (v==5) %Quinta condición donde se definen operaciones para una velocidad de 25 km/h.

    nivnorm=[52 52 51.6 50.6 49.6 50.6 51.6 51.6 52.6 52.6 52.6 52.6 50.6 48.6 45.6 42.6 40.6 37.6]; %Definición de los niveles de presión sonora
    mínimos en dB(A)
    %según normativa para 25 km/h.
    plot(vectfrec,nivnorm,'-ro','LineWidth',2,'MarkerSize',7,'MarkerFaceColor',[1 .6 .6],'MarkerEdgeColor','r') %Representación de los niveles
    anteriormente definidos.

else if (v==6) %Sexta condición donde se definen operaciones para una velocidad de 30 km/h.

    nivnorm=[53 53 53 52 51 52 53 53 54 54 54 54 52 50 47 44 42 39]; %Definición de los niveles de presión sonora mínimos en dB(A) según
    normativa para 30 km/h.
    plot(vectfrec,nivnorm,'-ro','LineWidth',2,'MarkerSize',7,'MarkerFaceColor',[1 .6 .6],'MarkerEdgeColor','r') %Representación de los niveles
    anteriormente definidos.

end %Cierre de sexta condición.

end %Cierre de quinta condición.

end %Cierre de cuarta condición.

end %Cierre de tercera condición.

end %Cierre de segunda condición.

end %Cierre de condiciones.

hold off %Eliminación de mantenimiento de gráfica en ventana.
grid minor % Colocación cuadrícula en la gráfica.
set(gca,'XTick',vectfrec); % Dibuja las marcas del eje x necesarias.
set(gca,'XTickLabel',F); % Renombra las marcas con las frecuencias.
set(gca,'XTickLabelRotation',45); % Gira las marcas del eje X.
xlabel('Frecuencia [Hz]'); % Colocación de etiqueta en el eje X para el gráfico de barras.
ylabel('Presión sonora en dB(A)'); % Colocación de etiqueta en el eje Y para el gráfico de barras.
title('ESPECTRO DE FRECUENCIAS (TERCIO DE OCTAVA)') % Colocación de título para el gráfico de barras.
    
```

### Programa selectorpasadas

```

clc %Limpieza del espacio de trabajo.
clear %Limpieza de variables.

PETICIÓN WARNING SOUND PASADAS CORTAS O LARGAS.
display('1-Pasadas Cortas') %Escribir por pantalla significado del código 1.
display('2-Pasadas Largas') %Escribir por pantalla significado del código 2.

pas=input('¿Qué tipo de warning sound quiere analizar? '); %Pregunta por pantalla la clase de warning sound a analizar y lo almacena.

switch pas %Estructura de control switch que realiza las funciones que se definan para cada clase de warning sound según sean pasadas largas o
cortas.

    case 1 %Operaciones para opción Pasadas Cortas.
    
```

PETICIÓN DE WARNING SOUND PASADAS CORTAS A EJECUTAR.

```
display('E1-Warning sound E1') %Mostrar por pantalla que código es el warning sound E1.
display('E2-Warning sound E2') %Mostrar por pantalla que código es el warning sound E2.
display('E3-Warning sound E3') %Mostrar por pantalla que código es el warning sound E3.
display('E4-Warning sound E4') %Mostrar por pantalla que código es el warning sound E4.
display('E5-Warning sound E5') %Mostrar por pantalla que código es el warning sound E5.
display('E6-Warning sound E6') %Mostrar por pantalla que código es el warning sound E6.
display('C1-Warning sound C1') %Mostrar por pantalla que código es el warning sound C1.
display('C1 E-Warning sound C1 E') %Mostrar por pantalla que código es el warning sound C1 E.
display('C1 WS-Warning sound C1 WS') %Mostrar por pantalla que código es el warning sound C1 WS.
```

```
ws=input('¿Qué warning sound quiere estudiar?(En mayúsculas) ','s'); %Pregunta por pantalla el warning sound a analizar y lo almacena.
```

LISTADO PASADAS CORTAS.

```
switch ws %Estructura de control switch que realiza las funciones que se definan para cada warning sound de Pasadas Cortas.
```

```
case 'E1' %Operaciones para warning sound E1.
```

```
run('./Tarea 1\E1\obtpressoneuno.m') %Ejecución del script con nombre "obtpressoneuno".
```

```
case 'E2' %Operaciones para warning sound E2.
```

```
run('./Tarea 1\E2\obtpressonedos.m') %Ejecución del script con nombre "obtpressnedos".
```

```
case 'E3' %Operaciones para warning sound E3.
```

```
run('./Tarea 1\E3\obtpressonetres.m') %Ejecución del script con nombre "obtpressonetres".
```

```
case 'E4' %Operaciones para warning sound E4.
```

```
run('./Tarea 1\E4\obtpressonecuatro.m') %Ejecución del script con nombre "obtpressonecuatro".
```

```
case 'E5' %Operaciones para warning sound E5.
```

```
run('./Tarea 1\E5\obtpressonecinco.m') %Ejecución del script con nombre "obtpressonecinco".
```

```
case 'E6' %Operaciones para warning sound E6.
```

```
run('./Tarea 1\E6\obtpressoneseis.m') %Ejecución del script con nombre "obtpressoneseis".
```

```
case 'C1' %Operaciones para warning sound C1.
```

```
run('./Tarea 1\C1\obtpressoncuno.m') %Ejecución del script con nombre "obtpressoncuno".
```

```
case 'C1 E' %Operaciones para warning sound C1 E.
```

```
run('./Tarea 1\C1 E\obtpressoncunoe.m') %Ejecución del script con nombre "obtpressoncunoe".
```

```
case 'C1 WS' %Operaciones para warning sound C1 WS.
```

```
run('./Tarea 1\C1 WS\obtpressoncunows.m') %Ejecución del script con nombre "obtpressoncunows".
```

```
otherwise %En el caso de que el código introducido no coincida con los casos anteriores se realiza lo siguiente.
```

```
display('ERROR') %Escribir por pantalla error.
```

```
display('Introduzca un valor dentro del intervalo') %Escribir por pantalla que se debe hacer para que no de error.
```

```
return %Salida de la estructura de control.
```

```
end %Cierre de la estructura de control switch para códigos de Pasadas Cortas.
```

```
case 2 %Operaciones para opción Pasadas Largas.
```

PETICIÓN DE WARNING SOUND PASADAS LARGAS.

```
display('P1-Warning sound P1') %Mostrar por pantalla que código es el warning sound P1.
display('P2-Warning sound P2') %Mostrar por pantalla que código es el warning sound P2.
display('P3-Warning sound P3') %Mostrar por pantalla que código es el warning sound P3.
display('P4-Warning sound P4') %Mostrar por pantalla que código es el warning sound P4.
display('P5-Warning sound P5') %Mostrar por pantalla que código es el warning sound P5.
display('P8-Warning sound P8') %Mostrar por pantalla que código es el warning sound P8.
display('P9-Warning sound P9') %Mostrar por pantalla que código es el warning sound P9.
display('P10-Warning sound P10') %Mostrar por pantalla que código es el warning sound P10.
display('P11-Warning sound P11') %Mostrar por pantalla que código es el warning sound P11.
display('P12-Warning sound P12') %Mostrar por pantalla que código es el warning sound P12.
display('P13-Warning sound P13') %Mostrar por pantalla que código es el warning sound P13.
display('P16-Warning sound P16') %Mostrar por pantalla que código es el warning sound P16.
display('P17-Warning sound P17') %Mostrar por pantalla que código es el warning sound P17.
display('P18-Warning sound P18') %Mostrar por pantalla que código es el warning sound P18.
display('P19-Warning sound P19') %Mostrar por pantalla que código es el warning sound P19.
display('P20-Warning sound P20') %Mostrar por pantalla que código es el warning sound P20.
```

```
ws=input('¿Qué warning sound quiere estudiar?(En mayúsculas) ','s'); %Pregunta por pantalla el warning sound a analizar y lo almacena.
```

LISTADO PASADAS LARGAS.

`switch ws` %Estructura de control `switch` que realiza las funciones que se definan para cada warning sound de Pasadas Largas.

`case 'P1'` %Operaciones para warning sound P1.

```
run('..\Tarea 1\Pasadaslargas\P1\pasadalargauno.m') %Ejecución del script con nombre "pasadalargauno".
time=xanalog{microphone}; %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".
dBAmx1=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tf1=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P1\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Cálculo número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre "info".
```

`case 'P2'` %Operaciones para warning sound P2.

```
run('..\Tarea 1\Pasadaslargas\P2\pasadalargados.m') %Ejecución del script con nombre "pasadalargados".
time=xanalog{microphone}; %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".
dBAmx1=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tf1=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P2\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Cálculo número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre "info".
```

`case 'P3'` %Operaciones para warning sound P3.

```
run('..\Tarea 1\Pasadaslargas\P3\pasadalargatres.m') %Ejecución del script con nombre "pasadalargatres".
time=xanalog{microphone}; %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".
dBAmx1=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tf1=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P3\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Cálculo de número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre "info".
```

`case 'P4'` %Operaciones para warning sound P4.

```
run('..\Tarea 1\Pasadaslargas\P4\pasadalargacuatro.m') %Ejecución del script con nombre "pasadalargacuatro".
time=xanalog{microphone}; %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".
dBAmx1=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tf1=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P4\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Cálculo de número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre "info".
```

`case 'P5'` %Operaciones para warning sound P5.

```
run('..\Tarea 1\Pasadaslargas\P5\pasadalargacinco.m') %Ejecución del script con nombre "pasadalargacinco".
time=xanalog{microphone}; %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".
dBAmx1=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tf1=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P5\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Número de archivos con extensión .m
run('..\Tarea 1\info.m') %Ejecución del script con nombre "info".
```

`case 'P8'` %Operaciones para warning sound P8.

```
run('..\Tarea 1\Pasadaslargas\P8\pasadalargaocho.m') %Ejecución del script con nombre "pasadalargaocho".
time=xanalog{microphone}; %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".
dBAmx1=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tf1=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir('..\Tarea 1\Pasadaslargas\P8\*.m'); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Número de archivos con extensión .m
```

```
run(../Tarea 1/info.m) %Ejecución del script con nombre "info".
```

```
case 'P9' %Operaciones para warning sound P9.
```

```
run(../Tarea 1/Pasadaslargas/P9/pasadalgarganueve.m) %Ejecución del script con nombre "pasadalarganueve".  
time=xanalog{microphone}; %Almacenamiento del vector tiempo en variable.  
v=4; %Definición del código de velocidad para este caso.  
run(../Tarea 1/operaciones.m) %Ejecución del script con nombre "operaciones".  
dBAmaxl=dBAmacr; %Almacenamiento del valor máximo de dB(A) redondeado en variable.  
t1=tmaxdBar; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.  
tf1=tf; %Almacenamiento del valor del último instante del vector tiempo.  
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.  
arch=dir(../Tarea 1/Pasadaslargas/P9/*.m); %Obtención de lista de archivos con extensión .m  
numarch=length(arch); %Número de archivos con extensión .m  
run(../Tarea 1/info.m) %Ejecución del script con nombre "info".
```

```
case 'P10' %Operaciones para warning sound P10.
```

```
run(../Tarea 1/Pasadaslargas/P10/pasadalgargadiez.m) %Ejecución del script con nombre "pasadalargadiez".  
time=xanalog{microphone}; %Almacenamiento del vector tiempo en variable.  
v=4; %Definición del código de velocidad para este caso.  
run(../Tarea 1/operaciones.m) %Ejecución del script con nombre "operaciones".  
dBAmaxl=dBAmacr; %Almacenamiento del valor máximo de dB(A) redondeado en variable.  
t1=tmaxdBar; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.  
tf1=tf; %Almacenamiento del valor del último instante del vector tiempo.  
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.  
arch=dir(../Tarea 1/Pasadaslargas/P10/*.m); %Obtención de lista de archivos con extensión .m  
numarch=length(arch); %Número de archivos con extensión .m  
run(../Tarea 1/info.m) %Ejecución del script con nombre "info".
```

```
case 'P11' %Operaciones para warning sound P11.
```

```
run(../Tarea 1/Pasadaslargas/P11/pasadalgargaonce.m) %Ejecución del script con nombre "pasadalargaonce".  
time=xanalog{microphone}; %Almacenamiento del vector tiempo en variable.  
v=4; %Definición del código de velocidad para este caso.  
run(../Tarea 1/operaciones.m) %Ejecución del script con nombre "operaciones".  
dBAmaxl=dBAmacr; %Almacenamiento del valor máximo de dB(A) redondeado en variable.  
t1=tmaxdBar; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.  
tf1=tf; %Almacenamiento del valor del último instante del vector tiempo.  
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.  
arch=dir(../Tarea 1/Pasadaslargas/P11/*.m); %Obtención de lista de archivos con extensión .m  
numarch=length(arch); %Número de archivos con extensión .m  
run(../Tarea 1/info.m) %Ejecución del script con nombre "info".
```

```
case 'P12' %Operaciones para warning sound P12.
```

```
run(../Tarea 1/Pasadaslargas/P12/pasadalgargadoce.m) %Ejecución del script con nombre "pasadalargadoce".  
time=xanalog{microphone}; %Almacenamiento del vector tiempo en variable.  
v=4; %Definición del código de velocidad para este caso.  
run(../Tarea 1/operaciones.m) %Ejecución del script con nombre "operaciones".  
dBAmaxl=dBAmacr; %Almacenamiento del valor máximo de dB(A) redondeado en variable.  
t1=tmaxdBar; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.  
tf1=tf; %Almacenamiento del valor del último instante del vector tiempo.  
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.  
arch=dir(../Tarea 1/Pasadaslargas/P12/*.m); %Obtención de lista de archivos con extensión .m  
numarch=length(arch); %Número de archivos con extensión .m  
run(../Tarea 1/info.m) %Ejecución del script con nombre "info".
```

```
case 'P13' %Operaciones para warning sound P13.
```

```
run(../Tarea 1/Pasadaslargas/P13/pasadalgargatrece.m) %Ejecución del script con nombre "pasadalargatrece".  
time=xanalog{microphone}; %Almacenamiento del vector tiempo en variable.  
v=4; %Definición del código de velocidad para este caso.  
run(../Tarea 1/operaciones.m) %Ejecución del script con nombre "operaciones".  
dBAmaxl=dBAmacr; %Almacenamiento del valor máximo de dB(A) redondeado en variable.  
t1=tmaxdBar; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.  
tf1=tf; %Almacenamiento del valor del último instante del vector tiempo.  
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.  
arch=dir(../Tarea 1/Pasadaslargas/P13/*.m); %Obtención de lista de archivos con extensión .m  
numarch=length(arch); %Número de archivos con extensión .m  
run(../Tarea 1/info.m) %Ejecución del script con nombre "info".
```

```
case 'P14' %Operaciones para warning sound P14.
```

```
run(../Tarea 1/Pasadaslargas/P14/pasadalgargatorce.m) %Ejecución del script con nombre "pasadalargatorce".  
time=xanalog{microphone}; %Almacenamiento del vector tiempo en variable.  
v=4; %Definición del código de velocidad para este caso.  
run(../Tarea 1/operaciones.m) %Ejecución del script con nombre "operaciones".  
dBAmaxl=dBAmacr; %Almacenamiento del valor máximo de dB(A) redondeado en variable.  
t1=tmaxdBar; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.  
tf1=tf; %Almacenamiento del valor del último instante del vector tiempo.  
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.  
arch=dir(../Tarea 1/Pasadaslargas/P14/*.m); %Obtención de lista de archivos con extensión .m  
numarch=length(arch); %Número de archivos con extensión .m  
run(../Tarea 1/info.m) %Ejecución del script con nombre "info".
```

```
case 'P16' %Operaciones para warning sound P16.

run(..\Tarea 1\Pasadaslargas\P16\pasadalargadieiciseis.m) %Ejecución del script con nombre "pasadalargadieiciseis".
time=xanalog{microphone}; %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run(..\Tarea 1\operaciones.m) %Ejecución del script con nombre "operaciones".
dBAmxl=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBar; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir(..\Tarea 1\Pasadaslargas\P16*.m); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Número de archivos con extensión .m
run(..\Tarea 1\info.m) %Ejecución del script con nombre "info".

case 'P17' %Operaciones para warning sound P17.

run(..\Tarea 1\Pasadaslargas\P17\pasadalargadieicisiete.m) %Ejecución del script con nombre "pasadalargadieicisiete".
time=xanalog{microphone}; %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run(..\Tarea 1\operaciones.m) %Ejecución del script con nombre "operaciones".
dBAmxl=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBar; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir(..\Tarea 1\Pasadaslargas\P17*.m); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Número de archivos con extensión .m
run(..\Tarea 1\info.m) %Ejecución del script con nombre "info".

case 'P18' %Operaciones para warning sound P18.

run(..\Tarea 1\Pasadaslargas\P18\pasadalargadieiciocho.m) %Ejecución del script con nombre "pasadalargadieiciocho".
time=xanalog{microphone}; %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run(..\Tarea 1\operaciones.m) %Ejecución del script con nombre "operaciones".
dBAmxl=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBar; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir(..\Tarea 1\Pasadaslargas\P18*.m); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Número de archivos con extensión .m
run(..\Tarea 1\info.m) %Ejecución del script con nombre "info".

case 'P19' %Operaciones para warning sound P19.

run(..\Tarea 1\Pasadaslargas\P19\pasadalargadieicinueve.m) %Ejecución del script con nombre "pasadalargadieicinueve".
time=xanalog{microphone}; %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run(..\Tarea 1\operaciones.m) %Ejecución del script con nombre "operaciones".
dBAmxl=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBar; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir(..\Tarea 1\Pasadaslargas\P19*.m); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Número de archivos con extensión .m
run(..\Tarea 1\info.m) %Ejecución del script con nombre "info".

case 'P20' %Operaciones para warning sound P20.

run(..\Tarea 1\Pasadaslargas\P20\pasadalargaveinte.m) %Ejecución del script con nombre "pasadalargaveinte".
time=xanalog{microphone}; %Almacenamiento del vector tiempo en variable.
v=4; %Definición del código de velocidad para este caso.
run(..\Tarea 1\operaciones.m) %Ejecución del script con nombre "operaciones".
dBAmxl=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable.
t1=tmaxdBar; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable.
tfl=tf; %Almacenamiento del valor del último instante del vector tiempo.
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava.
arch=dir(..\Tarea 1\Pasadaslargas\P20*.m); %Obtención de lista de archivos con extensión .m
numarch=length(arch); %Número de archivos con extensión .m
run(..\Tarea 1\info.m) %Ejecución del script con nombre "info".

otherwise %En el caso de que el código introducido no coincida con los casos anteriores se realiza lo siguiente.

display (ERROR) %Escribir por pantalla error.
display (Introduzca un valor dentro del intervalo) %Escribir por pantalla que se debe hacer para que no de error.
return %Salga de la estructura de control.

end %Cierre de la estructura de control switch para códigos de Pasadas Largas.

otherwise %En el caso de que el código introducido no coincida con los casos anteriores se realiza lo siguiente.

display (ERROR) %Escribir por pantalla error.
display (Introduzca un valor dentro del intervalo) %Escribir por pantalla que se debe hacer para que no de error.
return %Salga de la estructura de control.
```

```
end %Cierre de la estructura de control switch para tipos de pasadas.
```

### Programa obtpressoneuno

```
PETICIÓN DE VELOCIDAD A EJECUTAR.
display('1-Velocidad a 5 km/h') %Mostrar por pantalla que código es la velocidad a 5 km/h.
display('2-Velocidad a 10 km/h') %Mostrar por pantalla que código es la velocidad a 10 km/h.
display('3-Velocidad a 15 km/h') %Mostrar por pantalla que código es la velocidad a 15 km/h.
display('4-Velocidad a 20 km/h') %Mostrar por pantalla que código es la velocidad a 20 km/h.
display('5-Velocidad a 25 km/h') %Mostrar por pantalla que código es la velocidad a 25 km/h.
display('6-Velocidad a 30 km/h') %Mostrar por pantalla que código es la velocidad a 30 km/h.

v=input('¿Qué velocidad quiere estudiar? '); %Pregunta por pantalla código de velocidad a estudiar y almacenamiento en variable.

LISTADO VELOCIDADES
switch v %Estructura de control switch que realiza las funciones que se definan para cada caso de velocidad para warning sound E1.

    case 1 %Operaciones para velocidad de 5 km/h.

        arch=dir('C1 E1 5\*.m'); %Obtención de lista de archivos con extensión .m para 5 km/h.
        numarch=length(arch); %Número de archivos que permiten obtener los datos con extensión .m para 5 km/h.

        for i=1:numarch %Bucle for para cambiar el valor de i (i es el número de ensayo) para esa velocidad.

            switch i %Estructura de control switch que realice las funciones especificadas para cada ensayo de 5 km/h.

                case 1 %Operaciones para primer ensayo a 5 km/h.

                    run('..E1\C1 E1 5\15uno.m') %Ejecución del archivo del primer ensayo.
                    pressonpa1=data; %Almacenamiento de los datos de presión sonora a esa velocidad del primer ensayo en variable.
                    time1=xanalog{microphone}; %Almacenamiento de los datos de tiempo a esa velocidad del primer ensayo en variable.

                case 2 %Operaciones para segundo ensayo a 5 km/h.

                    run('..E1\C1 E1 5\15dos.m') %Ejecución del archivo del segundo ensayo.
                    pressonpa2=data; %Almacenamiento de los datos de presión sonora a esa velocidad del segundo ensayo en variable.
                    time2=xanalog{microphone}; %Almacenamiento de los datos de tiempo a esa velocidad del segundo ensayo en variable.

            end %Cierre para los ensayos a 5 km/h.

        end %Cierre del bucle.

    case 2 %Operaciones para velocidad de 10 km/h.

        arch=dir('C1 E1 10\*.m'); %Obtención de lista de archivos con extensión .m para 10 km/h.
        numarch=length(arch); %Número de archivos que permiten obtener los datos con extensión .m para 10 km/h.

        for i=1:numarch %Bucle for para cambiar el valor de i (i es el número de ensayo) para esa velocidad.

            switch i %Estructura de control switch que realice las funciones especificadas para cada ensayo de 5 km/h.

                case 1 %Operaciones para primer ensayo a 10 km/h.

                    run('..E1\C1 E1 10\110uno.m') %Ejecución del archivo del primer ensayo.
                    pressonpa1=data; %Almacenamiento de los datos de presión sonora a esa velocidad del primer ensayo en variable.
                    time1=xanalog{microphone}; %Almacenamiento de los datos de tiempo a esa velocidad del primer ensayo en variable.

            end %Cierre para los ensayos a 10 km/h.

        end %Cierre del bucle.

    case 3 %Operaciones para velocidad de 15 km/h.

        arch=dir('C1 E1 15\*.m'); %Obtención de lista de archivos con extensión .m para 15 km/h.
        numarch=length(arch); %Número de archivos que permiten obtener los datos con extensión .m para 15 km/h.

        for i=1:numarch %Bucle for para cambiar el valor de i (i es el número de ensayo) para esa velocidad.

            switch i %Estructura de control switch que realice las funciones especificadas para cada ensayo de 15 km/h.

                case 1 %Operaciones para primer ensayo a 15 km/h.

                    run('..E1\C1 E1 15\115uno.m') %Ejecución del archivo del primer ensayo.
                    pressonpa1=data; %Almacenamiento de los datos de presión sonora a esa velocidad del primer ensayo en variable.
                    time1=xanalog{microphone}; %Almacenamiento de los datos de tiempo a esa velocidad del primer ensayo en variable.

                case 2 %Operaciones para segundo ensayo a 15 km/h.

                    run('..E1\C1 E1 15\115dos.m') %Ejecución del archivo del segundo ensayo.
                    pressonpa2=data; %Almacenamiento de los datos de presión sonora a esa velocidad del segundo ensayo en variable.
```

```
time2=xanalog{microphone}; % Almacenamiento de los datos de tiempo a esa velocidad del segundo ensayo en variable.

end % Cierre para los ensayos a 15 km/h.

end % Cierre del bucle.

case 4 % Operaciones para velocidad de 20 km/h.

arch=dir('C1 E1 20*.m'); % Obtención de lista de archivos con extensión .m para 20 km/h.
numarch=length(arch); % Número de archivos que permiten obtener los datos con extensión .m para 20 km/h.

for i=1:numarch % Estructura de control switch que realice las funciones especificadas para cada ensayo de 20 km/h.

switch i % Estructura de control switch que realice las funciones especificadas para cada ensayo de 20 km/h.

case 1 % Operaciones para segundo ensayo a 20 km/h.

run('..\E1\C1 E1 20\120uno.m') % Ejecución del archivo del primer ensayo.
pressonpa1=data; % Almacenamiento de los datos de presión sonora a esa velocidad del primer ensayo en variable.
time1=xanalog{microphone}; % Almacenamiento de los datos de tiempo a esa velocidad del primer ensayo en variable.

case 2 % Operaciones para segundo ensayo a 20 km/h.

run('..\E1\C1 E1 20\120dos.m') % Ejecución del archivo del segundo ensayo.
pressonpa2=data; % Almacenamiento de los datos de presión sonora a esa velocidad del segundo ensayo en variable.
time2=xanalog{microphone}; % Almacenamiento de los datos de tiempo a esa velocidad del segundo ensayo en variable.

end % Cierre para los ensayos a 20 km/h.

end % Cierre del bucle.

case 5 % Operaciones para velocidad de 25 km/h.

arch=dir('C1 E1 25*.m'); % Obtención de lista de archivos con extensión .m para 25 km/h.
numarch=length(arch); % Número de archivos que permiten obtener los datos con extensión .m para 25 km/h.

for i=1:numarch % Estructura de control switch que realice las funciones especificadas para cada ensayo de 25 km/h.

switch i % Estructura de control switch que realice las funciones especificadas para cada ensayo de 25 km/h.

case 1 % Operaciones para segundo ensayo a 25 km/h.

run('..\E1\C1 E1 25\125uno.m') % Ejecución del archivo del primer ensayo.
pressonpa1=data; % Almacenamiento de los datos de presión sonora a esa velocidad del primer ensayo en variable.
time1=xanalog{microphone}; % Almacenamiento de los datos de tiempo a esa velocidad del primer ensayo en variable.

case 2 % Operaciones para segundo ensayo a 25 km/h.

run('..\E1\C1 E1 25\125dos.m') % Ejecución del archivo del segundo ensayo.
pressonpa2=data; % Almacenamiento de los datos de presión sonora a esa velocidad del segundo ensayo en variable.
time2=xanalog{microphone}; % Almacenamiento de los datos de tiempo a esa velocidad del segundo ensayo en variable.

end % Cierre para los ensayos a 25 km/h.

end % Cierre del bucle.

case 6 % Operaciones para velocidad de 30 km/h.

arch=dir('C1 E1 30*.m'); % Obtención de lista de archivos con extensión .m para 30 km/h.
numarch=length(arch); % Número de archivos que permiten obtener los datos con extensión .m para 30 km/h.

for i=1:numarch % Estructura de control switch que realice las funciones especificadas para cada ensayo de 30 km/h.

switch i % Estructura de control switch que realice las funciones especificadas para cada ensayo de 30 km/h.

case 1 % Operaciones para segundo ensayo a 30 km/h.

run('..\E1\C1 E1 30\130uno.m') % Ejecución del archivo del primer ensayo.
pressonpa1=data; % Almacenamiento de los datos de presión sonora a esa velocidad del primer ensayo en variable.
time1=xanalog{microphone}; % Almacenamiento de los datos de tiempo a esa velocidad del primer ensayo en variable.

case 2 % Operaciones para segundo ensayo a 30 km/h.

run('..\E1\C1 E1 30\130dos.m') % Ejecución del archivo del segundo ensayo.
pressonpa2=data; % Almacenamiento de los datos de presión sonora a esa velocidad del segundo ensayo en variable.
time2=xanalog{microphone}; % Almacenamiento de los datos de tiempo a esa velocidad del segundo ensayo en variable.

end % Cierre para los ensayos a 30 km/h.

end % Cierre del bucle.

otherwise % En el caso de que el código introducido no coincida con los casos anteriores se realiza lo siguiente.

display('ERROR') % Escribir por pantalla error.
display('Introduzca un valor dentro del intervalo especificado') % Escribir por pantalla que se debe hacer para que no de error.
```

```
return %Salga de la estructura de control.  
  
end %Cierre de la estructura de control para E1.  
  
run('..\Tarea 1\reps.m') %Ejecución del script con nombre "reps".
```

## Programa reps

```
DEFINICIÓN DEL VECTOR PRESIÓN PARA ENTRAR EN EL BUCLE.  
switch numarch %Estructura de control switch que realiza las funciones que se definan para cada caso en función del número de archivos según su extensión.  
  
case 1 %Operaciones para casos con 1 archivo de extensión determinada.  
  
Ejecución del script para el primer archivo.  
data=pressonpa1; %Almacenamiento datos vector presión sonora en variable del primer archivo.  
time=time1; %Almacenamiento datos vector tiempo en variable del primer archivo.  
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".  
dBAmx1=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable del primer archivo.  
t1=tmaxdBA; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable del primer archivo.  
tf1=tf; %Almacenamiento del valor del último instante del vector tiempo del primer archivo.  
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava del primer archivo.  
  
case 2 %Operaciones para casos con 2 archivos de extensión determinada.  
  
Ejecución del script para el primer archivo.  
data=pressonpa1; %Almacenamiento datos vector presión sonora en variable del primer archivo.  
time=time1; %Almacenamiento datos vector tiempo en variable del primer archivo.  
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".  
dBAmx1=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable del primer archivo.  
t1=tmaxdBA; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable del primer archivo.  
tf1=tf; %Almacenamiento del valor del último instante del vector tiempo del primer archivo.  
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava del primer archivo.  
  
Ejecución del script para el segundo archivo.  
data=pressonpa2; %Almacenamiento datos vector presión sonora en variable del segundo archivo.  
time=time2; %Almacenamiento datos vector tiempo en variable del segundo archivo.  
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".  
dBAmx2=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable del segundo archivo.  
t2=tmaxdBA; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable del segundo archivo.  
tf2=tf; %Almacenamiento del valor del último instante del vector tiempo del segundo archivo.  
P2=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava del segundo archivo.  
  
case 3 %Operaciones para casos con 3 archivos de extensión determinada.  
  
Ejecución del script para el primer archivo.  
data=pressonpa1; %Almacenamiento datos vector presión sonora en variable del primer archivo.  
time=time1; %Almacenamiento datos vector tiempo en variable del primer archivo.  
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".  
dBAmx1=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable del primer archivo.  
t1=tmaxdBA; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable del primer archivo.  
tf1=tf; %Almacenamiento del valor del último instante del vector tiempo del primer archivo.  
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava del primer archivo.  
  
Ejecución del script para el segundo archivo.  
data=pressonpa2; %Almacenamiento datos vector presión sonora en variable del segundo archivo.  
time=time2; %Almacenamiento datos vector tiempo en variable del segundo archivo.  
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".  
dBAmx2=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable del segundo archivo.  
t2=tmaxdBA; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable del segundo archivo.  
tf2=tf; %Almacenamiento del valor del último instante del vector tiempo del segundo archivo.  
P2=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava del segundo archivo.  
  
Ejecución del script para el tercer archivo.  
data=pressonpa3; %Almacenamiento datos vector presión sonora en variable del tercer archivo.  
time=time3; %Almacenamiento datos vector tiempo en variable del tercer archivo.  
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".  
dBAmx3=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable del tercer archivo.  
t3=tmaxdBA; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable del tercer archivo.  
tf3=tf; %Almacenamiento del valor del último instante del vector tiempo del tercer archivo.  
P3=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava del tercer archivo.  
  
case 4 %Operaciones para casos con 4 archivos de extensión determinada.  
  
Ejecución del script para el primer archivo.  
data=pressonpa1; %Almacenamiento datos vector presión sonora en variable del primer archivo.  
time=time1; %Almacenamiento datos vector tiempo en variable del primer archivo.  
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".  
dBAmx1=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable del primer archivo.  
t1=tmaxdBA; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable del primer archivo.  
tf1=tf; %Almacenamiento del valor del último instante del vector tiempo del primer archivo.
```



```
P1=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava del primer archivo.

Ejecución del script para el segundo archivo.
data=pressonpa2; %Almacenamiento datos vector presión sonora en variable del segundo archivo.
time=time2; %Almacenamiento datos vector tiempo en variable del segundo archivo.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".
dBAmx2=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable del segundo archivo.
t2=tmaxdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable del segundo archivo.
tf2=tf; %Almacenamiento del valor del último instante del vector tiempo del segundo archivo.
P2=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava del segundo archivo.

Ejecución del script para el tercer archivo.
data=pressonpa3; %Almacenamiento datos vector presión sonora en variable del tercer archivo.
time=time3; %Almacenamiento datos vector tiempo en variable del tercer archivo.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".
dBAmx3=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable del tercer archivo.
t3=tmaxdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable del tercer archivo.
tf3=tf; %Almacenamiento del valor del último instante del vector tiempo del tercer archivo.
P3=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava del tercer archivo.

Ejecución del script para el cuarto archivo.
data=pressonpa4; %Almacenamiento datos vector presión sonora en variable del cuarto archivo.
time=time4; %Almacenamiento datos vector tiempo en variable del cuarto archivo.
run('..\Tarea 1\operaciones.m') %Ejecución del script con nombre "operaciones".
dBAmx4=dBAmx; %Almacenamiento del valor máximo de dB(A) redondeado en variable del cuarto archivo.
t4=tmaxdBAr; %Almacenamiento del valor de tiempo de ese máximo dB(A) redondeado en variable del cuarto archivo.
tf4=tf; %Almacenamiento del valor del último instante del vector tiempo del cuarto archivo.
P4=P; %Almacenamiento de presión para cada banda de frecuencia de tercio de octava del cuarto archivo.

end %Cierre switch que realiza las funciones que se definan para cada caso en función del número de archivos según su extensión.

EJECUCIÓN PROGRAMA INFO.
run('..\Tarea 1\info.m') %Ejecución del script con nombre "info".
```

## Programa info

```
DEFINICIÓN DE VARIABLES DE PARTIDA.
tipvar = {'string','double','double','double'}; %Determinación del tipo de variable para cada columna de la tabla.
nomvar = {'Recording','dBA','Instant','Total'}; %Asignación del nombre de encabezado de columna.
tamano=[1 4]; %Definición inicial del tamaño de la tabla de información.
tabla = table('Size',tamano,'VariableTypes',tipvar,'VariableNames',nomvar); %Generación de tabla informativa.

DESARROLLO DE LA TABLA INFORMATIVA.
switch numarch %Estructura de control switch que realiza las funciones que se definan para cada caso en función del número de archivos según su extensión.

case 1 %Operaciones para casos con 1 archivo de extensión determinada.

archivo1=arch.name; %Almacenamiento del nombre del archivo .m correspondiente a la primera pasada.
tamano = [1 4]; %Definición del tamaño de la tabla para este caso.
tabla(1,:) = {archivo1,dBAmx1,t1,tf1} %Establecimiento de los valores que irá en la primera fila de la tabla.

case 2 %Operaciones para casos con 2 archivos de extensión determinada.

arch1=arch(1); %Selección del primer archivo .m correspondiente a la primera pasada.
arch2=arch(2); %Selección del primer archivo .m correspondiente a la segunda pasada.
archivo1=arch1.name; %Almacenamiento del nombre del archivo .m correspondiente a la primera pasada.
archivo2=arch2.name; %Almacenamiento del nombre del archivo .m correspondiente a la segunda pasada.
tamano = [2 4]; %Definición del tamaño de la tabla para este caso.
tabla(1,:) = {archivo2,dBAmx1,t1,tf1} %Establecimiento de los valores que irá en la primera fila de la tabla.
tabla(2,:) = {archivo1,dBAmx2,t2,tf2} %Establecimiento de los valores que irá en la segunda fila de la tabla.

case 3 %Operaciones para casos con 3 archivos de extensión determinada.

arch1=arch(1); %Selección del primer archivo .m correspondiente a la primera pasada.
arch2=arch(2); %Selección del primer archivo .m correspondiente a la segunda pasada.
arch3=arch(3); %Selección del primer archivo .m correspondiente a la tercera pasada.
archivo1=arch1.name; %Almacenamiento del nombre del archivo .m correspondiente a la primera pasada.
archivo2=arch2.name; %Almacenamiento del nombre del archivo .m correspondiente a la segunda pasada.
archivo3=arch3.name; %Almacenamiento del nombre del archivo .m correspondiente a la tercera pasada.
tamano = [3 4]; %Definición del tamaño de la tabla para este caso.
tabla(1,:) = {archivo3,dBAmx1,t1,tf1} %Establecimiento de los valores que irá en la primera fila de la tabla.
tabla(2,:) = {archivo2,dBAmx2,t2,tf2} %Establecimiento de los valores que irá en la segunda fila de la tabla.
tabla(3,:) = {archivo1,dBAmx3,t3,tf3} %Establecimiento de los valores que irá en la tercera fila de la tabla.

case 4 %Operaciones para casos con 4 archivos de extensión determinada.

arch1=arch(1); %Selección del primer archivo .m correspondiente a la primera pasada.
arch2=arch(2); %Selección del primer archivo .m correspondiente a la segunda pasada.
arch3=arch(3); %Selección del primer archivo .m correspondiente a la tercera pasada.
```

```
arch4=arch(4); %Selección del primer archivo .m correspondiente a la cuarta pasada.
archivo1=arch1.name; %Almacenamiento del nombre del archivo .m correspondiente a la primera pasada.
archivo2=arch2.name; %Almacenamiento del nombre del archivo .m correspondiente a la segunda pasada.
archivo3=arch3.name; %Almacenamiento del nombre del archivo .m correspondiente a la tercera pasada.
archivo4=arch4.name; %Almacenamiento del nombre del archivo .m correspondiente a la cuarta pasada.
tamano = [4 4]; %Definición del tamaño de la tabla para este caso.
tabla(1,:) = {archivo4,dBAmax1,t1,tf1} %Establecimiento de los valores que irá en la primera fila de la tabla.
tabla(2,:) = {archivo2,dBAmax2,t2,tf2} %Establecimiento de los valores que irá en la segunda fila de la tabla.
tabla(3,:) = {archivo3,dBAmax3,t3,tf3} %Establecimiento de los valores que irá en la tercera fila de la tabla.
tabla(4,:) = {archivo1,dBAmax4,t4,tf4} %Establecimiento de los valores que irá en la cuarta fila de la tabla.

end %Cierre de la estructura de control switch para generación de las tablas.

EJECUCIÓN DEL PROGRAMA SELECTOREXC.
run('..\Tarea 1\selectorexc.m') %Ejecución programa selectorexc.
```

### Programa selectorexc

```
SELECCIÓN DE LA RUTA A PROGRAMA TRANSFERIDOR DE DATOS A HOJA DE EXCEL.
switch pas %Estructura de control switch que realiza las funciones que se definan para cada clase de warning sound según sean pasadas largas o cortas.

LISTADO PASADAS CORTAS.
case 1 %Operaciones para opción Pasadas Cortas.

switch ws %Estructura de control switch que realiza las funciones que se definan para cada warning sound de Pasadas Cortas.

case 'E1' %Operaciones para warning sound E1.

run('..\Tarea 1\E1\velexce1.m') %Ejecución del script con nombre "velexce1".

case 'E2' %Operaciones para warning sound E2.

run('..\Tarea 1\E2\velexce2.m') %Ejecución del script con nombre "velexce2".

case 'E3' %Operaciones para warning sound E3.

run('..\Tarea 1\E3\velexce3.m') %Ejecución del script con nombre "velexce3".

case 'E4' %Operaciones para warning sound E4.

run('..\Tarea 1\E4\velexce4.m') %Ejecución del script con nombre "velexce4".

case 'E5' %Operaciones para warning sound E5.

run('..\Tarea 1\E5\velexce5.m') %Ejecución del script con nombre "velexce5".

case 'E6' %Operaciones para warning sound E6.

run('..\Tarea 1\E6\velexce6.m') %Ejecución del script con nombre "velexce6".

case 'C1' %Operaciones para warning sound C1.

run('..\Tarea 1\C1\velexcc1.m') %Ejecución del script con nombre "velexcc1".

case 'C1 E' %Operaciones para warning sound C1 E.

run('..\Tarea 1\C1 E\velexcc1e.m') %Ejecución del script con nombre "velexcc1e".

case 'C1 WS' %Operaciones para warning sound C1 WS.

run('..\Tarea 1\C1 WS\velexcc1ws.m') %Ejecución del script con nombre "velexcc1ws".

end %Cierre de la estructura de control switch para códigos de Pasadas Cortas.

LISTADO PASADAS LARGAS.
case 2 %Operaciones para opción Pasadas Largas.

switch ws %Estructura de control switch que realiza las funciones que se definan para cada warning sound de Pasadas Cortas.

case 'P1' %Operaciones para warning sound P1.

filename='..\Tarea 1\Pasadaslargas\P1\p1.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P1','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.
textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F;P1]; %Almacenamiento del espectro de frecuencias en variable.
hoja1='Espectro frecuencia P1'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
```

```
xlswrite(filename,txbandf,hoja1); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 'P2' %Operaciones para warning sound P1.

filename='.\Tarea 1\Pasadaslargas\P2\p2.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P2','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.
txbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F;P1]; %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
hoja1='Espectro frecuencia P2'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,txbandf,hoja1); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 'P3' %Operaciones para warning sound P3.

filename='.\Tarea 1\Pasadaslargas\P3\p3.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P3','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.
txbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F;P1]; %Almacenamiento del espectro de frecuencias en variable.
hoja1='Espectro frecuencia P3'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,txbandf,hoja1); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 'P4' %Operaciones para warning sound P4.

filename='.\Tarea 1\Pasadaslargas\P4\p4.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P4','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.
txbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F;P1]; %Almacenamiento del espectro de frecuencias en variable.
hoja1='Espectro frecuencia P4'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,txbandf,hoja1); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 'P5' %Operaciones para warning sound P5.

filename='.\Tarea 1\Pasadaslargas\P5\p5.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P5','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.
txbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F;P1]; %Almacenamiento del espectro de frecuencias en variable.
hoja1='Espectro frecuencia P5'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,txbandf,hoja1); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 'P8' %Operaciones para warning sound P8.

filename='.\Tarea 1\Pasadaslargas\P8\p8.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P8','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.
txbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F;P1]; %Almacenamiento del espectro de frecuencias en variable.
hoja1='Espectro frecuencia P8'; %Establecimiento de nombre la hoja.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,txbandf,hoja1); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 'P9' %Operaciones para warning sound P9.

filename='.\Tarea 1\Pasadaslargas\P9\p9.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P9','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.
txbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F;P1]; %Almacenamiento del espectro de frecuencias en variable.
hoja1='Espectro frecuencia P9'; %Posición donde colocar el espectro de frecuencia.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,txbandf,hoja1); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.

case 'P10' %Operaciones para warning sound P10.

filename='.\Tarea 1\Pasadaslargas\P10\p10.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
```

```
writetable(tabla,filename,'Sheet','Datos Ensayo P10','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.
textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F;P1]; %Almacenamiento del espectro de frecuencias en variable.
hoja1='Espectro frecuencia P10'; %Posición donde colocar el espectro de frecuencia.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hoja1); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.
```

case 'P11' %Operaciones para warning sound P11.

```
filename='.\Tarea 1\Pasadaslargas\P11p11.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P11','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.
textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F;P1]; %Almacenamiento del espectro de frecuencias en variable.
hoja1='Espectro frecuencia P11'; %Posición donde colocar el espectro de frecuencia.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hoja1); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.
```

case 'P12' %Operaciones para warning sound P12.

```
filename='.\Tarea 1\Pasadaslargas\P12p12.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P12','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.
textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F;P1]; %Almacenamiento del espectro de frecuencias en variable.
hoja1='Espectro frecuencia P12'; %Posición donde colocar el espectro de frecuencia.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hoja1); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.
```

case 'P13' %Operaciones para warning sound P13.

```
filename='.\Tarea 1\Pasadaslargas\P13p13.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P13','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.
textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F;P1]; %Almacenamiento del espectro de frecuencias en variable.
hoja1='Espectro frecuencia P13'; %Posición donde colocar el espectro de frecuencia.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hoja1); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.
```

case 'P14' %Operaciones para warning sound P14.

```
filename='.\Tarea 1\Pasadaslargas\P14p14.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P14','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.
textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F;P1]; %Almacenamiento del espectro de frecuencias en variable.
hoja1='Espectro frecuencia P14'; %Posición donde colocar el espectro de frecuencia.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hoja1); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.
```

case 'P16' %Operaciones para warning sound P16.

```
filename='.\Tarea 1\Pasadaslargas\P16p16.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P16','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.
textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F;P1]; %Almacenamiento del espectro de frecuencias en variable.
hoja1='Espectro frecuencia P16'; %Posición donde colocar el espectro de frecuencia.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
xlswrite(filename,textbandf,hoja1); %Transferencia del encabezado de las columnas.
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.
```

case 'P17' %Operaciones para warning sound P17.

```
filename='.\Tarea 1\Pasadaslargas\P17p17.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla,filename,'Sheet','Datos Ensayo P17','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.
textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf=[F;P1]; %Almacenamiento del espectro de frecuencias en variable.
hoja1='Espectro frecuencia P17'; %Posición donde colocar el espectro de frecuencia.
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.
```

```
xlswrite(filename, textbandf, hoja1); % Transferencia del encabezado de las columnas.
xlswrite(filename, bandf, hoja1, xIRange); % Transferencia de los datos del espectro de frecuencias.

case 'P18' % Operaciones para warning sound P18.

filename = '\Tarea 1\Pasadaslargas\P18\p18.xlsx'; % Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla, filename, 'Sheet', 'Datos Ensayo P18', 'WriteVariableNames', true); % Transferencia de la tabla informativa a la hoja de cálculo.
textbandf = {'Banda Tercio de Octava', 'dBA'}; % Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1 = round(P1, 1); % Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf = [F; P1]; % Almacenamiento del espectro de frecuencias en variable.
hoja1 = 'Espectro frecuencia P18'; % Posición donde colocar el espectro de frecuencia.
xIRange = 'A2'; % Posición donde colocar el espectro de frecuencia.
xlswrite(filename, textbandf, hoja1); % Transferencia del encabezado de las columnas.
xlswrite(filename, bandf, hoja1, xIRange); % Transferencia de los datos del espectro de frecuencias.

case 'P19' % Operaciones para warning sound P19.

filename = '\Tarea 1\Pasadaslargas\P19\p19.xlsx'; % Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla, filename, 'Sheet', 'Datos Ensayo P19', 'WriteVariableNames', true); % Transferencia de la tabla informativa a la hoja de cálculo.
textbandf = {'Banda Tercio de Octava', 'dBA'}; % Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1 = round(P1, 1); % Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf = [F; P1]; % Almacenamiento del espectro de frecuencias en variable.
hoja1 = 'Espectro frecuencia P19'; % Posición donde colocar el espectro de frecuencia.
xIRange = 'A2'; % Posición donde colocar el espectro de frecuencia.
xlswrite(filename, textbandf, hoja1); % Transferencia del encabezado de las columnas.
xlswrite(filename, bandf, hoja1, xIRange); % Transferencia de los datos del espectro de frecuencias.

case 'P20' % Operaciones para warning sound P20.

filename = '\Tarea 1\Pasadaslargas\P20\p20.xlsx'; % Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla, filename, 'Sheet', 'Datos Ensayo P20', 'WriteVariableNames', true); % Transferencia de la tabla informativa a la hoja de cálculo.
textbandf = {'Banda Tercio de Octava', 'dBA'}; % Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1 = round(P1, 1); % Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf = [F; P1]; % Almacenamiento del espectro de frecuencias en variable.
hoja1 = 'Espectro frecuencia P20'; % Posición donde colocar el espectro de frecuencia.
xIRange = 'A2'; % Posición donde colocar el espectro de frecuencia.
xlswrite(filename, textbandf, hoja1); % Transferencia del encabezado de las columnas.
xlswrite(filename, bandf, hoja1, xIRange); % Transferencia de los datos del espectro de frecuencias.

end % Cierre de la estructura de control switch para códigos de Pasadas Cortas.
end % Cierre de la estructura de control switch para todas las velocidades.
```

## Programa vexcel

```
LISTADO VELOCIDADES.
switch v % Estructura de control switch que realiza las funciones que se definan para cada caso de velocidad para warning sound E1.

case 1 % Operaciones para velocidad de 5 km/h.

filename = '\E1\C1 E1 5\e1cinco.xlsx'; % Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla, filename, 'Sheet', 'Datos Ensayo 5km', 'WriteVariableNames', true); % Transferencia de la tabla informativa a la hoja de cálculo.

textbandf = {'Banda Tercio de Octava', 'dBA'}; % Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1 = round(P1, 1); % Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf = [F; P1]; % Almacenamiento del espectro de frecuencias en variable.
hoja1 = 'Espectro frecuencia e15uno'; % Establecimiento de nombre la hoja.
xIRange = 'A2'; % Posición donde colocar el espectro de frecuencia.
xlswrite(filename, textbandf, hoja1); % Transferencia del encabezado de las columnas.
xlswrite(filename, bandf, hoja1, xIRange); % Transferencia de los datos del espectro de frecuencias.

textbandf = {'Banda Tercio de Octava', 'dBA'}; % Designación del nombre a las columnas de los datos en la hoja de cálculo.
P2 = round(P2, 1); % Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf = [F; P2]; % Almacenamiento del espectro de frecuencias en variable.
hoja2 = 'Espectro frecuencia e15dos'; % Establecimiento de nombre la hoja.
xIRange = 'A2'; % Posición donde colocar el espectro de frecuencia.
xlswrite(filename, textbandf, hoja2); % Transferencia del encabezado de las columnas.
xlswrite(filename, bandf, hoja2, xIRange); % Transferencia de los datos del espectro de frecuencias.

case 2 % Operaciones para velocidad de 10 km/h.

filename = '\E1\C1 E1 10\e1diez.xlsx'; % Descripción de la ruta que dirige a la hoja de cálculo.
writetable(tabla, filename, 'Sheet', 'Datos Ensayo 10km', 'WriteVariableNames', true); % Transferencia de la tabla informativa a la hoja de cálculo.

textbandf = {'Banda Tercio de Octava', 'dBA'}; % Designación del nombre a las columnas de los datos en la hoja de cálculo.
P1 = round(P1, 1); % Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.
bandf = [F; P1]; % Almacenamiento del espectro de frecuencias en variable.
hoja1 = 'Espectro frecuencia e110uno'; % Establecimiento de nombre la hoja.
xIRange = 'A2'; % Posición donde colocar el espectro de frecuencia.
xlswrite(filename, textbandf, hoja1); % Transferencia del encabezado de las columnas.
```

```
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.
```

**case 3** %Operaciones para velocidad de 15 km/h.

```
filename='.\E1\CI E1 15\15e15quince.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.  
writetable(tabla,filename,'Sheet','Datos Ensayo 15km','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.
```

```
textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.  
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.  
bandf=[F;P1]; %Almacenamiento del espectro de frecuencias en variable.  
hoja1='Espectro frecuencia e115uno'; %Establecimiento de nombre la hoja.  
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.  
xlswrite(filename,textbandf,hoja1); %Transferencia del encabezado de las columnas.  
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.
```

```
textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.  
P2=round(P2,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.  
bandf=[F;P2]; %Almacenamiento del espectro de frecuencias en variable.  
hoja2='Espectro frecuencia e115dos'; %Establecimiento de nombre la hoja.  
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.  
xlswrite(filename,textbandf,hoja2); %Transferencia del encabezado de las columnas.  
xlswrite(filename,bandf,hoja2,xlRange); %Transferencia de los datos del espectro de frecuencias.
```

**case 4** %Operaciones para velocidad de 20 km/h.

```
filename='.\E1\CI E1 20\20e1veinte.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.  
writetable(tabla,filename,'Sheet','Datos Ensayo 20km','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.
```

```
textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.  
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.  
bandf=[F;P1]; %Almacenamiento del espectro de frecuencias en variable.  
hoja1='Espectro frecuencia e120uno'; %Establecimiento de nombre la hoja.  
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.  
xlswrite(filename,textbandf,hoja1); %Transferencia del encabezado de las columnas.  
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.
```

```
textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.  
P2=round(P2,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.  
bandf=[F;P2]; %Almacenamiento del espectro de frecuencias en variable.  
hoja2='Espectro frecuencia e120dos'; %Establecimiento de nombre la hoja.  
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.  
xlswrite(filename,textbandf,hoja2); %Transferencia del encabezado de las columnas.  
xlswrite(filename,bandf,hoja2,xlRange); %Transferencia de los datos del espectro de frecuencias.
```

**case 5** %Operaciones para velocidad de 25 km/h.

```
filename='.\E1\CI E1 25\25e1veinticinco.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.  
writetable(tabla,filename,'Sheet','Datos Ensayo 25km','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.
```

```
textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.  
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.  
bandf=[F;P1]; %Almacenamiento del espectro de frecuencias en variable.  
hoja1='Espectro frecuencia e125uno'; %Establecimiento de nombre la hoja.  
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.  
xlswrite(filename,textbandf,hoja1); %Transferencia del encabezado de las columnas.  
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.
```

```
textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.  
P2=round(P2,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.  
bandf=[F;P2]; %Almacenamiento del espectro de frecuencias en variable.  
hoja2='Espectro frecuencia e125dos'; %Establecimiento de nombre la hoja.  
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.  
xlswrite(filename,textbandf,hoja2); %Transferencia del encabezado de las columnas.  
xlswrite(filename,bandf,hoja2,xlRange); %Transferencia de los datos del espectro de frecuencias.
```

**case 6** %Operaciones para velocidad de 30 km/h.

```
filename='.\E1\CI E1 30\30e1treinta.xlsx'; %Descripción de la ruta que dirige a la hoja de cálculo.  
writetable(tabla,filename,'Sheet','Datos Ensayo 30km','WriteVariableNames',true); %Transferencia de la tabla informativa a la hoja de cálculo.
```

```
textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.  
P1=round(P1,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.  
bandf=[F;P1]; %Almacenamiento del espectro de frecuencias en variable.  
hoja1='Espectro frecuencia e130uno'; %Establecimiento de nombre la hoja.  
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.  
xlswrite(filename,textbandf,hoja1); %Transferencia del encabezado de las columnas.  
xlswrite(filename,bandf,hoja1,xlRange); %Transferencia de los datos del espectro de frecuencias.
```

```
textbandf={'Banda Tercio de Octava','dBA'}; %Designación del nombre a las columnas de los datos en la hoja de cálculo.  
P2=round(P2,1); %Redondeo a una cifra decimal del nivel de presión sonora de todas las bandas del espectro de frecuencias en tercios de octava.  
bandf=[F;P2]; %Almacenamiento del espectro de frecuencias en variable.  
hoja2='Espectro frecuencia e130dos'; %Establecimiento de nombre la hoja.  
xlRange = 'A2'; %Posición donde colocar el espectro de frecuencia.  
xlswrite(filename,textbandf,hoja2); %Transferencia del encabezado de las columnas.  
xlswrite(filename,bandf,hoja2,xlRange); %Transferencia de los datos del espectro de frecuencias.
```

end %Cierre de la estructura de control switch para las velocidades del warning sound E1.



## ANEXO II. PROGRAMAS PARA ANÁLISIS DIRECTIVIDAD VERTICAL

### Programa plottridimensional

```
CÁLCULO DE DIMENSIONES A EMPLEAR.
if (pas==1 & ws==9 & v==1 & con==1) %Primera condición donde se establece el caso donde no es necesario hacer arreglos a la matriz de presión sonora.

    tabppalutil=tabppal; %Almacenamiento de la parte útil de la matriz de presión sonora en variable.
    tam=size(tabppalutil); %Cálculo del tamaño de la matriz de presión sonora.

else %Segunda condición donde se establece el procedimiento para el resto de casos.

    tambruto=size(tabppal); %Cálculo del tamaño de la matriz de presión sonora.
    tamxbruto=tambruto(2); %Determinación dimensión x de la matriz de presión sonora.
    tamybruto=tambruto(1); %Determinación dimensión y de la matriz de presión sonora.
    vectutil=[2:1:tamybruto]; %Eliminación de la primera fila de la matriz de presión sonora,
    tabppalutil=tabppal(vectutil,:); %Almacenamiento de la parte útil de la matriz de presión sonora en variable.
    tam=size(tabppalutil); % Cálculo del tamaño de la matriz de presión sonora.

end %Cierre de condiciones.

tamx=tam(2); %Determinación del tamaño en dimensión X de la matriz de presión sonora.
tamy=tam(1); %Determinación del tamaño en dimensión Y de la matriz de presión sonora.

DETERMINACIÓN DE LOS VALORES DE LOS EJES.
if (pas==2 & ws==1) %Primera condición donde se designan los vectores X e Y a usar según el caso.

    X=[-1100:(2200/tamx):1099]; %Definición longitud y valores del eje X.
    Y=[800:(-1600/tamy):-799]; %Definición longitud y valores del eje Y.

else if (pas==2) %Segunda condición donde se designan los vectores X e Y a usar según el caso.

    X=[-1100:(2200/tamx):1099]; %Definición longitud y valores del eje X.
    Y=[900:(-1800/tamy):-899]; %Definición longitud y valores del eje Y.

else %Tercera condición donde se designan los vectores X e Y a usar según el caso.

    X=[-3000:(6000/tamx):2999]; %Definición longitud y valores del eje X.
    Y=[2200:(-4400/tamy):-2199]; %Definición longitud y valores del eje Y.

end %Cierre de la segunda condición.

end %Cierre de condiciones.

TRANSFORMACIÓN DE LOS VALORES DE LA MATRIZ DE PRESIÓN SONORA DE Pa a dB.
tabppalutil2=tabppalutil.^2; %Elevación de la matriz de presión sonora al cuadrado.
dbtabppalutil=10*log10(tabppalutil2/(20*10^(-6))^2); %Transformación de los valores de presión sonora de Pa a dB.

DETERMINACIÓN DEL MÁXIMO Y MÍNIMO NIVEL DE PRESIÓN SONORA.
dbmax=max(max(dbtabppalutil)); %Cálculo del valor de presión sonora en dB máximo.
dbmin=min(min(dbtabppalutil)); %Cálculo del valor de presión sonora en dB mínimo.

DESIGNACIÓN DEL PUNTO DE REFERENCIA.
if (ws==1 & v==1 && con==1) %Primera condición donde se designa el punto de referencia a emplear según el caso.

    Lppa=tabppalutil; %Almacenamiento de la parte útil de la matriz de presión sonora en variable.
    photox=2320; %Componente X del punto de referencia de la imagen.
    photoy=-700; %Componente Y del punto de referencia de la imagen.
    posx=find(photox<=X & photox+30>=X); %Búsqueda de la posición de dicho punto en el vector X.
    posy=find(photoy>=Y & photoy-40<=Y); %Búsqueda de la posición de dicho punto en el vector Y.
    cx=X(posx); %Valor del punto de referencia encontrado en el vector X.
    cy=Y(posy); %Valor del punto de referencia encontrado en el vector Y.

else if (pas==1 & (con==1 | con==3)) %Segunda condición donde se designa el punto de referencia a emplear según el caso.

    Lppa=tabppalutil; %Almacenamiento de la parte útil de la matriz de presión sonora en variable.
    photox=2320; %Componente X del punto de referencia de la imagen.
    photoy=-690; %Componente Y del punto de referencia de la imagen.
    posx=find(photox<=X & photox+20>=X); %Búsqueda de la posición de dicho punto en el vector X.
    posy=find(photoy>=Y & photoy-20<=Y); %Búsqueda de la posición de dicho punto en el vector Y.
    cx=X(posx); %Valor del punto de referencia encontrado en el vector X.
    cy=Y(posy); %Valor del punto de referencia encontrado en el vector Y.

else if (pas==1 & (con==2 | con==4)) %Tercera condición donde se designa el punto de referencia a emplear según el caso.
```



```

Lppa=tabppalutil; %Almacenamiento de la parte útil de la matriz de presión sonora en variable.
photox=-2320; %Componente X del punto de referencia de la imagen.
photoy=-690; %Componente Y del punto de referencia de la imagen.
posx=find(photox>=X & photox-20<=X); %Búsqueda de la posición de dicho punto en el vector X.
posy=find(photoy>=Y & photoy-20<=Y); %Búsqueda de la posición de dicho punto en el vector Y.
cx=X(posx); %Valor del punto de referencia encontrado en el vector X.
cy=Y(posy); %Valor del punto de referencia encontrado en el vector Y.

else if (pas==2 & ws==1) %Cuarta condición donde se designa el punto de referencia a emplear según el caso.

Lppa=tabppalutil; %Almacenamiento de la parte útil de la matriz de presión sonora en variable.
photox=960; %Componente X del punto de referencia de la imagen.
photoy=-600; %Componente Y del punto de referencia de la imagen.
posx=find(photox<=X & photox+10>=X); %Búsqueda de la posición de dicho punto en el vector X.
posy=find(photoy>=Y & photoy-10<=Y); %Búsqueda de la posición de dicho punto en el vector Y.
cx=X(posx); %Valor del punto de referencia encontrado en el vector X.
cy=Y(posy); %Valor del punto de referencia encontrado en el vector Y.

else if (pas==2 & mod(ws,2)~=0) %Quinta condición donde se designa el punto de referencia a emplear según el caso.

Lppa=tabppalutil; %Almacenamiento de la parte útil de la matriz de presión sonora en variable.
photox=960; %Componente X del punto de referencia de la imagen.
photoy=-600; %Componente Y del punto de referencia de la imagen.
posx=find(photox<=X & photox+10>=X); %Búsqueda de la posición de dicho punto en el vector X.
posy=find(photoy<=Y & photoy+10>=Y); %Búsqueda de la posición de dicho punto en el vector Y.
cx=X(posx); %Valor del punto de referencia encontrado en el vector X.
cy=Y(posy); %Valor del punto de referencia encontrado en el vector Y.

else %Sexta condición donde se designa el punto de referencia a emplear para el resto de casos.

Lppa=tabppalutil; %Almacenamiento de la parte útil de la matriz de presión sonora en variable.
photox=-960; %Componente X del punto de referencia de la imagen.
photoy=-600; %Componente Y del punto de referencia de la imagen.
posx=find(photox>=X & photox-10<=X); %Búsqueda de la posición de dicho punto en el vector X.
posy=find(photoy<=Y & photoy+10>=Y); %Búsqueda de la posición de dicho punto en el vector Y.
cx=X(posx); %Valor del punto de referencia encontrado en el vector X.
cy=Y(posy); %Valor del punto de referencia encontrado en el vector Y.

end %Cierre de la quinta condición.
end %Cierre de la cuarta condición.
end %Cierre de la tercera condición.
end %Cierre de la segunda condición.

end %Cierre de condiciones.

GRAFICACIÓN EN 3D.
figure %Representación del gráfico en una ventana independiente.

surf(X,Y,dbtabppalutil) %Representación en 3D de la matriz de los datos.
shading flat %Cambio en la definición en la representación de la gráfica 3D
colormap(hsv); %Cambio en el espectro de colores en la representación del gráfico 3D.

title('PRESIÓN SONORA (dB)') %Colocación de título para el gráfico 3D.
xlabel('Eje X (mm)') %Colocación de etiqueta del eje X para el gráfico 3D.
ylabel('Eje Y (mm)') %Colocación de etiqueta del eje Y para el gráfico 3D.
xlim([X(1) X(length(X))]) %Definición de los límites del eje X para el gráfico 3D.
ylim([Y(length(Y)) Y(1)]) %Definición de los límites del eje Y para el gráfico 3D.

c=colorbar; %Aplicación de barra de definición de valores numéricos para el espectro de colores.
c.Limits = [dbmin dbmax]; %Definición de los valores máximos y mínimos de dicha barra.
caxis([dbmin dbmax]); %Definición de los límites de dicha barra.
c.Label.String = 'Presión Sonora en dB'; %Colocación de etiqueta para barra.

REPRESENTACIÓN DEL PUNTO DE ESTUDIO.
if (pas==1) %Primera condición donde se grafica el punto de referencia empleado según el caso.

puntestX=num2str(cx,4); %Transformación de datos numéricos a array de caracteres del elemento en X del punto de referencia.
puntestY=num2str(cy,5); %Transformación de datos numéricos a array de caracteres del elemento en Y del punto de referencia.
txt1=['\bf X = ',puntestX,' mm']; %Definición del texto para marcador de posición X.
texto1=text(cx+60,cy+170,dbmax,txt1); %Designación de la posición del texto para marcador de posición X.
txt2=['\bf Y = ',puntestY,' mm']; %Definición del texto para marcador de posición Y.
texto2=text(cx+60,cy+70,dbmax,txt2); %Designación de la posición del texto para marcador de posición Y.
texto3=text(cx,cy,dbmax,'\bf \bullet'); %Colocación de símbolo cruz contenida en círculo que marca el punto de estudio.
texto3.FontSize=15; %Aumento del tamaño del símbolo cruz contenida en círculo.

else %Segunda condición donde se grafica el punto de referencia empleado para el resto de casos.

puntestX=num2str(cx,4); %Transformación de datos numéricos a array de caracteres el elemento en X del punto de referencia.
puntestY=num2str(cy,5); %Transformación de datos numéricos a array de caracteres el elemento en Y del punto de referencia.
txt1=['\bf X = ',puntestX,' mm']; %Definición del texto para marcador de posición X.

```

```
texto1=text(cx-100,cy+80,dbmax,txt1); %Designación de la posición del texto para marcador de posición X.
txt2=['\bf Y = ',puntestY,' mm']; %Definición del texto para marcador de posición Y.
texto2=text(cx-100,cy+30,dbmax,txt2); %Designación de la posición del texto para marcador de posición Y.
texto3=text(cx,cy,dbmax,'\bf \bullet'); %Colocación de símbolo de cruz contenida en círculo que marca el punto de estudio.
texto3.FontSize=15; %Aumento del tamaño del símbolo cruz contenida en círculo.

end %Cierre de condiciones.

if (pas==1 & (con==1 | con==3)) %Primera condición donde se colorean los textos definidos anteriormente según el caso.

    texto1.Color='White'; %Definición de color del marcador de posición X.
    texto2.Color='White'; %Definición de color del marcador de posición Y.
    texto3.Color='White'; %Definición de color del símbolo de cruz contenida en círculo.

else if (pas==1 & (con==2 | con==4)) %Segunda condición donde se colorean los textos definidos anteriormente según el caso.

    texto1.Color='Black'; %Definición de color del marcador de posición X.
    texto2.Color='Black'; %Definición de color del marcador de posición Y.
    texto3.Color='Black'; %Definición de color del símbolo de cruz contenida en círculo.

else if (pas==2 & mod(ws,2)~=0) %Tercera condición donde se colorean los textos definidos anteriormente según el caso.

    texto1.Color='White'; %Definición de color del marcador de posición X.
    texto2.Color='White'; %Definición de color del marcador de posición Y.
    texto3.Color='White'; %Definición de color del símbolo de cruz contenida en círculo.

else %Cuarta condición donde se colorean los textos definidos anteriormente para el resto de casos.

    texto1.Color='Black'; %Definición de color del marcador de posición X
    texto2.Color='Black'; %Definición de color del marcador de posición Y.
    texto3.Color='Black'; %Definición de color del símbolo de cruz contenida en círculo.

    end %Cierre de la tercera condición.

    end %Cierre de la segunda condición.

end %Cierre de condiciones.
```

### Programa directividadizq

```
DETERMINACIÓN DE NUEVO SISTEMA DE REFERENCIA.
Xm=(X-cx)/1000; %Cambio de los valores del eje X al nuevo sistema de referencia.
Ym=(Y-cy)/1000; %Cambio de los valores del eje y al nuevo sistema de referencia.

CÁLCULO DEL RADIO QUE SE GENERA DESDE EL ORIGEN DE NUESTRO SISTEMA DE REFERENCIA A CADA PUNTO.
for i=1:tamx %Primer bucle para recorrer el vector X nuevo de izquierda a derecha y crear nuevas variables.

    for j=1:tamy %Segundo bucle para recorrer el vector Y nuevo de arriba a abajo y crear nuevas variables.

        if Xm(i)==0 %Primera condición donde se establece el caso en el que el vector X nuevo sea 0.

            hipotenusa(j,i)=abs(Ym(j)); %Asignación de hipotenusa como valor absoluto del punto del vector Y nuevo.

        else if Ym(j)==0 %Segunda condición donde se establece el caso en el que el vector Y nuevo sea 0.

            hipotenusa(j,i)=abs(Xm(i)); %Asignación de hipotenusa el valor absoluto del punto del vector X nuevo.

        else %Tercera condición donde se establece el procedimiento para el resto de casos.

            hipotenusa(j,i)=sqrt(((Xm(i)).^2)+((Ym(j)).^2)); %Cálculo de hipotenusa donde catetos son valores de eje X e Y nuevos.

            end %Cierre de la segunda condición.

        end %Cierre de condiciones.

    end %Cierre del segundo bucle.

end %Cierre del primer bucle.

CÁLCULO DE RADIO A EMPLEAR.
radio=Ym(1); %Determinación del radio a emplear para tomar datos.

DETERMINACIÓN DE LÍMITES DE SECTOR A UTILIZAR.
if (pas==1) %Primera condición donde se establecen las directrices a seguir para los casos de Pasadas Cortas.

    if (v==1 & ws==1 & con==1) %Primera condición correspondiente al primer caso particular dentro de Pasadas Cortas para establecer los límites a estudiar.

        photoextx=0; %Definición de la coordenada X del punto límite a estudiar de la imagen del caso a analizar.
        photoexty=-1410; %Definición de la coordenada Y del punto límite a estudiar de la imagen del caso a analizar.
```

```

posextx=find(photoextx<=X & photoextx+30>=X); %Búsqueda de la posición de la coordenada X límite dentro del vector eje X.
posexy=find(photoexty>=Y & photoexty-10<=Y); %Búsqueda de la posición de la coordenada Y límite dentro del vector eje Y.
nx=X(posextx); % Valor de la coordenada X del punto límite a utilizar del vector eje X.
ny=Y(posexy); % Valor de la coordenada Y del punto límite a utilizar del vector eje Y.
disty=abs(ny-cy); %Distancia resultante de la resta entre el punto límite a usar y el punto de referencia encontrado para componente Y.
angext=asind(disty/(radio*1000)); % Ángulo extra a analizar.
posfinal=posexy; %Búsqueda de la posición en el vector Y de esa distancia extra.

else %Segunda condición donde se establecen los límites para el resto de casos de Pasadas Cortas.

    photoextx=0; %Definición de la coordenada X del punto límite a estudiar de la imagen del caso a analizar.
    photoexty=-1420; %Definición de la coordenada Y del punto límite a estudiar de la imagen del caso a analizar.
    posextx=find(photoextx<=X & photoextx+10>=X); %Búsqueda de la posición de la coordenada X límite dentro del vector eje X.
    posexy=find(photoexty<=Y & photoexty+10>=Y); %Búsqueda de la posición de la coordenada Y límite dentro del vector eje Y.
    nx=X(posextx); % Valor de la coordenada X del punto límite a utilizar del vector eje X.
    ny=Y(posexy); % Valor de la coordenada Y del punto límite a utilizar del vector eje Y.
    disty=abs(ny-cy); %Distancia resultante de la resta entre el punto límite a usar y el punto de referencia encontrado para componente Y.
    angext=asind(disty/radio); % Ángulo extra a analizar.
    posfinal=posexy; %Búsqueda de la posición en el vector Y de esa distancia extra.

end %Cierre de condiciones para los establecer los límites en los distintos casos de Pasadas Cortas.

else %Segunda condición donde se establecen las directrices a seguir para los casos de Pasadas Largas.

    posfinal=tamy; %Definición de la posición en el vector Y de esa distancia extra como la posición máxima del vector Y.

end %Cierre de condiciones para las distintas pasadas.

DETERMINACIÓN DE PUNTOS ÚTILES DE LA MATRIZ DE PRESIÓN SONORA.
k=1; %Definición de contador para definir nuevas variables.
for i=1:tamy %Primer bucle para recorrer el eje Y de la matriz de hipotenusas de arriba a abajo, crear y modificar variables.

    for j=1:tamx %Segundo bucle para recorrer el eje X de la matriz de hipotenusas de izquierda a derecha, crear y modificar variables.

        if (hipotenusa(i,j)>=radio+0.03 | hipotenusa(i,j)<=radio-0.03) | i>posfinal %Primera condición donde se establecen puntos inútiles de la matriz de presión sonora en Pa.

            Lppa(i,j)=0; %Asignación de aquellos valores que cumplan la condición anterior del valor 0.

        else %Segunda condición donde se establecen qué puntos son útiles de la matriz de presión sonora en Pa.

            vectapoymed(k)=Lppa(i,j); %Almacenamiento del valor del punto útil en variable.
            k=k+1; %Aumento del contador en una unidad.

        end %Cierre de condiciones.

    end %Cierre del segundo bucle.

end %Cierre del primer bucle.

CÁLCULO DEL ÍNDICE DE DIRECTIVIDAD Y TRANSFORMACIÓN DE LA MATRIZ DE PRESIÓN SONORA DE Pa A dB.
Lppaomni=mean(vectapoymed); %Cálculo de la media de la variable donde se almacenan los valores de los puntos útiles de presión sonora en Pa.
Q=(Lppa.^2)/(Lppaomni.^2); %Cálculo del factor de directividad.
DI=10*log10(Q); %Cálculo del índice de directividad.
LpdB=10*log10(Lppa.^2/(20*10^(-6))^2); %Transformación de la matriz de presión sonora de Pa a dB.

CÁLCULO, ARREGLO Y DEFINICIÓN DE PARÁMETROS PARA GRAFICAR.
z=1; %Definición de contador para definir nuevas variables.
posyinter=round(tamy/2); %Cálculo del punto hasta el cual realizar pasadas de izquierda a derecha.

for i=tamy:-1:posyinter %Primer bucle para recorrer el eje Y de la matriz de presión sonora en Pa de abajo a arriba, crear y modificar variables.

    for j=1:tamx %Segundo bucle para recorrer el eje X de la matriz de presión sonora en Pa de izquierda a derecha, crear y modificar variables.

        if Lppa(i,j)~=0 %Primera condición donde se establecen las directrices para puntos de la matriz de presión sonora en Pa distintos a 0.

            copuesto(z)=abs(Ym(i)); %Cálculo del cateto opuesto para puntos útiles.
            ccontiguo(z)=abs(Xm(j)); %Cálculo del cateto contiguo para puntos útiles.

            if i<=tamy & i>=posy %Primera condición para cálculo correcto de ángulo para graficar.

                theta(z)=0-atand(copuesto(z)/ccontiguo(z)); %Directriz para casos con ángulos menores o iguales a 0°.

            else %Segunda condición para cálculo correcto de ángulo para graficar.

                theta(z)=atand(copuesto(z)/ccontiguo(z)); %Directriz para casos con ángulos mayores de 0°.

            end %Cierre de condiciones.

            rho(z)=DI(i,j); %Traspasso de valores de índice de directividad de matriz a vector de 1 fila.
            rhodB(z)=LpdB(i,j); %Traspasso de valores de presión sonora en dB a vector de 1 fila.
            DI(i,j)=0; %Asignación del valor 0 para el valor que ha sido tomado.
            z=z+1; %Aumento de contador en una unidad.

        end

    end

end

```

```

end %Cierre de condiciones.

end %Cierre del segundo bucle.

end %Cierre del primer bucle.

for j=1:tamx %Primer bucle para recorrer el eje X de la matriz de presión sonora en Pa de izquierda a derecha, crear y modificar variables.

for i=(posyinter-1):-1:1 %Segundo bucle para recorrer el eje Y de la matriz de presión sonora en Pa de abajo a arriba, crear y modificar variables.

if Lppa(i,j)~=0 %Primera condición donde se establecen las directrices para puntos de la matriz de presión sonora en Pa distintos a 0.

    copuesto(z)=abs(Ym(i)); %Cálculo del cateto opuesto para puntos útiles.
    ccontiguo(z)=abs(Xm(j)); %Cálculo del cateto contiguo para puntos útiles.

    if j<=posx %Primera condición para cálculo correcto de ángulo para graficar.

        theta(z)=atand(copuesto(z)/ccontiguo(z)); %Directriz para casos con ángulos mayores de 0° y menores a 90°.

    else %Segunda condición para cálculo correcto de ángulo para graficar.

        theta(z)=(90-atand(copuesto(z)/ccontiguo(z)))+90; %Directriz para casos con ángulos mayores de 90°.

    end %Cierre de condiciones.

    rho(z)=DI(i,j); %Traspaso de valores de índice de directividad de matriz a vector de 1 fila.
    rhodB(z)=LpdB(i,j); %Traspaso de valores de presión sonora en dB a vector de 1 fila.
    z=z+1; %Aumento de contador en una unidad.

end %Cierre de condiciones.

end %Cierre del segundo bucle.

end %Cierre del primer bucle.

thetar=deg2rad(theta); %Transformación de la variable de grados a radianes.

rhomax=max(rho); %Designación del máximo valor del índice de directividad.
rhodBmax=max(rhodB); %Designación del máximo valor de presión sonora en dB.
rhodBmax=ceil(rhodBmax); %Redondeo a 1 cifra decimal del valor máximo de presión sonora en dB.

rhomin=min(rho); %Designación del mínimo valor del índice de directividad.
rhodBmin=min(rhodB); %Designación del mínimo valor de presión sonora en dB.
rhodBminr=fix(rhodBmin); %Truncamiento del valor máximo de presión sonora en dB.

REPRESENTACIÓN DE DIRECTIVIDAD.
%GRÁFICO EN COORDENADAS CARTESIANAS DEL ÍNDICE DE DIRECTIVIDAD.
figure %Representación del gráfico en una ventana independiente.
plot(thetar,rho,'Color',(1/255)*[119,136,153],'LineWidth',3) %Representación del índice de directividad en coordenadas cartesianas.
hold on %Mantenimiento del gráfico anterior.
plot(thetar,rho,'s','MarkerEdge',(1/255)*[128,128,128],'MarkerFace',(1/255)*[240,255,255],'MarkerSize',7) %Representación de marcadores en cada punto.
hold off %Eliminación del mantenimiento de gráficos.
xlim([thetar(1) thetar(length(thetar))]) %Definición del extremo del eje X del gráfico en coordenadas cartesianas.
title('DIRECTIVIDAD','FontSize',15) %Asignación de título.
ylabel('Índice de Directividad (dB)') %Asignación de etiqueta para eje Y.
xlabel('Dirección (rad)') %Asignación de etiqueta para eje X.
grid minor %Aplicación de cuadrícula de tamaño pequeño.

%GRÁFICO EN COORDENADAS POLARES DEL ÍNDICE DE DIRECTIVIDAD.
figure %Representación del gráfico en una ventana independiente.
polarplot(thetar,rho,'Color',(1/255)*[188,143,143],'LineWidth',2) %Representación del índice de directividad en coordenadas polares.
hold on %Mantenimiento del gráfico anterior.
polarplot(thetar,rho,'o','MarkerEdge','k','MarkerFace',(1/255)*[255,255,224],'MarkerSize',4) %Representación de marcadores en cada punto.
hold off %Eliminación del mantenimiento de gráficos.
rlim([rhomin-0.5 rhomax+0.12]) %Asignación de los radios extremos del gráfico en coordenadas polares.
thetalim([theta(1) theta(length(theta))]) %Asignación de los ángulos extremos del gráfico en coordenadas polares.
graf=gca; %Definición de objeto gráfico.
graf.ThetaDir='clockwise'; %Disposición de los ángulos en sentido antihorario.
graf.ThetaZeroLocation='left'; %Posición del número 0 en el gráfico polar.
title('DIRECTIVIDAD (DI)','FontSize',15) %Asignación de título.
rticklabels({'DI = -0.5','DI = 0','DI = 0.5','DI = 1'}) %Definición de las etiquetas de los resultados de los resultados de índice de directividad obtenidos.
rtickangle(graf,30) %Giro de las etiquetas de la línea anterior 30°.
grid minor %Aplicación de cuadrícula de tamaño pequeño.

%GRÁFICO EN COORDENADAS POLARES DEL NIVEL DE PRESIÓN SONORA EN dB.
figure %Representación del gráfico en una ventana independiente.
polarplot(thetar,rhodB,'Color',(1/255)*[75,0,130],'LineWidth',2) %Representación del nivel de presión sonora en dB en coordenadas polares.
hold on %Mantenimiento del gráfico anterior.

```

```
polarplot(thetar,rhodB,'o','MarkerEdge','k','MarkerFace',(1/255)*[255,240,245],'MarkerSize',4) %Representación de marcadores en cada punto.
hold off %Eliminación del mantenimiento de gráficos.
rlim((rhodBmin rhodBmax)) %Asignación de los radios extremos del gráfico en coordenadas polares.
thetalim((theta(1) theta(length(theta)))) %Asignación de los ángulos extremos del gráfico en coordenadas polares.
graf=gca; %Definición de objeto gráfico.
graf.ThetaDir='clockwise'; %Disposición de los ángulos en sentido antihorario.
graf.ThetaZeroLocation='left'; %Posición del número 0 en el gráfico polar.
title('DIRECTIVIDAD (dB)','FontSize',15) %Asignación de título.
rticklabels({'Lp = 58 dB','Lp = 58.5 dB','Lp = 59 dB','Lp = 59.5 dB','Lp = 64 dB','Lp = 59.8 dB','Lp = 60 dB'}) %Definición etiquetas de resultados en dB obtenidos.
rtickangle(graf,30) %Giro de las etiquetas de la línea anterior 30°.
grid minor %Aplicación de cuadrícula de tamaño pequeño.
```

### Programa directividaddcha

```
DETERMINACIÓN DE NUEVO SISTEMA DE REFERENCIA.
Xm=(X-cx)/1000; %Cambio de los valores del eje X al nuevo sistema de referencia.
Ym=(Y-cy)/1000; %Cambio de los valores del eje y al nuevo sistema de referencia.

CÁLCULO DEL RADIO QUE SE GENERA DESDE EL ORIGEN DE NUESTRO SISTEMA DE REFERENCIA A CADA PUNTO.
for i=1:tamx %Primer bucle para recorrer el vector X con nuevo sistema de referencia de izquierda a derecha.

    for j=1:tamy %Segundo bucle para recorrer el vector Y con nuevo sistema de referencia de arriba a abajo.

        if Xm(i)==0 %Primera condición donde se establece el caso en el que el vector X referido al nuevo sistema sea 0.

            hipotenusa(j,i)=abs(Ym(j)); %Para el caso anterior, se toma como hipotenusa el valor absoluto del punto correspondiente del vector Y referido al nuevo sistema.

            else if Ym(j)==0 %Segunda condición donde se establece el caso en el que el vector Y referido al nuevo sistema sea 0.

                hipotenusa(j,i)=abs(Xm(i)); %Para el caso anterior, se toma como hipotenusa el valor absoluto del punto correspondiente del vector X referido al nuevo sistema.

                else %Tercera condición donde se establece el procedimiento para el resto de casos.

                    hipotenusa(j,i)=sqrt(((Xm(i)).^2)+(Ym(j)).^2)); %Para el caso anterior, se calcula la hipotenusa tomando como catetos los valores del eje X e Y referidos al nuevo sistema.

                    end %Cierre de la segunda condición.

                end %Cierre de la primera condición.

            end %Cierre del segundo bucle.

        end %Cierre del primer bucle.

CÁLCULO DE RADIO A EMPLEAR.
radio=Ym(1); %Determinación del radio a emplear para tomar datos.

DETERMINACIÓN DE LÍMITES DE SECTOR A UTILIZAR.
if (pas==1) %Primera condición donde se establecen las directrices a seguir para los casos de Pasadas Cortas.

    photoextx=0; %Definición de la coordenada X del punto límite a estudiar de la imagen del caso a analizar.
    photoexty=-1420; %Definición de la coordenada Y del punto límite a estudiar de la imagen del caso a analizar.
    posextx=find(photoextx<=X & photoextx+10>=X); %Búsqueda de la posición de la coordenada X límite dentro del vector eje X.
    posextx=find(photoextx<=Y & photoextx+10>=Y); %Búsqueda de la posición de la coordenada Y límite dentro del vector eje Y.
    nx=X(posextx); %Valor de la coordenada X del punto límite a utilizar del vector eje X.
    ny=Y(posextx); %Valor de la coordenada Y del punto límite a utilizar del vector eje Y.
    disty=abs(ny-cy); %Distancia resultante de la resta entre el punto límite a utilizar y el punto de referencia encontrado para componente Y.
    angext=asind(disty/(radio*1000)); %Ángulo extra a analizar.
    posfinal=posextx; %Búsqueda de la posición en el vector Y de esa distancia extra.

    else %Segunda condición donde se establecen las directrices a seguir para los casos de Pasadas Largas.

        posfinal=tamy; %Definición de la posición en el vector Y de esa distancia extra como la posición máxima del vector Y.

    end %Cierre de condiciones para las distintas pasadas.

DETERMINACIÓN DE PUNTOS ÚTILES DE LA MATRIZ DE PRESIÓN SONORA.
k=1; %Definición de contador para definir nuevas variables.
for i=1:tamy %Primer bucle para recorrer el eje Y de la matriz de hipotenusas de arriba a abajo, crear y modificar variables.

    for j=1:tamx %Segundo bucle para recorrer el eje X de la matriz de hipotenusas de izquierda a derecha, crear y modificar variables.

        if (hipotenusa(i,j)>=radio+0.03 | hipotenusa(i,j)<=radio-0.03) | i>posfinal

            Lppa(i,j)=0; %Asignación de aquellos valores que cumplan la condición anterior del valor 0.

            else %Segunda condición donde se establecen qué puntos son útiles de la matriz de presión sonora en Pa.
```

```

    vectapoymed(k)=Lppa(i,j); %Almacenamiento del valor del punto útil en variable.
    k=k+1; % Aumento del contador en una unidad.

end %Cierre de condiciones.

end %Cierre del segundo bucle.

end %Cierre del primer bucle.

CÁLCULO DEL ÍNDICE DE DIRECTIVIDAD Y TRANSFORMACIÓN DE LA MATRIZ DE PRESIÓN SONORA DE Pa A dB.
Lppaomni=mean(vectapoymed); %Cálculo de la media de la variable donde se almacenan los valores de los puntos útiles de presión sonora en Pa.
Q=(Lppa.^2)/(Lppaomni.^2); %Cálculo del factor de directividad.
DI=10*log10(Q); %Cálculo del índice de directividad.
LpdB=10*log10(Lppa.^2/(20*10^(-6))^2); %Transformación de la matriz de presión sonora de Pa a dB.

CÁLCULO, ARREGLO Y DEFINICIÓN DE PARÁMETROS PARA GRAFICAR.
z=1; %Definición de contador para definir nuevas variables.
posyinter=round(tamy/2); %Cálculo del punto hasta el cual realizar pasadas de izquierda a derecha.

for i=tamy:-1:posyinter %Primer bucle para recorrer el eje Y de la matriz de presión sonora en Pa de abajo a arriba, crear y modificar variables.

for j=tamx:-1:1 %Segundo bucle para recorrer el eje X de la matriz de presión sonora en Pa de derecha a izquierda, crear y modificar variables.

if Lppa(i,j)~=0 %Primera condición donde se establecen las directrices para puntos de la matriz de presión sonora en Pa distintos a 0.

    copuesto(z)=abs(Ym(i)); %Cálculo del cateto opuesto para puntos útiles.
    ccontiguo(z)=abs(Xm(j)); %Cálculo del cateto contiguo para puntos útiles.

    if i<=tamy & i>=posy %Primera condición para cálculo correcto de ángulo para graficar.

        theta(z)=0-atand(copuesto(z)/ccontiguo(z)); %Directriz para casos con ángulos menores o iguales a 0°.

    else %Segunda condición para cálculo correcto de ángulo para graficar.

        theta(z)=atand(copuesto(z)/ccontiguo(z)); %Directriz para casos con ángulos mayores de 0°.

    end %Cierre de condiciones.

    rho(z)=DI(i,j); %Traspaso de valores de índice de directividad de matriz a vector de 1 fila.
    rhodB(z)=LpdB(i,j); %Traspaso de valores de presión sonora en dB a vector de 1 fila.
    DI(i,j)=0; %Asignación del valor 0 para el valor que ha sido tomado.
    z=z+1; %Aumento de contador en una unidad.

end %Cierre de condiciones.

end %Cierre de condiciones.

end %Cierre del primer bucle.

for j=tamx:-1:1 %Primer bucle para recorrer el eje X de la matriz de presión sonora en Pa de derecha a izquierda, crear y modificar variables.

for i=(posyinter-1):-1:1 %Segundo bucle para recorrer el eje Y de la matriz de presión sonora en Pa de abajo a arriba, crear y modificar variables.

if Lppa(i,j)~=0 %Primera condición donde se establecen las directrices para puntos de la matriz de presión sonora en Pa distintos a 0.

    copuesto(z)=abs(Ym(i)); %Cálculo del cateto opuesto para puntos útiles.
    ccontiguo(z)=abs(Xm(j)); %Cálculo del cateto contiguo para puntos útiles.

    if j<=posx %Primera condición para cálculo correcto de ángulo para graficar.

        theta(z)=(90-atand(copuesto(z)/ccontiguo(z))+90); %Directriz para casos con ángulos mayores de 0° y menores a 90°.

    else %Segunda condición para cálculo correcto de ángulo para graficar.

        theta(z)=atand(copuesto(z)/ccontiguo(z)); %Directriz para casos con ángulos mayores de 90°.

    end %Cierre de condiciones.

    rho(z)=DI(i,j); %Traspaso de valores de índice de directividad de matriz a vector de 1 fila.
    rhodB(z)=LpdB(i,j); %Traspaso de valores de presión sonora en dB a vector de 1 fila.
    z=z+1; %Aumento de contador en una unidad.

end %Cierre de condiciones.

end %Cierre del segundo bucle.

end %Cierre del primer bucle.

DEFINICIÓN DE VARIABLES PARA GRAFICAR.
thetar=deg2rad(theta); %Transformación de la variable de grados a radianes.

rhomax=max(rho); %Designación del máximo valor del índice de directividad.
rhodBmax=max(rhodB); %Designación del máximo valor de presión sonora en dB.

```

```

rhodBmaxr=ceil(rhodBmax); %Redondeo a 1 cifra decimal del valor máximo de presión sonora en dB.

rhomin=min(rho); %Designación del mínimo valor del índice de directividad.
rhodBmin=min(rhodB); %Designación del mínimo valor de presión sonora en dB.
rhodBminr=fix(rhodBmin); %Truncamiento del valor máximo de presión sonora en dB.

REPRESENTACIÓN DE DIRECTIVIDAD.
%GRÁFICO EN COORDENADAS CARTESIANAS DEL ÍNDICE DE DIRECTIVIDAD.
figure %Representación del gráfico en una ventana independiente.
plot(thetar,rho,'Color',(1/255)*[119,136,153],'LineWidth',3) %Representación del índice de directividad en coordenadas cartesianas.
hold on %Mantenimiento del gráfico anterior.
plot(thetar,rho,'s','MarkerEdge',(1/255)*[128,128,128],'MarkerFace',(1/255)*[240,255,255],'MarkerSize',7) %Graficación de marcadores en cada punto.
hold off %Eliminación del mantenimiento de gráficos.
xlim([thetar(1) thetar(length(thetar))]) %Definición del extremo del eje X del gráfico en coordenadas cartesianas.
title('DIRECTIVIDAD','FontSize',15) %Asignación de título.
ylabel('Índice de Directividad (DI)') %Asignación de etiqueta para eje Y.
xlabel('Dirección (rad)') %Asignación de etiqueta para eje X.
grid minor %Aplicación de cuadrícula de tamaño pequeño.

%GRÁFICO EN COORDENADAS POLARES DEL ÍNDICE DE DIRECTIVIDAD.
figure %Representación del gráfico en una ventana independiente.
polarplot(thetar,rho,'Color',(1/255)*[188,143,143],'LineWidth',2) %Representación del índice de directividad en coordenadas polares.
hold on %Mantenimiento del gráfico anterior.
polarplot(thetar,rho,'o','MarkerEdge','k','MarkerFace',(1/255)*[255,255,224],'MarkerSize',4) %Graficación de marcadores en cada punto.
hold off %Eliminación del mantenimiento de gráficos.
rlim([rhomin-0.4 rhomax+0.07]) %Asignación de los radios extremos del gráfico en coordenadas polares.
thetalim([theta(1) theta(length(theta))]) %Asignación de los ángulos extremos del gráfico en coordenadas polares.
graf=gca; %Definición de objeto gráfico.
title('DIRECTIVIDAD (DI)','FontSize',15) %Asignación de título.
rticklabels({'DI = -0.2','DI = 0','DI = 0.2'}) %Definición de las etiquetas de los resultados de índice de directividad obtenidos.
rtickangle(graf,30) %Giro de las etiquetas de la línea anterior 30°.
grid minor %Aplicación de cuadrícula de tamaño pequeño.

%GRÁFICO EN COORDENADAS POLARES DEL NIVEL DE PRESIÓN SONORA EN dB.
figure %Representación del gráfico en una ventana independiente.
polarplot(thetar,rhodB,'Color',(1/255)*[75,0,130],'LineWidth',2) %Representación del nivel de presión sonora en dB en coordenadas polares.
hold on %Mantenimiento del gráfico anterior.
polarplot(thetar,rhodB,'o','MarkerEdge','k','MarkerFace',(1/255)*[255,240,245],'MarkerSize',4) %Graficación de marcadores en cada punto.
hold off %Eliminación del mantenimiento de gráficos.
rlim([rhodBminr rhodBmax]) %Asignación de los radios extremos del gráfico en coordenadas polares.
thetalim([theta(1) theta(length(theta))]) %Asignación de los ángulos extremos del gráfico en coordenadas polares.
graf=gca; %Definición de objeto gráfico.
title('DIRECTIVIDAD (dB)','FontSize',15) %Asignación de título.
rticklabels({'Lp = 58 dB','Lp = 58.5 dB','Lp = 59 dB','Lp = 59.5 dB','Lp = 60 dB'}) %Definición de las etiquetas de los resultados de presión sonora en dB obtenidos.
rtickangle(graf,30) %Giro de las etiquetas de la línea anterior 30°.
grid minor %Aplicación de cuadrícula de tamaño pequeño.

```

### Programa selectorpasadas

```

clear %Limpieza del espacio de trabajo.
clc %Limpieza de variables.

PETICIÓN WARNING SOUND PASADAS CORTAS O LARGAS.
display('1-Pasadas Cortas') %Escribir por pantalla significado del código 1.
display('2-Pasadas Largas') %Escribir por pantalla significado del código 2.

pas=input('¿Qué tipo de warning sound quiere analizar? '); %Pregunta por pantalla el tipo de warning sound a analizar y lo almacena.

switch pas %Estructura de control switch que realiza las funciones que se definan para cada clase de warning sound según sean pasadas largas o cortas.

case 1 %Operaciones para opción Pasadas Cortas.

PETICIÓN DE WARNING SOUND PASADAS CORTAS A EJECUTAR.
display('1-Warning sound E1') %Mostrar por pantalla que código es el warning sound E1.
display('2-Warning sound E2') %Mostrar por pantalla que código es el warning sound E2.
display('3-Warning sound E3') %Mostrar por pantalla que código es el warning sound E3.
display('4-Warning sound E4') %Mostrar por pantalla que código es el warning sound E4.
display('5-Warning sound E5') %Mostrar por pantalla que código es el warning sound E5.
display('6-Warning sound E6') %Mostrar por pantalla que código es el warning sound E6.
display('7-Warning sound C1') %Mostrar por pantalla que código es el warning sound C1.
display('8-Warning sound C1 E') %Mostrar por pantalla que código es el warning sound C1 E.
display('9-Warning sound C1 WS') %Mostrar por pantalla que código es el warning sound C1 WS.
display('10-Warning sound C1 E2') %Mostrar por pantalla que código es el warning sound C1 E2.

ws=input('¿Qué warning sound quiere estudiar? '); %Pregunta por pantalla el warning sound a analizar y lo almacena.

LISTADO PASADAS CORTAS.
switch ws %Estructura de control switch que realiza las funciones que se definan para cada warning sound de Pasadas Cortas.

```

```
case 1 %Operaciones para warning sound E1.

run(../Tarea 2/E1\distribeuno.m') %Ejecución del script con nombre "distribeuno".

case 2 %Operaciones para warning sound E2.

run(../Tarea 2/E2\distribedos.m') %Ejecución del script con nombre "distribedos".

case 3 %Operaciones para warning sound E3.

run(../Tarea 2/E3\distribetres.m') %Ejecución del script con nombre "distribetres".

case 4 %Operaciones para warning sound E4.

run(../Tarea 2/E4\distribecuatro.m') %Ejecución del script con nombre "distribecuatro".

case 5 %Operaciones para warning sound E5.

run(../Tarea 2/E5\distribecinco.m') %Ejecución del script con nombre "distribecinco".

case 6 %Operaciones para warning sound E6.

run(../Tarea 2/E6\distribeseis.m') %Ejecución del script con nombre "distribeseis".

case 7 %Operaciones para warning sound C1.

run(../Tarea 2/C1\distribcuno.m') %Ejecución del script con nombre "distribcuno".

case 8 %Operaciones para warning sound C1 E.

run(../Tarea 2/C1 E\distribcunoe.m') %Ejecución del script con nombre "distribcunoe".

case 9 %Operaciones para warning sound C1 WS.

run(../Tarea 2/C1 WS\distribcunows.m') %Ejecución del script con nombre "distribcunows".

case 10 %Operaciones para warning sound C1 E2.

run(../Tarea 2/C1 E2\distribcunoedos.m') %Ejecución del script con nombre "distribcunoedos".

otherwise %En el caso de que el código introducido no coincida con los casos anteriores se realiza lo siguiente.

display('ERROR') %Escribir por pantalla error.
display('Introduzca un valor dentro del intervalo') %Escribir por pantalla que se debe hacer para que no de error.
return %Salida de la estructura de control.

end %Cierre de la estructura de control switch para códigos de Pasadas Cortas.

case 2 %Operaciones para opción Pasadas Largas.

PETICIÓN DE WARNING SOUND PASADAS LARGAS.

display('1-Warning sound P1') %Mostrar por pantalla que código es el warning sound P1.
display('2-Warning sound P2') %Mostrar por pantalla que código es el warning sound P2.
display('3-Warning sound P3') %Mostrar por pantalla que código es el warning sound P3.
display('4-Warning sound P4') %Mostrar por pantalla que código es el warning sound P4.
display('5-Warning sound P5') %Mostrar por pantalla que código es el warning sound P5.
display('6-Warning sound P8') %Mostrar por pantalla que código es el warning sound P8.
display('7-Warning sound P9') %Mostrar por pantalla que código es el warning sound P9.
display('8-Warning sound P10') %Mostrar por pantalla que código es el warning sound P10.
display('9-Warning sound P11') %Mostrar por pantalla que código es el warning sound P11.
display('10-Warning sound P12') %Mostrar por pantalla que código es el warning sound P12.
display('11-Warning sound P13') %Mostrar por pantalla que código es el warning sound P13.
display('12-Warning sound P16') %Mostrar por pantalla que código es el warning sound P16.
display('13-Warning sound P17') %Mostrar por pantalla que código es el warning sound P17.
display('14-Warning sound P18') %Mostrar por pantalla que código es el warning sound P18.
display('15-Warning sound P19') %Mostrar por pantalla que código es el warning sound P19.
display('16-Warning sound P20') %Mostrar por pantalla que código es el warning sound P20.

ws=input('¿Qué warning sound quiere estudiar?(En mayúsculas) ','s'); %Pregunta por pantalla el warning sound a analizar y lo almacena.

LISTADO PASADAS LARGAS

switch ws %Estructura de control switch que realiza las funciones que se definan para cada warning sound de Pasadas Largas.

case 1 %Operaciones para warning sound P1.

load(../Tarea 2/Pasadaslargas\Variable pasadalarga1\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora.
run(../Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".
run(../Tarea 2\directividadizq.m') %Ejecución del script con nombre "directividadizq".

case 2 %Operaciones para warning sound P2.

load(../Tarea 2/Pasadaslargas\Variable pasadalarga2\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora.
run(../Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".
```



```
run('..\Tarea 2\directivaddcha.m') %Ejecución del script con nombre "directivaddcha".
```

**case 3** %Operaciones para warning sound P3.

```
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga3\tabppal.mat') % Importación de la variable que contiene la matriz de presión sonora.  
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".  
run('..\Tarea 2\directivadizq.m') %Ejecución del script con nombre "directivadizq".
```

**case 4** %Operaciones para warning sound P4.

```
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga4\tabppal.mat') % Importación de la variable que contiene la matriz de presión sonora.  
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".  
run('..\Tarea 2\directivaddcha.m') %Ejecución del script con nombre "directivaddcha".
```

**case 5** %Operaciones para warning sound P5.

```
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga5\tabppal.mat') % Importación de la variable que contiene la matriz de presión sonora.  
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".  
run('..\Tarea 2\directivadizq.m') %Ejecución del script con nombre "directivadizq".
```

**case 6** %Operaciones para warning sound P8.

```
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga8\tabppal.mat') % Importación de la variable que contiene la matriz de presión sonora.  
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".  
run('..\Tarea 2\directivaddcha.m') %Ejecución del script con nombre "directivaddcha".
```

**case 7** %Operaciones para warning sound P9.

```
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga9\tabppal.mat') % Importación de la variable que contiene la matriz de presión sonora.  
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".  
run('..\Tarea 2\directivadizq.m') %Ejecución del script con nombre "directivadizq".
```

**case 8** %Operaciones para warning sound P10.

```
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga10\tabppal.mat') % Importación de la variable que contiene la matriz de presión sonora.  
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".  
run('..\Tarea 2\directivaddcha.m') %Ejecución del script con nombre "directivaddcha".
```

**case 9** %Operaciones para warning sound P11.

```
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga11\tabppal.mat') % Importación de la variable que contiene la matriz de presión sonora.  
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".  
run('..\Tarea 2\directivadizq.m') %Ejecución del script con nombre "directivadizq".
```

**case 10** %Operaciones para warning sound P12.

```
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga12\tabppal.mat') % Importación de la variable que contiene la matriz de presión sonora.  
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".  
run('..\Tarea 2\directivaddcha.m') %Ejecución del script con nombre "directivaddcha".
```

**case 11** %Operaciones para warning sound P13.

```
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga13\tabppal.mat') %Carga de la variable que contiene la matriz de presión sonora.  
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".  
run('..\Tarea 2\directivadizq.m') %Ejecución del script con nombre "directivadizq".
```

**case 12** %Operaciones para warning sound P16.

```
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga16\tabppal.mat') % Importación de la variable que contiene la matriz de presión sonora.  
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".  
run('..\Tarea 2\directivaddcha.m') %Ejecución del script con nombre "directivaddcha".
```

**case 13** %Operaciones para warning sound P17.

```
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga17\tabppal.mat') % Importación de la variable que contiene la matriz de presión sonora.  
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".  
run('..\Tarea 2\directivadizq.m') %Ejecución del script con nombre "directivadizq".
```

**case 14** %Operaciones para warning sound P18.

```
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga18\tabppal.mat') % Importación de la variable que contiene la matriz de presión sonora.  
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".  
run('..\Tarea 2\directivaddcha.m') %Ejecución del script con nombre "directivaddcha".
```

**case 15** %Operaciones para warning sound P19.

```
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga19\tabppal.mat') % Importación de la variable que contiene la matriz de presión sonora.  
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".  
run('..\Tarea 2\directivadizq.m') %Ejecución del script con nombre "directivadizq".
```

**case 16** %Operaciones para warning sound P20.

```
load('..\Tarea 2\Pasadaslargas\Variable pasadalarga20\tabppal.mat') % Importación de la variable que contiene la matriz de presión sonora.  
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".  
run('..\Tarea 2\directivaddcha.m') %Ejecución del script con nombre "directivaddcha".
```

```
otherwise %En el caso de que el código introducido no coincida con los casos anteriores se realiza lo siguiente.

display('ERROR') %Escribir por pantalla error.
display('Introduzca un valor dentro del intervalo') %Escribir por pantalla que se debe hacer para que no de error.
return %Salida de la estructura de control.

end %Cierre de la estructura de control switch para casos de Pasadas Largas.

otherwise %En el caso de que el código introducido no coincida con los casos anteriores se realiza lo siguiente.

display('ERROR') %Escribir por pantalla error.
display('Introduzca un valor dentro del intervalo') %Escribir por pantalla que se debe hacer para que no de error.
return %Salida de la estructura de control.

end %Cierre de la estructura de control switch para tipos de pasadas.
```

## Programa distribeuno

```
PETICIÓN DE VELOCIDAD A EJECUTAR PARA WARNING SOUND E1
display('1-Velocidad a 5 km/h') %Mostrar por pantalla que código es la velocidad a 5 km/h.
display('2-Velocidad a 10 km/h') %Mostrar por pantalla que código es la velocidad a 10 km/h.
display('3-Velocidad a 15 km/h') %Mostrar por pantalla que código es la velocidad a 15 km/h.
display('4-Velocidad a 20 km/h') %Mostrar por pantalla que código es la velocidad a 20 km/h.
display('5-Velocidad a 25 km/h') %Mostrar por pantalla que código es la velocidad a 25 km/h.
display('6-Velocidad a 30 km/h') %Mostrar por pantalla que código es la velocidad a 30 km/h.

v=input('¿Qué velocidad quiere estudiar? '); %Pregunta por pantalla código de velocidad a estudiar y almacenamiento en variable.

LISTADO DE VELOCIDADES PARA WARNING SOUND E1
switch v %Estructura de control switch que realiza las funciones que se definan para cada caso de velocidad para warning sound E1.

    case 1 %Operaciones para velocidad de 5 km/h.

        arch=dir('C1 E1 5*Photo2Dcent.txt'); %Obtención de lista de archivos con extensión .m para 5 km/h.
        numarch=length(arch); %Número de archivos que permiten obtener los datos con extensión .m para 5 km/h.

        for con=1:numarch %Bucle for para ejecutar los distintos ensayos para la velocidad definida anteriormente.

            switch con %Estructura de control switch que realice las funciones especificadas para cada ensayo de 5 km/h.

                case 1 %Operaciones para primer ensayo a 5 km/h.

                    load('..\E1\C1 E1 5\Variables C1E15uno(tabppal.mat)') %Importación de la variable que contiene la matriz de presión sonora del primer ensayo.
                    run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".
                    run('..\Tarea 2\directividadizq.m') %Ejecución del script con nombre "directividadizq".
                    clear %Limpieza del espacio de trabajo.
                    clc %Limpieza de variables.
                    pas=1; %Redefinición de variable pasada con valor correspondiente.
                    con=1; %Redefinición de variable contador con valor correspondiente.
                    ws=1; %Redefinición de variable warning sound con valor correspondiente.
                    v=1; %Redefinición de variable velocidad con valor correspondiente.

                case 2 %Operaciones para segundo ensayo a 5 km/h.

                    load('..\E1\C1 E1 5\Variables C1E15dos(tabppal.mat)') %Importación de la variable que contiene la matriz de presión sonora del segundo ensayo.
                    run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".
                    run('..\Tarea 2\directivaddcha.m') %Ejecución del script con nombre "directivaddcha".

            end %Cierre de la estructura de control switch para ensayos de la primera velocidad.

        end %Cierre del bucle que ejecuta los distintos ensayos.

    case 2 %Operaciones para velocidad de 10 km/h.

        arch=dir('C1 E1 10*Photo2Dcent.txt'); %Obtención de lista de archivos con extensión .m para 10 km/h.
        numarch=length(arch); %Número de archivos que permiten obtener los datos con extensión .m para 10 km/h.

        for con=1:numarch %Bucle for para ejecutar los distintos ensayos para la velocidad definida anteriormente.

            switch con %Estructura de control switch que realice las funciones especificadas para cada ensayo de 10 km/h.

                case 1 %Operaciones para primer ensayo a 10 km/h.

                    load('..\E1\C1 E1 10\Variables C1E110uno(tabppal.mat)') %Importación de la variable que contiene la matriz de presión sonora del primer ensayo.
                    run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".
                    run('..\Tarea 2\directividadizq.m') %Ejecución del script con nombre "directividadizq".

            end %Cierre de la estructura de control switch para ensayos de la segunda velocidad.
```

```
end %Cierre del bucle que ejecuta los distintos ensayos.

case 3 %Operaciones para velocidad de 15 km/h.

arch=dir('C1 E1 15*Photo2Dcent.txt'); % Obtención de lista de archivos con extensión .m para 15 km/h.
numarch=length(arch); % Número de archivos que permiten obtener los datos con extensión .m para 15 km/h.

for con=1:numarch %Bucle for para cambiar el valor de i (i es el número de ensayo) para esa velocidad.

    switch con %Estructura de control switch que realice las funciones especificadas para cada ensayo de 15 km/h.

        case 1 %Operaciones para primer ensayo a 15 km/h.

            load('..\E1\C1 E1 15\Variables C1E115uno\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora del primer ensayo.
            run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".
            run('..\Tarea 2\directividadizq.m') %Ejecución del script con nombre "directividadizq"
            clear %Limpieza del espacio de trabajo.
            clc %Limpieza de variables.
            pas=1; %Redefinición de variable pasada con valor correspondiente.
            con=1; %Redefinición de variable contador con valor correspondiente.
            ws=1; %Redefinición de variable warning sound con valor correspondiente.
            v=3; %Redefinición de variable velocidad con valor correspondiente.

        case 2 %Operaciones para segundo ensayo a 15 km/h.

            load('..\E1\C1 E1 15\Variables C1E115dos\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora del segundo ensayo.
            run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".
            run('..\Tarea 2\directivaddcha.m') %Ejecución del script con nombre "directivaddcha".

    end %Cierre de la estructura de control switch para ensayos de la tercera velocidad.

end %Cierre del bucle que ejecuta los distintos ensayos.

case 4 %Operaciones para velocidad de 20 km/h.

arch=dir('C1 E1 20*Photo2Dcent.txt'); % Obtención de lista de archivos con extensión .m para 20 km/h.
numarch=length(arch); % Número de archivos que permiten obtener los datos con extensión .m para 20 km/h.

for con=1:numarch %Estructura de control switch que realice las funciones especificadas para cada ensayo de 20 km/h.

    switch con %Estructura de control switch que realice las funciones especificadas para cada ensayo de 20 km/h.

        case 1 %Operaciones para primer ensayo a 20 km/h.

            load('..\E1\C1 E1 20\Variables C1E120uno\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora del primer ensayo.
            run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".
            run('..\Tarea 2\directividadizq.m') %Ejecución del script con nombre "directividadizq"
            clear %Limpieza del espacio de trabajo.
            clc %Limpieza de variables.
            pas=1; %Redefinición de variable pasada con valor correspondiente.
            con=1; %Redefinición de variable contador con valor correspondiente.
            ws=1; %Redefinición de variable warning sound con valor correspondiente.
            v=4; %Redefinición de variable velocidad con valor correspondiente.

        case 2 %Operaciones para segundo ensayo a 20 km/h.

            load('..\E1\C1 E1 20\Variables C1E120dos\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora del segundo ensayo.
            run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".
            run('..\Tarea 2\directivaddcha.m') %Ejecución del script con nombre "directivaddcha".

    end %Cierre de la estructura de control switch para ensayos de la cuarta velocidad.

end %Cierre del bucle que ejecuta los distintos ensayos.

case 5 %Operaciones para velocidad de 25 km/h.

arch=dir('C1 E1 25*Photo2Dcent.txt'); % Obtención de lista de archivos con extensión .m para 25 km/h.
numarch=length(arch); % Número de archivos que permiten obtener los datos con extensión .m para 25 km/h.

for con=1:numarch %Estructura de control switch que realice las funciones especificadas para cada ensayo de 25 km/h.

    switch con %Estructura de control switch que realice las funciones especificadas para cada ensayo de 25 km/h.

        case 1 %Operaciones para primer ensayo a 25 km/h.

            load('..\E1\C1 E1 25\Variables C1E125uno\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora del primer ensayo.
            run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".
            run('..\Tarea 2\directividadizq.m') %Ejecución del script con nombre "directividadizq"
            clear %Limpieza del espacio de trabajo.
```

```
clc %Limpieza de variables.
pas=1; %Redefinición de variable pasada con valor correspondiente.
con=1; %Redefinición de variable contador con valor correspondiente.
ws=1; %Redefinición de variable warning sound con valor correspondiente.
v=5; %Redefinición de variable velociad con valor correspondiente.

case 2 %Operaciones para segundo ensayo a 25 km/h.

load('..\E1\C1 E1 25\Variables C1E125dos\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora del segundo ensayo.
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre
run('..\Tarea 2\directivaddcha.m') %Ejecución del script con nombre "directivaddcha".

end %Cierre de la estructura de control switch para ensayos de la quinta velocidad.

end %Cierre del bucle que ejecuta los distintos ensayos.

case 6 %Operaciones para velocidad de 30 km/h.

arch=dir('C1 E1 30*\Photo2Dcent.txt'); % Obtención de lista de archivos con extensión .m para 30 km/h.
numarch=length(arch); % Número de archivos que permiten obtener los datos con extensión .m para 30 km/h.

for con=1:numarch %Estructura de control switch que realice las funciones especificadas para cada ensayo de 30 km/h.

switch con %Estructura de control switch que realice las funciones especificadas para cada ensayo de 30 km/h.

case 1 %Operaciones para primer ensayo a 30 km/h.

load('..\E1\C1 E1 30\Variables C1E130uno\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora del primer ensayo.
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".
run('..\Tarea 2\directividadizq.m') %Ejecución del script con nombre "directividadizq"
clear %Limpieza del espacio de trabajo.
clc %Limpieza de variables.
pas=1; %Redefinición de variable pasada con valor correspondiente.
con=1; %Redefinición de variable contador con valor correspondiente.
ws=1; %Redefinición de variable warning sound con valor correspondiente.
v=6; %Redefinición de variable velociad con valor correspondiente.

case 2 %Operaciones para segundo ensayo a 30 km/h.

load('..\E1\C1 E1 30\Variables C1E130dos\tabppal.mat') %Importación de la variable que contiene la matriz de presión sonora del segundo ensayo.
run('..\Tarea 2\plottridimensional.m') %Ejecución del script con nombre "plottridimensional".
run('..\Tarea 2\directivaddcha.m') %Ejecución del script con nombre "directivaddcha".

end %Cierre de la estructura de control switch para ensayos de la sexta velocidad.

end %Cierre del bucle que ejecuta los distintos ensayos.

otherwise %En el caso de que el código introducido no coincida con los casos anteriores se realiza lo anterior.

display('ERROR') %Escribir por pantalla error.
display('Introduzca un valor dentro del intervalo especificado') %Escribir por pantalla que se debe hacer para que no de error.
return %Salida de la estructura de control.

end %Cierre de la estructura de control switch para casos del warning sound E1.
```