



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

Red de sensores para la detección de drones aéreos en interiores

Grado en Ingeniería Mecánica

ALUMNO: Francisco de Asís Aznar de la Riera

DIRECTORES: Norberto Fernández García

Milagros Fernández Gavilanes

CURSO ACADÉMICO: 2024-2025

Universida_{de}Vigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

Red de sensores para la detección de drones aéreos en interiores

Grado en Ingeniería Mecánica
Intensificación en Tecnología Naval
Cuerpo General

Universida_{de}Vigo

RESUMEN

El auge de los drones en conflictos modernos ha puesto de manifiesto la necesidad de desarrollar sistemas de detección eficientes en entornos urbanos. En este contexto, este trabajo presenta el diseño e implementación de una red de sensores capaz de detectar la presencia de un dron en interiores basándose en el análisis de su firma acústica. La propuesta se basa en el despliegue de microcontroladores Raspberry Pi con micrófonos conectados a un sistema central, donde los datos acústicos son analizados en tiempo real mediante filtrado de frecuencia y umbrales de detección previamente calibrados. Para establecer la técnica de detección, se realizó un análisis detallado del sonido del dron. Aplicando un filtro paso-banda y estableciendo un umbral basado en la energía acústica en decibelios, se logró discriminar la presencia del dron en entornos ruidosos. Las pruebas realizadas validaron la eficacia del sistema, confirmando su capacidad de detección con una fiabilidad considerable. Los resultados demuestran que la metodología desarrollada es viable y escalable. Si bien en este estudio solo se ha trabajado con un dron, el procedimiento permite su aplicación a otros modelos. Este sistema representa un primer paso hacia la implementación de redes de sensores para la defensa en escenarios urbanos, con posibles aplicaciones en seguridad y vigilancia.

PALABRAS CLAVE

UAV, Dron, Detección, Red, Sensores, Micrófonos

AGRADECIMIENTOS

En primer lugar, quiero agradecer a la Escuela Naval Militar por este periodo de formación de 5 años, brindándome muchas experiencias y momentos que indudablemente me acompañarán toda la vida. A todos los mandos que he tenido, desde comandantes de brigada hasta brigadieres, por enseñarme que, más allá de los cuatro valores que se leen en el Patio de Aulas “Álvaro de Bazán”, la humildad y la entrega a los demás son los pilares fundamentales que deben sustentar los cimientos morales de cualquier militar.

A mis tutores, Norberto y Milagros, por su paciencia, apoyo y empeño durante el transcurso del trabajo, orientándome en todo momento y resolviéndome las dudas necesarias hasta el final.

A todos los “pilotos” que durante horas han estado volando el dron mientras realizaba diversas pruebas al sistema. En particular, al AA Martínez Iglesias y al GM2º García Amorena por colaborar con su dron particular.

A mis compañeros, la promoción 425-155, por todos los momentos vividos que nos han hecho una brigada especial.

Finalmente, agradecer a mi familia, por su apoyo constante durante estos años, por estar siempre, independientemente de la situación. En especial, quiero agradecer a mis padres, por haber sido un ejemplo de sacrificio, humildad y entrega por los demás, seréis siempre mi referencia.

Por último, quiero agradecer a los que siempre estuvieron, a los que, aunque se fueron, siguen estando, y a los que, aunque ausentes, nunca se marcharon.

CONTENIDO

Contenido	1
Índice de Figuras	6
Índice de Tablas.....	9
1 Introducción y objetivos	10
1.1 Introducción y motivación	10
1.2 Objetivos	11
1.3 Estructura	12
2 Estado del arte	14
2.1 Evolución del arma aérea	14
2.1.1 Definición e impacto del arma aérea	14
2.1.2 Contexto histórico.....	15
2.1.2.1 Primeros usos y “La Gran Guerra”	15
2.1.2.2 Consolidación en la Segunda Guerra Mundial	15
2.1.2.3 La Guerra fría y la tecnología de misiles	16
2.1.2.4 Conflictos contemporáneos y la llegada de los UAVs	16
2.2 El arma aérea en la actualidad: La era del dron	17
2.2.1 Clasificación de RPAS	17
2.2.1.1 Modelos de drones utilizados en la actualidad.	19
2.2.2 Aplicaciones militares de drones	19
2.2.3 Fases para hacer frente a un dron	21
2.3 Sistemas de detección	22
2.3.1 Definición de un sistema de detección	22
2.3.2 Componentes	22
2.3.3 Reseña histórica sobre los sistemas de detección de drones.....	22
2.4 Detección de UAVs	24
2.4.1 Métodos de detección actuales	24
2.4.2 Detección acústica	25
2.4.3 Detección óptica	25
2.4.4 Detección por radiofrecuencia	26
2.4.5 Detección multisensorial	26
2.5 Técnicas y sensores empleados en la detección sonora de UAVs	27
2.5.1 Principios básicos de la detección acústica de drones	27
2.5.2 Adquisición de datos.....	28
2.5.2.1 Tipos de sensores utilizados	29

2.5.2.2 Distintas configuraciones de sensores	29
2.5.3 Procesamiento de datos.....	30
2.5.4 Extracción y representación de las características acústicas	31
2.5.4.1 MFCC	32
2.5.4.2 HMM	32
2.5.5 Técnicas de identificación	32
2.5.5.1 Correlación de patrones e identificación de armónicos.....	32
2.5.5.2 Modelos estadísticos.....	33
2.5.5.3 Aprendizaje máquina o Machine Learning	33
2.6 Redes de sensores.....	35
2.6.1 Concepto general	35
2.6.2 Módulos de comunicación	35
2.6.2.1 WiFi.....	35
2.6.2.2 Bluetooth	36
2.6.2.3 ZigBee	36
2.6.2.4 LoRa	36
2.6.3 Aplicaciones militares de las redes de sensores.....	37
3 Desarrollo del TFG.....	39
3.1 Metodología	39
3.1.1 Propósito	39
3.1.2 Desarrollo de fases.....	40
3.2 Consideraciones previas.....	40
3.2.1 Hardware empleado	41
3.2.1.1 R��ter MicroTik.....	41
3.2.1.2 Micr��fono de condensador TONOR TC30	42
3.2.1.3 Raspberry Pi 3 modelo B.....	42
3.2.1.4 Dron.....	43
3.2.1.5 Port��til	44
3.2.2 Software, editor gr��fico y librer��as empleadas	44
3.2.2.1 Uso de la IA generativa	46
3.2.3 Escenario de pruebas	46
3.3 Primera fase: Establecimiento de red de sensores	46
3.3.1 Configuraci��n del r��ter.....	47
3.3.2 Configuraci��n de dispositivos	48
3.3.2.1 Configuraci��n de ordenador central	48
3.3.2.2 Raspberry Pi Modelo 3 B	48
3.3.2.3 Acceso remoto mediante comando ssh.....	49

3.4 Segunda fase: Recopilación, transmisión y recepción de datos	51
3.4.1 Captura y transmisión de sonido desde las Raspberry Pi	51
3.4.1.1 Flujo de ejecución del script.....	52
3.4.2 Recepción del audio en el sistema de procesamiento	53
3.4.2.1 Definición de variables y configuración inicial.....	54
3.4.2.2 Inicio en la recepción del audio.....	54
3.4.2.3 Almacenamiento del audio	56
3.4.2.4 Finalización y reinicio de la sesión	56
3.4.2.5 Ejemplo implementación.....	57
3.5 Tercera fase: Análisis de datos.....	59
3.5.1 Análisis espectral	59
3.5.1.1 Flujo de ejecución del script.....	59
3.5.1.2 Transformada de Fourier de Corto Plazo (STFT)	60
3.5.1.3 Espectrogramas	61
3.5.2 Análisis frecuencial	62
3.6 Cuarta fase: Técnica de detección para identificación de dron.....	64
3.6.1 Aplicación del filtro paso-banda.....	65
3.6.2 Generación de los espectrogramas filtrados	65
3.6.3 Extracción de energía en las bandas de interés.....	67
3.6.4 Definición del umbral de detección	67
3.7 Quinta fase: Aplicación de la técnica de detección en tiempo real.....	68
3.7.1 Configuración del sistema y carga del umbral de detección	69
3.7.2 Definición y aplicación de filtro paso-banda.....	69
3.7.3 Procesamiento de archivos de audio	70
3.7.4 Generación de espectrogramas, extracción de energía y comparación con umbral	70
3.7.5 Almacenamiento de resultados y monitoreo en tiempo real.....	71
4 Resultados, discusión y prueba de validación	74
4.1 Prueba 1: Prueba general de funcionamiento (sin transmisión en tiempo real).....	74
4.1.1 Procedimiento	74
4.1.2 Flujo de ejecución.....	75
4.1.2.1 Carga y filtrado de audio	75
4.1.2.2 Generación de espectrograma	75
4.1.2.3 Comparación con el umbral de detección	75
4.1.3 Resultados y discusión.....	76
4.2 Prueba 2: Análisis de falsos positivos.....	78
4.2.1 Propósito	78

4.2.2 Resultados y discusión.....	78
4.3 Prueba 3: Análisis de consumo energético	81
4.3.1 Batería portátil	81
4.3.2 Modificación de script “Send_audio.py”	82
4.3.3 Resultado	83
4.4 Prueba 4: Cambio de sensor acústico.....	83
4.4.1 Modelo utilizado	83
4.4.2 Grabación de audios y comprobación con umbral original	84
4.4.3 Ajuste de umbral y filtro paso-banda.....	86
4.4.4 Generación de espectrogramas e identificación de frecuencias dominantes	86
4.4.5 Comprobación de funcionamiento con nuevo umbral.....	89
4.4.6 Integración con sistema en tiempo real	90
4.5 Escenario de validación: Red de dos sensores con detección del dron en tiempo real	91
4.5.1 Escenario propuesto.....	91
4.5.2 Secuencia de simulación.....	93
4.5.3 Resultados.....	94
4.5.4 Discusión de resultados	96
4.5.5 Eliminación de umbral superior.....	99
4.6 Prueba 5: Impacto del ruido sin umbral superior	100
4.6.1 Flujo de ejecución.....	101
4.6.2 Resultados.....	101
5 Conclusiones y líneas futuras	103
5.1 Conclusiones	103
5.2 Líneas futuras	104
6 Bibliografía.....	106
Anexo I: Implicaciones Sociales, y/o Económicas, y/o Ambientales	111
Anexo II: Reflexiones Éticas y Sociales	112
Anexo III: Código de envío de audio desde raspberry pi.....	113
Anexo IV: Código de recepción de audio en tiempo real	114
Anexo V: Código de generación de espectrogramas.....	117
Anexo VI: Código de análisis frecuencial.....	118
Anexo VII: Código de generación de boxplot.....	120
Anexo VIII: Código de filtro paso-banda y definición de umbral	122
Anexo IX: Código de pruebas del dron.....	124

Anexo X: Código de detección en tiempo real.....	126
Anexo XI: Código de creación de ruido.....	129
Anexo XII: QR para audios del UAV grabados desde TONOR TC30.....	130
Anexo XIII: QR para audios sonido externo.....	131
Anexo XIV: QR para audios del UAV grabados desde Logitech C615	132
Anexo XV: QR para audios de ruido generado.....	133

ÍNDICE DE FIGURAS

Figura 2-1 Aeronave Nievport 17 con ametralladora en el plano superior [9].	15
Figura 2-2 Memphis Belle, el primer B-17 en terminar 25 misiones de bombardeo [12]	16
Figura 2-3 Dron “Baba Yaga” utilizado por las fuerzas ucranianas [17].	19
Figura 2-4 Operario ucraniano en escenario de combate manejando dron [21].	20
Figura 2-5 Fases para hacer frente a amenaza de UAV [14].	21
Figura 2-6 Componentes y proceso de detección.	22
Figura 2-7 Sistema de detección de dos “bocinas” en Boiling Field, Estados Unidos, 1921 [1].	23
Figura 2-8 Espejo acústico desarrollado por Reino Unido [25].	23
Figura 2-9 Técnicas de detección de drones [27].	25
Figura 2-10 Ejemplo de array de micrófonos [34]	29
Figura 2-11 Análisis en el dominio del tiempo de un dron [32]	30
Figura 2-12 Uso de la técnica de STFT [32]	31
Figura 2-13 Ejemplo de espectrograma de sonido de un dron y de sonido ambiente [39]	31
Figura 2-14 Proceso de obtención de los MFCC [38].	32
Figura 2-15 WSN en escenario de guerra [45].	37
Figura 3-1 Esquema de fases.	40
Figura 3-2 Arquitectura del sistema	41
Figura 3-3 Rúter MicroTik [52]	41
Figura 3-4 TONOR TC30 [54].	42
Figura 3-5 Vista general Raspberry Pi 3 Modelo B [55]	43
Figura 3-6 Dron profesional E88 4K empleado en las pruebas [56].	44
Figura 3-7 Sala de estar del cuartel Marqués de la Victoria	46
Figura 3-8 Configuración del rúter.	47
Figura 3-9 Configuración avanzada del rúter	47
Figura 3-10 Obtención de IP del ordenador	48
Figura 3-11 Interfaz gráfica Raspberry Pi Modelo 3 B y conexión a WiFi	48
Figura 3-12 Comando <i>ifconfig</i> para resaltar características de red	49
Figura 3-13 Comando <i>ping</i> desde ordenador a Raspberry Pi para comprobar conexión.	49
Figura 3-14 Comando <i>ssh</i> .	50
Figura 3-15 Estado de conexión de los dispositivos en la red de sensores	50
Figura 3-16 Configuración micrófono	51
Figura 3-17 Captura y transmisión de audio en las Raspberry Pi	53
Figura 3-18 Inicio de sistema de detección en tiempo real	55
Figura 3-19 Ejemplo de transmisión y recepción de audio en la Raspberry Pi 1.	57
Figura 3-20 Ejemplo de transmisión y recepción de audio en la Raspberry Pi 2.	57

Figura 3-21 Transmisión simultánea de ambas Raspberry Pi	58
Figura 3-22 Almacenamiento en formato WAV	58
Figura 3-23 Espectrogramas del dron a distintas distancias del micrófono	61
Figura 3-24 Secuencia del análisis frecuencial	62
Figura 3-25 Distribución frecuencial de los distintos audios del dron	63
Figura 3-26 Distribución de frecuencias predominantes en los audios del dron.....	64
Figura 3-27 Espectrogramas filtrados de los audios del dron a distintas distancias	66
Figura 3-28 Umbral de detección del dron.....	68
Figura 3-29 Archivo TXT que contiene el umbral de detección.....	69
Figura 4-1 Resultados de detección.....	77
Figura 4-2 Análisis de sonido de guerra a varias distancias.....	78
Figura 4-3 Análisis de sonido de disparos a varias distancias	79
Figura 4-4 Análisis de sonido de bombardeo aéreo a varias distancias	80
Figura 4-5 Análisis de sonido de lluvia a varias distancias.....	80
Figura 4-6 Batería portátil conectada a Raspberry Pi.....	82
Figura 4-7 Duración batería portátil	83
Figura 4-8 Cámara de web <i>Logitech C615</i> [60]	84
Figura 4-9 Audios almacenados en ordenador y grabados por el <i>Logitech C615</i>	84
Figura 4-10 Resultados de detección con nuevo micrófono	85
Figura 4-11 Espectrogramas del dron con micrófono <i>Logitech C615</i>	87
Figura 4-12 Análisis frecuencial del dron con micrófono <i>Logitech C615</i>	88
Figura 4-13 Distribución de frecuencias dominantes de los distintos audios	88
Figura 4-14 Umbral de detección para micrófono <i>Logitech C615</i>	89
Figura 4-15 Resultados micrófono <i>Logitech C615</i> con nuevo umbral de detección	90
Figura 4-16 Esquema de maniobra del UAV y distribución de sensores.....	92
Figura 4-17 Imágenes del escenario	93
Figura 4-18 Secuencia prueba de validación.....	94
Figura 4-19 Comparación de archivo con y sin dron	95
Figura 4-20 Detecciones.....	95
Figura 4-21 Resultado diario de detección.....	95
Figura 4-22 Comparación de latencias	96
Figura 4-23 Resumen de resultados en esquema de maniobra del UAV	97
Figura 4-24 Resultados Raspberry Pi 1	98
Figura 4-25 Resultados Raspberry Pi 2	99
Figura 4-26 Resultados totales sin umbral superior	99
Figura 4-27 Resultados Raspberry Pi 2 sin umbral superior.....	100

Figura 4-28 Resultados de detección de dron con ruido creado.....	102
Figura 4-29 Gráfica con resultados de audio creado y un solo umbral	102
Figura 4-30 Gráfica con resultados de audio creado y ambos umbrales.....	102

ÍNDICE DE TABLAS

Tabla 2-1 Clasificación de RPAS (tomada de [15])	18
Tabla 2-2 Características de los sistemas de detección (tomada de [14])	27
Tabla 2-3 Resumen técnicas de identificación	34
Tabla 2-4 Métodos de comunicación inalámbrica	36
Tabla 3-1 Características router MicroTik (tomada de [53])	42
Tabla 3-2 Parámetros captura de audio	51
Tabla 4-1 Características batería portátil <i>VEEKTOMX Mini Power Bank</i> (tomada de [59])	82
Tabla 4-2 Especificaciones <i>Logitech C615</i> (tomada de [60])	84
Tabla 4-3 Resumen de resultados de la prueba	96

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Introducción y motivación

El nacimiento de la aviación marcó un punto de inflexión en los conflictos armados, convirtiendo al cielo en un nuevo frente de batalla y obligando a desarrollar métodos innovadores para defenderse de amenazas aéreas. Durante la Primera Guerra Mundial, la humanidad se enfrentó por primera vez a la necesidad de detectar amenazas provenientes del aire. Sistemas rudimentarios como enormes dispositivos acústicos y observadores en torres fueron los precursores de lo que más tarde se convertiría en tecnologías más avanzadas [1]. Con el tiempo, el radar revolucionó la defensa aérea durante la Segunda Guerra Mundial, al proporcionar una capacidad sin precedentes para detectar aviones enemigos a largas distancias, incluso en condiciones de poca visibilidad [1]. Este avance permitió anticipar movimientos hostiles, organizar defensas más efectivas y proteger activos estratégicos con mayor eficacia frente a ataques aéreos. La afirmación de que *“Un hombre que quiere hacer un buen instrumento debe primero tener una comprensión precisa de para qué se va a usar el instrumento; y aquel que pretende construir un buen instrumento de guerra debe primero preguntarse cómo será la próxima guerra”* ([2]) ofrece un enfoque que resalta cómo la evolución de las herramientas de combate ha estado siempre vinculada a la necesidad de anticiparse y responder eficazmente a nuevas amenazas.

En los conflictos modernos, los drones representan la última iteración de esta dinámica. Estas plataformas no tripuladas incluyen diversas categorías, como los UAVs (*Unmanned Aerial Vehicles*), los UGVs (*Unmanned Ground Vehicles*), los USVs (*Unmanned Surface Vehicles*) y los UUVs (*Unmanned Underwater Vehicles*), cada uno adaptado a distintos entornos operativos. Sin embargo, por el contexto en el que nos encontramos, nos centramos en los UAVs, cuya versatilidad, bajo costo y facilidad de despliegue los han convertido en elementos clave tanto en operaciones aéreas militares como civiles. A pesar de su creciente presencia en el campo de batalla, su potencial aún no se ha explotado completamente debido a la falta de una visión estratégica unificada. Actualmente, no existe un “profeta del poder de los drones”, un líder que haya integrado plenamente estas herramientas en las doctrinas militares para transformar el arte de la guerra, como ocurrió con los comandantes de tanques en la década de 1930 [3]. Esto deja un terreno abierto para que el país o entidad que logre adoptar e innovar en el uso de drones, ya sea mediante inteligencia artificial, enjambres masivos o integración con aeronaves tripuladas, obtenga una ventaja táctica y estratégica significativa. Por su parte, España, no es ajena a esta cuestión, y, de hecho, el Almirante Jefe del Estado Mayor de la Armada (AJEMA), Almirante General Antonio Piñeiro Sánchez, precisamente comenta dicha importancia: *“En el 2050, para seguir cumpliendo nuestra misión debemos ser...vanguardistas en la tecnología...incluyendo la expansión y mejora de...sistemas no tripulados con mayor alcance, autonomía y resiliencia”* [4].

Frente a estas amenazas, los conflictos recientes han evidenciado la creciente importancia de los sistemas de detección de drones. En el enfrentamiento entre Armenia y Azerbaiyán por la región de Nagorno-Karabaj, los drones desempeñaron un papel central en las operaciones militares, obligando a ambas partes a buscar métodos efectivos de detección y neutralización [3]. De manera similar, en Siria, Libia y Yemen, el uso de drones por grupos insurgentes y fuerzas estatales subrayó la necesidad de innovar en tecnologías que permitan localizar y contrarrestar estas amenazas de manera eficiente [3]. Sin embargo, el conflicto en Ucrania ha llevado esta problemática a un nuevo nivel, mostrando cómo los avances en tecnologías de detección son esenciales para contrarrestar el uso masivo de drones. Redes de sensores acústicos desplegadas en entornos urbanos han demostrado ser una herramienta eficaz para identificar y rastrear drones enemigos. Miles de micrófonos interconectados recopilan datos en tiempo real, permitiendo localizar amenazas aéreas con alta precisión [5]. Estos micrófonos trabajan en conjunto formando una red sincronizada, en la que cada dispositivo registra las señales acústicas y las envía a un sistema central [5]. Este sistema analiza las diferencias de tiempo y amplitud entre las señales para triangular la posición exacta de los drones. Esta tecnología, desarrollada en un garaje por dos ingenieros ucranianos, ha logrado identificar con éxito drones enemigos en entornos complejos, demostrando su efectividad incluso en áreas densamente pobladas [5]. Este enfoque no solo reduce la dependencia de sistemas más costosos, sino que también ofrece una solución adaptada a los entornos urbanos, donde las características del terreno complican el uso de tecnologías tradicionales como el radar.

La red acústica desplegada en Ucrania no solo ha demostrado su efectividad, sino que ha captado la atención internacional como una solución prometedora en defensa aérea. Según el Teniente General Sean A. Gainey, comandante general de la defensa espacial y defensa de misiles de Estados Unidos o SMDC (*Space and Missile Defense Command*), estas tecnologías representan *“una opción de bajo costo que permite una detección temprana y precisa de amenazas aéreas en entornos desafiantes”* [6]. Además, enfatizó que *“su implementación no solo es viable, sino necesaria para afrontar las amenazas emergentes”* [6]. Este enfoque está siendo probado por países de la OTAN como Alemania y Rumanía, quienes ya han comenzado a evaluar la integración de estas redes en sus sistemas de defensa. Este interés internacional subraya la capacidad de estas tecnologías para complementar los sistemas tradicionales, especialmente en escenarios donde los recursos son limitados. Además, su implementación podría fortalecer significativamente la capacidad de defensa aérea integrada o IADS (*Integrated Air Defence System*) de la OTAN, mejorando la detección y la respuesta frente a amenazas emergentes [6].

En el contexto de la guerra urbana, donde proteger infraestructuras críticas y reducir los riesgos para la población civil son prioridades fundamentales, es crucial implementar redes de sensores que permitan una vigilancia continua sin exponer a las fuerzas de seguridad. En lugar de desplegar personal en cada punto de vigilancia, estas redes ofrecen una solución eficiente y segura mediante el monitoreo automatizado de los espacios. Además, los entornos urbanos presentan desafíos como la congestión y la complejidad estructural, lo que requiere de tecnologías que puedan operar eficazmente en áreas densamente pobladas. Por ello, el desarrollo de sistemas modernos de detección de drones no solo responde a una necesidad táctica, sino que también representa un paso hacia la protección sostenible de espacios críticos en escenarios urbanos.

Inspirados por las lecciones aprendidas en conflictos recientes y las oportunidades que ofrece la innovación tecnológica, esta propuesta busca diseñar y desarrollar una red de sensores capaz de operar en entornos urbanos, proporcionando una solución eficiente y automatizada para la detección de drones.

1.2 Objetivos

El objetivo principal de este trabajo fin de grado es investigar la viabilidad de una red de sensores de bajo coste, basada en microcontroladores y sensores acústicos, diseñada para la detección de UAVs en escenarios de guerra urbana. El enfoque del proyecto radica en el despliegue de esta red en pequeñas construcciones o edificios estratégicos, donde equipos militares, como una unidad de Infantería de Marina, deban defender una posición ante posibles operaciones de drones enemigos. Esta solución busca incrementar la seguridad y eficacia operativa al minimizar la exposición directa de personal humano.

Para alcanzar este objetivo, se emplearán sensores acústicos capaces de identificar la firma sonora característica de un UAV. A partir del estudio previo de sus características acústicas, los datos obtenidos en tiempo real podrán compararse con dicha referencia, permitiendo su detección incluso en entornos ruidosos típicos de un conflicto bélico, con sonidos de disparos y explosiones. Para este estudio, se realizarán grabaciones de sonidos reales del dron en un ambiente simulado, evaluando la capacidad de los sensores para diferenciar estas señales con precisión.

Finalmente, se diseñará e implementará la programación necesaria para garantizar una integración eficiente entre los sensores y el sistema central. Este software será clave para coordinar el procesamiento centralizado de la información de manera distribuida, estableciendo así las bases para futuras investigaciones y aplicaciones prácticas en la detección de drones en entornos urbanos.

1.3 Estructura

Tras establecer el marco teórico y definir los objetivos que orientan este trabajo, a continuación, se presenta la estructura que seguirá la memoria, con el fin de organizar de manera clara y concisa los procedimientos realizados y los resultados obtenidos. La documentación se divide en cinco capítulos principales, seguidos de la bibliografía y anexos. La estructura es la siguiente:

- Capítulo 1: Introducción y objetivos. Este capítulo establece el contexto del trabajo, destacando la relevancia de la detección de drones en escenarios de guerra urbana y su importancia en conflictos actuales y la motivación que subyace en el desarrollo de una red de sensores de bajo coste. Además, se describen los objetivos principales del proyecto, ofreciendo una visión clara del alcance del TFG.
- Capítulo 2: Estado del arte. En este apartado se realiza una revisión de los conceptos teóricos y prácticos relevantes para el desarrollo del trabajo. Se abordan los siguientes puntos:
 - Evolución histórica del arma aérea y los sistemas de detección, incluyendo la transición a los drones como amenazas específicas en los conflictos modernos.
 - Características y aplicaciones de drones en la actualidad y cómo su rol resulta crucial a la hora de realizar el planeamiento de una misión.
 - Técnicas de detección y análisis de datos acústicos para su posterior implementación en el sistema.
 - Redes de sensores aplicadas al ámbito militar, incluyendo sistemas de comunicación para la interconexión de nodos, análisis de tecnologías como WiFi, Bluetooth, o ZigBee.
- Capítulo 3: Desarrollo del TFG. En este capítulo se describen las herramientas y metodologías empleadas para desarrollar la red de sensores propuesta. Incluye el diseño e implementación de la red, la simulación de entornos urbanos para pruebas y el desarrollo de sistemas de detección mediante filtros y umbrales.
- Capítulo 4: Resultados obtenidos y validación. Este capítulo está dedicado a la presentación y análisis de los resultados obtenidos en distintas pruebas que validan el sistema de detección y la red de sensores propuesta. Se evalúa el rendimiento de los sensores en condiciones simuladas, la efectividad del procesamiento para identificar amenazas, así como un análisis del consumo energético de los dispositivos.
- Capítulo 5: Conclusiones y líneas futuras. En el último capítulo se sintetizan las conclusiones principales del trabajo, destacando la viabilidad y las limitaciones de la propuesta. También se plantean líneas de investigación futuras, enfocadas en la mejora de la autonomía de los sensores, la ampliación del rango de detección y el desarrollo de algoritmos más avanzados para el procesamiento de datos.

Al final del documento se incluye la bibliografía utilizada, así como los anexos necesarios para complementar la información presentada, proporcionando detalles adicionales que faciliten la comprensión del trabajo.

2 ESTADO DEL ARTE

El presente capítulo presenta el marco teórico que fundamenta el desarrollo de este trabajo. A través de un recorrido histórico y tecnológico, se contextualizan los avances en el uso del arma aérea, así como la evolución de los sistemas de detección, haciendo énfasis en los drones, y se examinan las herramientas tecnológicas actuales que sustentan el diseño de la red de sensores propuesta.

2.1 Evolución del arma aérea

La guerra moderna ha estado profundamente influenciada por los avances tecnológicos que se han ido sucediendo a lo largo de la historia, y el dominio del aire se ha convertido en un pilar fundamental en los conflictos armados. Desde los primeros intentos de ganar la altura como elemento estratégico hasta la llegada del UAV, la evolución del arma aérea ha cambiado radicalmente la manera de concebir y ejecutar las operaciones militares y, por tanto, el modo de “hacer la guerra” [2]. La afirmación de que *“la guerra no es sino una continuación de la política por otros medios”* [7] ilustra cómo los avances tecnológicos, las nuevas políticas y las estrategias están interconectados, de ahí que el avance del arma aérea, crucial en el desarrollo de una contienda, ha influido mutuamente en la planificación táctica y estratégica.

2.1.1 Definición e impacto del arma aérea

El arma aérea se puede definir como el conjunto de dispositivos tecnológicos utilizados para realizar operaciones militares en el aire. Desde sus inicios, esta categoría ha transformado profundamente la estrategia militar, abriendo una nueva dimensión en el campo de batalla. Esta capacidad para operar en el espacio aéreo ha permitido a los ejércitos superar barreras geográficas y alcanzar objetivos estratégicos con una rapidez y precisión sin precedentes [2].

A lo largo de su evolución, el arma aérea ha pasado de globos aerostáticos y aviones rudimentarios a plataformas altamente sofisticadas como cazas de quinta generación y drones autónomos [2]. Este avance tecnológico no solo ha permitido un mayor alcance y precisión, sino que también ha redefinido cómo se planifican y ejecutan las operaciones militares. La posibilidad de realizar ataques precisos desde el aire ha obligado a los estrategas a reconsiderar las dinámicas del combate terrestre y marítimo, integrando el control del espacio aéreo como un elemento clave en sus planes [2].

En el contexto moderno, el arma aérea desempeña un papel esencial en los conflictos. Las plataformas actuales no solo están diseñadas para el combate, sino también para tareas de vigilancia, reconocimiento y logística, lo que subraya su carácter multidimensional. Además, el desarrollo de UAVs ha ampliado aún más estas capacidades, permitiendo a los ejércitos operar con un menor riesgo para el personal mientras se mantienen altamente efectivos en el campo de batalla [8].

2.1.2 Contexto histórico

Desde sus primeras aplicaciones, el poder aéreo se concibió como un medio para alcanzar objetivos estratégicos, superar barreras geográficas y transformar la dinámica de los conflictos armados. Este apartado explora la evolución histórica del arma aérea, desde su concepción inicial hasta su consolidación como un componente esencial en el campo de batalla moderno.

2.1.2.1 Primeros usos y “La Gran Guerra”

El desarrollo inicial de las operaciones militares en el aire comenzó con el primer vuelo de los hermanos Wright en su avión *Flyer*, un avance que marcó el inicio de una nueva era en la movilidad y la comunicación. Aunque en un principio las aeronaves se destinaron a fines recreativos o exhibiciones, los ejércitos pronto reconocieron su gran potencial para la observación estratégica y el transporte táctico. En 1909, el mayor italiano Giulio Douhet predijo que las aeronaves serían fundamentales en los conflictos futuros [2], una idea que se materializó en la guerra entre Italia y Libia en 1911, donde se utilizaron para reconocimiento y bombardeo [9].

Durante la Primera Guerra Mundial, el uso de aeronaves evolucionó rápidamente, pasando de simples plataformas de observación a aviones de combate y bombardeo. Los modelos más destacados incluyeron aeronaves de observación temprana, también conocidas como AEW (*Air Early Warning*), y, hacia el final del conflicto, diseños especializados como el Bristol Scout, el Nieuport 17 (Figura 2-1) y el Havilland DH2, que comenzaron a integrar ametralladoras montadas de forma estratégica, teniendo la hélice en la parte posterior [9]. Estos avances demostraron el potencial del dominio aéreo para influir directamente en las operaciones terrestres y establecieron la base para las doctrinas de superioridad aérea.



Figura 2-1 Aeronave Nieuport 17 con ametralladora en el plano superior [9].

Roland Garros y Anthony Fokker desempeñaron un papel crucial en la evolución de la tecnología aérea al introducir sistemas de sincronización para las ametralladoras, permitiendo disparar a través de las hélices sin dañarlas. Esta innovación aumentó significativamente la efectividad de los aviones de combate y estableció un nuevo estándar en el diseño de aeronaves militares [9].

En España, la aviación militar comenzó a desarrollarse a principio del siglo XX. En 1917, se creó la Aeronáutica Naval, y en 1925, durante el desembarco de Alhucemas, se llevó a cabo la primera operación aeronaval de la historia, donde hidroaviones españoles desempeñaron un papel crucial en el éxito de la misión [10].

2.1.2.2 Consolidación en la Segunda Guerra Mundial

El poderío aéreo desempeñó un papel central durante la Segunda Guerra Mundial, demostrando su capacidad para realizar operaciones independientes y apoyar a las fuerzas terrestres y navales. Las fuerzas aéreas aliadas utilizaron diferentes enfoques para maximizar su impacto: mientras que los británicos emplearon bombardeos de saturación nocturnos para debilitar la moral del enemigo, los estadounidenses optaron por bombardeos de precisión diurnos dirigidos a objetivos industriales clave. Este último método, aunque arriesgado para las tripulaciones, permitió dañar significativamente la

capacidad de producción del Eje gracias a avances como el visor Norden y las aeronaves B-17 (mostrada en la Figura 2-2) y B-25 [11].



Figura 2-2 Memphis Belle, el primer B-17 en terminar 25 misiones de bombardeo [12]

En el Pacífico, el Ejército de los Estados Unidos llevó a cabo bombardeos incendiarios masivos sobre ciudades japonesas, culminando con el lanzamiento de las bombas atómicas en Hiroshima y Nagasaki desde el superbombardero B-29. Estos ataques no solo pusieron fin a la guerra en el Pacífico, sino que también demostraron el potencial decisivo del arma aérea para influir en el desenlace de los conflictos modernos [13].

La tecnología también avanzó en el ámbito de la escolta y protección de los bombarderos. Los cazas de largo alcance permitieron reducir las bajas aliadas al proporcionar cobertura durante misiones críticas, consolidando la importancia de la superioridad aérea para garantizar el éxito de las operaciones terrestres, como el Desembarco en Normandía [11].

2.1.2.3 La Guerra fría y la tecnología de misiles

El impacto del poderío aéreo durante la Segunda Guerra Mundial y la creciente amenaza soviética llevaron a la consolidación de una fuerza aérea independiente en 1947, estableciendo su rol como una herramienta estratégica clave. Durante la Guerra de Corea, las aeronaves de reacción y los sistemas de lanzamiento demostraron su eficacia, permitiendo a las fuerzas aliadas mantener la superioridad aérea y ofrecer apoyo directo en el campo de batalla. A medida que la Guerra Fría avanzó, el desarrollo de bombarderos estratégicos con capacidad de transportar armas nucleares, junto con mejoras como el reaprovisionamiento de combustible en vuelo, consolidó el papel del arma aérea como un disuasor nuclear global [11].

La introducción de misiles balísticos intercontinentales en la década de 1960 marcó un hito tecnológico significativo. Estos sistemas, ejemplificados por el misil Atlas, equipado con una cabeza de combate termonuclear, ampliaron el alcance y la capacidad destructiva del poderío aéreo. La tensión entre las superpotencias llevó al desarrollo de planes como el SIOP (*Single Integrated Operation Plan*), que preparaba una respuesta nuclear masiva en caso de un ataque soviético [9].

2.1.2.4 Conflictos contemporáneos y la llegada de los UAVs

En los últimos años, la llegada de los UAVs ha seguido transformando el panorama del poder aéreo, consolidándose como herramientas clave en la estrategia militar moderna. En conflictos ya mencionados como el de Nagorno-Karabaj, Azerbaiyán empleó de manera efectiva los drones para misiones de reconocimiento y ataque, logrando una ventaja significativa sobre las fuerzas armenias. Estos drones permitieron identificar y neutralizar objetivos estratégicos con precisión, alterando las dinámicas tradicionales del combate. De manera similar, en Afganistán, las fuerzas estadounidenses utilizaron drones para misiones de reconocimiento, proporcionando inteligencia en tiempo real y apoyo a las operaciones terrestres. Más recientemente, en el conflicto de Ucrania, los UAVs han sido empleados no

solo para reconocimiento y ataque, sino también para apoyo a la artillería, mejorando la precisión y efectividad de los bombardeos. Estas experiencias han demostrado la capacidad de los drones para adaptarse a diversos entornos operativos y su eficacia en la ejecución de misiones críticas [3].

De esta manera, el arma aérea ha evolucionado de ser un recurso complementario para convertirse en un componente esencial de la estrategia militar. Su capacidad para adaptarse a las necesidades de cada periodo histórico refleja no solo el avance tecnológico, sino también su impacto profundo en la forma en que se conciben y ejecutan las guerras. Desde los primeros aviones de reconocimiento hasta los drones modernos, el poder aéreo ha demostrado ser una fuerza decisiva en el campo de batalla.

2.2 El arma aérea en la actualidad: La era del dron

El dron, término que deriva del inglés “*drone*”, es un concepto que se ha adaptado al español y recientemente ha sido incluido en el diccionario de la Real Academia Española (RAE). Aunque su uso generalizado ha popularizado el término “dron”, existen otras denominaciones técnicas ampliamente aceptadas y que se utilizarán indistintamente a lo largo del trabajo, como RPAS (*Remotely Piloted Aircraft Systems*), UAS (*Unmanned Aircraft Systems*) o RPA (*Remotely Piloted Aircraft*) [14]. De estos términos el primero en usarse fue UAS, no obstante, se ha extendido el empleo de RPAS con la intención de aclarar que las aeronaves son pilotadas, aunque sea de manera remota. Esta última denominación será la utilizada en las Fuerzas Armadas de España [15].

Se pueden distinguir tres elementos principales en la composición de un dron [15]:

- Segmento aéreo: Este segmento incluye la aeronave pilotada remotamente (RPA) y su carga útil. Las aeronaves pueden ser de ala fija, ala rotatoria o aerostatos, y están diseñadas para transportar distintas cargas útiles clasificadas en sensores, relés de comunicaciones, armas y otros dispositivos.
- Segmento de superficie: Este segmento comprende:
 - Elemento de control y explotación: Estaciones que controlan la aeronave y la carga útil y gestionan la información captada.
 - Elementos de apoyo: Recursos necesarios para el despliegue, mantenimiento y recuperación del dron, que varían según la clase de dron.
- Enlace de datos: Constituye el medio de comunicación entre la aeronave y los elementos de control. Los enlaces pueden ser de alcance visual o LOS (*Line Of Sight*) o más allá del horizonte visual o BLOS (*Beyond Line Of Sight*).

2.2.1 Clasificación de RPAS

Los RPAS abarcan una amplia variedad de configuraciones y capacidades que les permiten adaptarse a diferentes roles y escenarios operativos. Para comprender mejor su diversidad, se pueden clasificar en función de varios criterios, como su tamaño, capacidad de carga, autonomía o tipo de misión. Esta categorización no solo permite identificar las características técnicas de cada sistema, sino también evaluar sus aplicaciones prácticas en contextos civiles y militares. En la Tabla 2-1 se puede ver esta clasificación, compartida tanto por la OTAN (“Organización Tratado Atlántico Norte”) como por las FAS [15].

Clase	Categoría	Empleo habitual	Altura de operación normal	Radio de Misión
Clase I (<150 kg)	Micro < 6,7 kg	Subunidad táctica (lanzamiento manual), operadores individuales	Hasta 60 m	Hasta 5 km (LOS)
	Mini < 15 kg	Subunidad táctica (lanzamiento manual), operadores individuales	Hasta 900 m	Hasta 25 km (LOS)
	Small 15 kg – 150 kg	Unidad táctica (utiliza sistema de lanzamiento)	Hasta 1500 m	50 km (LOS)
Clase II (150 kg – 600 kg)	Táctico	Formación táctica	Hasta 3 km	200 km (LOS)
Clase III (>600 kg)	MALE (<i>Medium Altitude Long Endurance</i>)	Operacional / de Teatro	Hasta 14 km	>200 km (BLOS)
	HALE (<i>High Altitude Long Endurance</i>)	Estratégico	Hasta 19 km	>200 km (BLOS)
	Ataque/Combate	Estratégico/Operacional	Hasta 19 km	>200 km (BLOS)

Tabla 2-1 Clasificación de RPAS (tomada de [15])

También se pueden clasificar según su forma [14]:

- Ala fija: Como si fuese una avioneta, tienen mayor autonomía y producen menos ruido.
- Multirrotores: Aquellos que tienen al menos tres hélices. Son los drones más comunes por su facilidad de uso y precio. Tienen menos autonomía y generan más ruido que los de ala fija.
- Globos y dirigibles: Son silenciosos y tienen gran autonomía, pero su velocidad es muy baja en comparación con ala fija y multirrotores.

Otra manera de clasificar los UAVs es según su método de control [14]:

- Autónomos: No necesitan pilotaje, la ruta se instala previamente, incluyendo altura y velocidad.
- Control remoto: Es pilotado a distancia por un operador.
- Supervisados o monitorizados: Combinación de las anteriores.

En este trabajo, nos centraremos en los RPAS de Clase I, multirrotores y de control remoto, ya que representan una de las categorías más utilizadas en conflictos modernos y por grupos terroristas debido a su alta adaptabilidad, bajo costo y eficacia operativa. Además, este tipo de plataformas, gracias a su reducido tamaño y maniobrabilidad, son ideales para operar en escenarios urbanos y de guerra asimétrica, incluyendo el despliegue en interiores de edificios y otros entornos confinados, centro de estudio de este trabajo. Su capacidad para infiltrarse en áreas de difícil acceso, combinada con su bajo perfil y la dificultad de ser detectados por sistemas convencionales como el radar, los convierte en una herramienta estratégica de gran impacto. Además, su versatilidad permite llevar a cabo misiones de reconocimiento, vigilancia y ataques de precisión, destacándose como una amenaza significativa en operaciones contemporáneas. Esto subraya la necesidad crítica de desarrollar sistemas de detección avanzados que puedan mitigar los riesgos asociados a este tipo de tecnologías.

2.2.1.1 Modelos de drones utilizados en la actualidad.

En el contexto del conflicto en Ucrania, los drones de producción nacional han demostrado ser herramientas clave en el campo de batalla. Entre los modelos más destacados se encuentran el *Baba Yaga* [16], mostrado en la Figura 2-3. Este vehículo autónomo, también bautizado con el nombre de “*Vampire*”, es un dron multirrotor diseñado para operaciones nocturnas de alta precisión. Con una velocidad máxima de 80 km/h, un alcance operativo de 10 km y una autonomía de 23 minutos, podemos encuadrarlo en la categoría Mini de la Tabla 2-1, destacando por su maniobrabilidad en entornos tácticos, incluso en condiciones de baja visibilidad. Equipado con una cámara térmica capaz de detectar objetivos a 120 metros y una capacidad de carga útil de hasta 15 kg, puede transportar municiones como proyectiles de fragmentación, cargas formadas y minas antitanque. Su propulsión silenciosa y tamaño compacto lo hacen ideal para infiltrarse en áreas urbanas y operar en interiores de edificios, ejecutando ataques precisos o misiones de reconocimiento sin ser detectado [17]. Es por ello por lo que se ha convertido en una de las principales amenazas para el ejército ruso.



Figura 2-3 Dron “Baba Yaga” utilizado por las fuerzas ucranianas [17].

Este modelo se alinea con el tipo de dron que los sistemas de detección planteados en este trabajo buscan interceptar. Sus características, como su tamaño reducido en comparación con el resto de los drones, baja detectabilidad y capacidad de operación en entornos urbanos, requieren de un alto nivel de atención y de implementación de sistemas de detección avanzados que sean capaces de identificar y neutralizar amenazas en tiempo real, incluso en escenarios urbanos complejos.

2.2.2 Aplicaciones militares de drones

Tras haber explorado la naturaleza y el impacto de los RPAS en los conflictos modernos, resulta fundamental examinar sus aplicaciones específicas en el campo militar. Desde pequeños drones comerciales utilizados para vigilancia hasta sistemas avanzados con capacidad de ataque de precisión, estos vehículos han ampliado significativamente el alcance y la efectividad de las fuerzas armadas modernas, de tal modo que se puede afirmar que no solo han redefinido la forma en que se llevan a cabo las operaciones, sino que también han ampliado el alcance de las estrategias disponibles, permitiendo a las fuerzas armadas adaptarse a los desafíos de los conflictos contemporáneos. Muestra de ello es que hasta la fecha, los drones del ejército de los Estados Unidos han acumulado más de 375,000 horas de vuelo y cerca de 130,000 salidas en apoyo de operaciones de combate en Irak y Afganistán [18].

En conflictos recientes, como el de Ucrania, los UAVs han sido empleados no solo para reconocimiento y adquisición de objetivos, sino también como plataformas para el lanzamiento de municiones, incluyendo explosivos antitanque y granadas, evidenciando su versatilidad en diversos escenarios [3]. En la Figura 2-4 se puede observar el uso de drones en dicho conflicto.

En el ámbito táctico, los drones pequeños, también conocidos como sUAS (*small Unmanned Aerial Systems*), pertenecientes a las clases I y II de la Tabla 2-1, como se destacó anteriormente, se han consolidado como herramientas indispensables gracias a su bajo costo, facilidad de despliegue y

versatilidad. Durante el conflicto en Ucrania, estos sUAS han sido modificados para transportar municiones y ejecutar ataques dirigidos contra posiciones enemigas, permitiendo a las fuerzas ucranianas neutralizar vehículos blindados y posiciones defensivas rusas de manera eficiente. Además de sus aplicaciones ofensivas, estos drones han sido fundamentales en misiones de vigilancia en tiempo real, proporcionando información clave para evitar emboscadas y ajustar tácticas en el campo de batalla. Entre estas tareas, la de reconocimiento y obtención de información resulta una de las prioritarias por la útil información que pueden proporcionar sin poner en riesgo a personal. Ejemplos de ello pueden ser desde la filmación del portaaviones británico *Queen Elizabeth* en 2017, el aterrizaje en la Casa Blanca o la grabación en la azotea del edificio del primer ministro de Japón, en ninguno de los casos siendo detectados [19]. Por otro lado, su sección transversal de radar reducida también les permite evadir sistemas modernos de defensa aérea, subrayando su efectividad en escenarios contemporáneos [3], [20].



Figura 2-4 Operario ucraniano en escenario de combate manejando dron [21]

En el ámbito operativo, drones de mayor tamaño clasificados en la clase III desempeñan funciones críticas que incluyen ataques estratégicos y operaciones de inteligencia a gran escala [20]. Estas operaciones subrayan la capacidad de los drones para realizar tareas de alta complejidad y complementar a las fuerzas tradicionales. Entre estas, los enjambres de drones han emergido como una evolución tecnológica clave. El *drone swarming* (enjambre de drones) o uso conjunto de drones en formación para una misión, ha llevado a los países occidentales a replantear sus estrategias de defensa aérea [3]. La primera operación realizada por este tipo de formaciones se produjo en 2018 en Siria, donde dos bases rusas fueron atacadas [22]. En la misma línea, en el 2019, los hutíes utilizaron enjambres para atacar instalaciones petroleras en Arabia Saudita, demostrando cómo estas formaciones coordinadas pueden superar las defensas aéreas tradicionales mediante saturación y ataques precisos [3]. Además, estas técnicas pueden combinarse con metodologías de ataque tradicionales, tal y como se demostró en la demostración “*Perdix*” en Estados Unidos donde tres cazas F-18 actuaron simultáneamente con 103 drones [3].

Otro concepto recientemente incorporado a las principales “guerras” es la guerra de información o IW (*Information Warfare*), definida por el Ejército de los Estados Unidos como las “*acciones llevadas a cabo para el logro de la superioridad de la información, afectando la información, los procesos basados en la información y los sistemas de información adversarios, mientras se protege la información, los procesos basados en la información y los sistemas de información propios*”. El uso de drones en este tipo de “guerra” no solo permite influir en la percepción pública del conflicto, sino que también resalta el valor estratégico de estas plataformas en la creación de narrativas militares, tal y como se ha demostrado en conflictos recientes en los que se muestran vídeos de los UAVs en acción. La centralidad de estos en las operaciones IW demuestra su impacto más allá del campo de batalla físico, extendiéndose al ámbito psicológico y mediático [3].

De esta manera, a lo largo de este apartado hemos analizado en detalle las múltiples aplicaciones de los drones y cómo estas influyen directamente en el transcurso de los conflictos modernos. Desde sus

capacidades en reconocimiento, ataque y apoyo logístico, hasta su impacto en la guerra de información, los UAVs han demostrado ser una herramienta versátil y esencial en el panorama militar contemporáneo. En este análisis también se ha evidenciado que los entornos urbanos, con sus características particulares, representan un desafío único para las operaciones con drones. Las ciudades y su infraestructura, definidas por factores geográficos y urbanísticos, afectan significativamente el modo en que las unidades operan en estos espacios [23].

Originalmente, los drones se empleaban para misiones de observación gracias a su capacidad de proporcionar imágenes en tiempo real. Sin embargo, su uso ha evolucionado hacia una amplia variedad de actividades, incluyendo el seguimiento de objetivos, la entrega de suministros médicos y la identificación precisa de amenazas sin exponer a las tropas al peligro inmediato. En este contexto, el uso de drones en interiores y edificios cobra especial relevancia. Los drones más pequeños, equipados con tecnología avanzada como cámaras térmicas, sensores y micrófonos, han demostrado ser altamente eficaces en entornos confinados, donde su maniobrabilidad y tamaño reducido les permiten sortear obstáculos y operar de manera eficiente [23].

Características como el tipo de alas, el nivel de autonomía, las dimensiones y el peso son factores clave que determinan la idoneidad de un dron para operaciones urbanas. En particular, los drones con alas rotatorias y menor tamaño han probado ser los más adecuados para estas misiones debido a su capacidad de despegue y aterrizaje en espacios reducidos, así como su facilidad para maniobrar en entornos con múltiples obstáculos. Por el contrario, los drones de mayor tamaño y con alas fijas resultan menos prácticos en estos escenarios [23].

El análisis de las características de los drones destaca la necesidad de contar con sistemas de detección avanzados que sean capaces de identificar y contrarrestar estas amenazas en entornos urbanos. La proliferación de drones, combinada con sus capacidades avanzadas y la reducción de su coste, subraya la urgencia de desarrollar tecnologías específicas para su detección. Este escenario plantea un desafío importante: diseñar soluciones eficaces que permitan detectar y responder a la amenaza que representan estos dispositivos en operaciones urbanas, asegurando la protección de infraestructuras y personal involucrado en estas misiones.

2.2.3 Fases para hacer frente a un dron

Para hacer frente a una amenaza de un UAV se definen las siguientes fases [14]. Podemos observarlas también en la Figura 2-5:

1. Detección: descubrimiento de una fuente que podría tratarse de un dron.
2. Identificación: Discernimiento de si es un dron o no.
3. Seguimiento: Dirección, altura, velocidad e intenciones.
4. Neutralización: Eliminación de la amenaza.
5. Inteligencia: Recopilación de datos para que el operador pueda tomar decisiones en el momento.

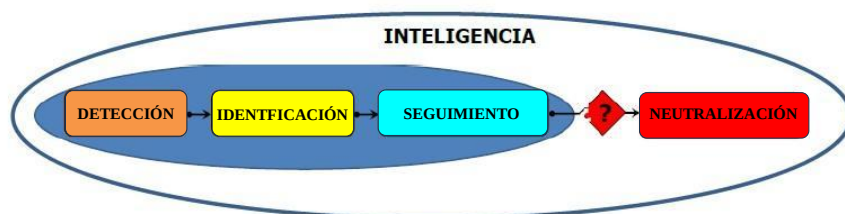


Figura 2-5 Fases para hacer frente a amenaza de UAV [14].

En el presente trabajo los esfuerzos van dirigidos a la fase de detección e identificación de la amenaza, es decir, en el descubrimiento de indicios de la presencia de un dron y la posterior identificación de que efectivamente se trata de un UAV.

2.3 Sistemas de detección

Como acabamos de mencionar, este trabajo se centra en el estudio de la fase de detección e identificación de un UAV. Para ello, es necesario comprender los principios básicos de estos conceptos.

2.3.1 Definición de un sistema de detección

Los sistemas de detección son tecnologías integrales diseñadas para identificar y medir diversos fenómenos físicos, utilizando principios fundamentales como las ondas sonoras, las ondas radioeléctricas y la luz. Estos sistemas sirven para propósitos variados, como el monitoreo de la integridad estructural, la detección de señales de interés y el análisis de características superficiales. Además, están diseñados para captar cambios en objetos o entornos físicos mediante sensores, unidades de procesamiento de señales y mecanismos de salida que trabajan en conjunto para identificar condiciones específicas o anomalías [24].

2.3.2 Componentes

Los sistemas de detección constan generalmente de tres componentes, mostrados también en la Figura 2-6:

- Sensores y unidades de adquisición: Estos componentes recogen datos del entorno, como imágenes en sistemas de vigilancia, cambios físicos en objetos, o ruido detectado.
- Unidades de procesamiento: Analizan los datos adquiridos, comparándolos con criterios predefinidos para identificar cambios significativos o acciones necesarias.
- Mecanismos de salida: Una vez que se realiza una detección, los sistemas suelen incluir mecanismos de notificación que alertan a los usuarios sobre anomalías o muestran información relevante, como marcadores de posición en sistemas de monitoreo.

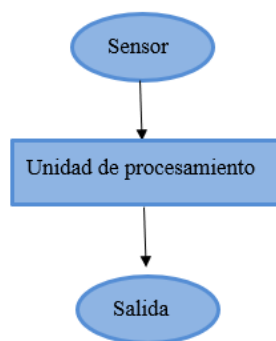


Figura 2-6 Componentes y proceso de detección.

2.3.3 Reseña histórica sobre los sistemas de detección de drones

Desde los primeros pasos en el desarrollo del arma aérea, analizados anteriormente, la detección de aeronaves se consolidó como un elemento crucial en la planificación y ejecución de estrategias de defensa aérea. Antes de la introducción del radar, se utilizaban técnicas de detección pasiva basadas en el sonido, aprovechando el ruido generado por los motores de los aviones. Una de las soluciones más destacadas fue el uso de grandes dispositivos acústicos o “bocinas”, que ofrecían una ganancia significativa y una mayor direccionalidad. Este diseño, cuyo diámetro era mayor comparado con los oídos humanos, permitía a los observadores localizar con mayor precisión la dirección de un sonido [1]. En la Figura 2-7 podemos observar un sistema de “cuernos” diseñado por el Ejército de Estados Unidos.



Figura 2-7 Sistema de detección de dos “bocinas” en Boiling Field, Estados Unidos, 1921 [1]

Estas técnicas tenían la ventaja de observar alrededor de esquinas y sobre colinas gracias a la refracción del sonido, lo que resultaba especialmente útil en terrenos irregulares. Los instrumentos de defensa aérea de la época solían consistir en bocinas acústicas o micrófonos conectados a los oídos de los operadores mediante tubos, similares a un gran estetoscopio. En paralelo, Reino Unido, principal promotor de estas innovaciones de detección, planteó una red de espejos acústicos. Estos dispositivos funcionaban posicionando micrófonos móviles para encontrar el ángulo que maximizara la amplitud del sonido recibido ([1]), capturando ondas sonoras distantes y concentrándolas en un solo punto, lo que también permitía determinar la demora del objetivo. Posteriormente, un técnico descifraba los sonidos y determinaba la presencia de aeronaves que se aproximaban. Al comparar las diferencias de sonido y presión generadas por el paso de proyectiles y los sonidos de los cañones que los disparaban, era posible localizar la posición exacta de las armas enemigas, y por tanto, de las aeronaves que las portaban. Esta técnica se utilizó para detectar bombarderos alemanes, lo que marcó un avance significativo en la detección aérea [25]. En la Figura 2-8 podemos observar uno de estos espejos acústicos.

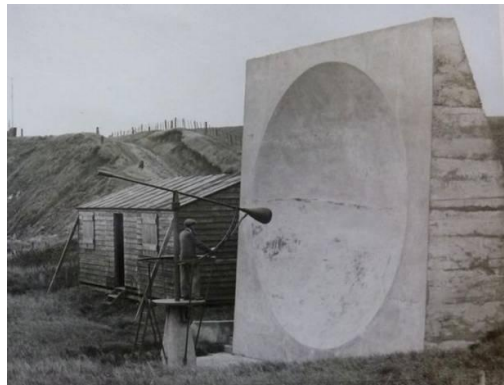


Figura 2-8 Espejo acústico desarrollado por Reino Unido [25].

Cuando se analizan estos sistemas desde la perspectiva moderna, es posible identificar cómo sus componentes pueden alinearse con los elementos básicos de un sistema de detección. Las “bocinas” y espejos acústicos funcionaban como sensores y unidades de adquisición, capturando datos del entorno en forma de sonido. La triangulación y el análisis de la amplitud sonora realizados por los operadores equivalen a las unidades de procesamiento, que interpretaban y comparaban la información para localizar aeronaves. Finalmente, los mecanismos como las señales visuales o verbales proporcionadas por los observadores cumplían el rol de mecanismos de salida, alertando sobre la presencia y ubicación del objetivo.

Sin embargo, estas tecnologías fueron progresivamente reemplazadas antes y durante la Segunda Guerra Mundial por el radar, que ofrecía una mayor eficacia [1]. A pesar de ello, durante el conflicto, se sugirió que los motores de los bombarderos pesados alemanes se mantenían desincronizados intencionadamente para dificultar su localización mediante métodos acústicos [1]. Además, estas

innovaciones desarrolladas por varios países establecieron los cimientos para el desarrollo de sistemas de detección modernos.

En el contexto actual, aunque el radar ha demostrado ser altamente efectivo, su utilidad en entornos confinados, como edificios o zonas urbanas densas, es limitada. Además, los UAVs presenta una sección transversal radar o RCS (*Radar Cross Section*) significativamente menor en comparación con aeronaves tripuladas convencionales. Esta característica se debe a su tamaño reducido, diseño y materiales utilizados en su construcción, lo que resulta en una menor reflectividad de las señales de radar. Como consecuencia, los radares tradicionales, optimizados para detectar objetos con RCS más grandes, enfrentan dificultades para identificar y rastrear eficazmente a los drones [26]. Por ello, los principios subyacentes de la detección acústica reaparecen como una solución viable para la identificación y localización de drones en estos entornos.

2.4 Detección de UAVs

En la era contemporánea, los UAVs presentan desafíos específicos que requieren una adaptación estratégica de los sistemas de detección. Si bien en el apartado anterior se discutió cómo los radares tradicionales son ineficaces en entornos urbanos y confinados, este apartado aborda cómo la proliferación de sUAS ha intensificado la necesidad de sistemas especializados que superen las limitaciones actuales.

La proliferación de drones, especialmente los pertenecientes a la clase I, ha puesto de manifiesto su capacidad para evadir las tecnologías de defensa convencionales debido a su baja firma de radar y su maniobrabilidad impredecible [3]. Además, el aumento del uso de enjambres y drones autónomos, que operan sin señales de radiofrecuencia, subraya la necesidad de enfoques avanzados y multifacéticos para su detección y neutralización.

En este contexto, se destacan las tecnologías emergentes como los sensores acústicos, cámaras de alta resolución y sistemas de guerra electrónica. Estas herramientas están diseñadas para superar las limitaciones inherentes de los radares tradicionales y ofrecer una solución viable para entornos complejos y urbanos. Por ejemplo, los sensores acústicos pueden detectar el ruido característico de las hélices de los drones, mientras que las cámaras térmicas y visibles permiten identificar movimientos o firmas de calor en tiempo real. Además, la integración de múltiples tecnologías en sistemas combinados mejora significativamente la precisión y eficacia de la detección.

Con el avance continuo de la inteligencia artificial y el aprendizaje automático, estas tecnologías están evolucionando para adaptarse rápidamente a las amenazas emergentes, analizando grandes volúmenes de datos en tiempo real y detectando patrones anómalos. Esta evolución tecnológica marca el camino hacia soluciones más robustas y eficaces para contrarrestar la amenaza creciente que representan los UAVs, especialmente en entornos urbanos y confinados donde las tecnologías tradicionales son insuficientes.

2.4.1 Métodos de detección actuales

Antes de entrar en detalles con los sistemas de detección actuales debemos hacer un apunte con la diferenciación entre sistemas activos y pasivos para poder comprender posteriormente cómo funciona cada mecanismo.

Los sistemas de detección activos y pasivos representan los dos enfoques principales en la identificación de drones y otros objetos aéreos. Los sistemas activos, como el radar tradicional, operan enviando señales radioeléctricas y analizando su reflejo para determinar la posición, dirección y distancia de un objeto. Estos sistemas, son eficaces en entornos abiertos, permitiendo una detección a largas distancias y el seguimiento de múltiples objetivos simultáneamente. Sin embargo, presentan limitaciones significativas en entornos urbanos debido a la interferencia causada por el *clutter*, es decir, ecos no deseados generados por edificios, vegetación y otros elementos del entorno. Además, los drones

pequeños presentan una baja reflectividad, lo que dificulta su detección precisa en estas condiciones [14].

Por su parte, los sistemas pasivos utilizan señales existentes emitidas por el blanco o por otros dispositivos, como estaciones de TV o transmisores de RF, en lugar de emitir sus propias señales. Esto los convierte en soluciones más económicas y fáciles de desplegar, con una buena cobertura a bajas altitudes. Sin embargo, su efectividad frente a drones comerciales es limitada, ya que su capacidad para detectar objetos depende del tamaño del dron, necesitando que el diámetro sea mayor a 120 cm para garantizar su identificación. Aunque estos sistemas son prometedores en términos de ahorro de energía y reducción de emisiones, aún presentan desafíos en escenarios donde los drones son extremadamente pequeños o voladores a baja altura [14].

Para comprender cómo los sistemas de detección abordan estas limitaciones, es importante explorar las tecnologías específicas que actualmente se emplean. A continuación, se describen los principales métodos de detección utilizados en el contexto de UAVs. Estos son, detección acústica, óptica, por radiofrecuencia y multisensorial. Un resumen de las técnicas lo encontramos en la Figura 2-9.

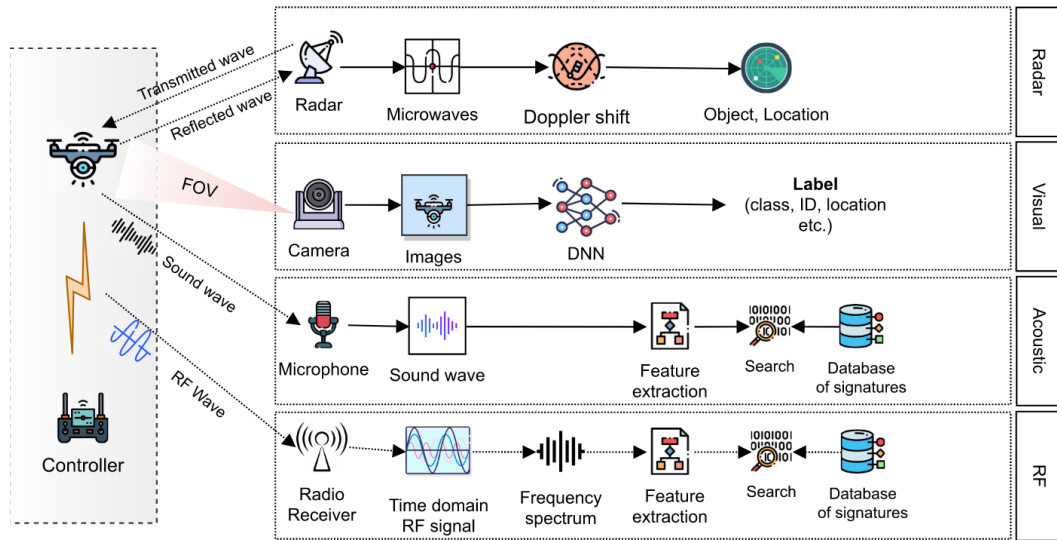


Figura 2-9 Técnicas de detección de drones [27]

2.4.2 Detección acústica

La detección acústica identifica drones a través de las características únicas de los sonidos producidos por sus rotores. Este método resulta útil en entornos con baja visibilidad o como complemento a otras técnicas, pero su alcance puede ser limitado en ambientes urbanos debido al ruido ambiental [14], [28].

En investigaciones recientes, se han utilizado técnicas avanzadas como Coeficientes Cepstrales de Frecuencias de Mel o MFCC (*Mel Frequency Cepstrum Coefficients*) y clasificadores basados en *Long Short-Term Memory* o LSTM, mostrando una mejora en la precisión de detección y una reducción de falsas alarmas [28], [29]. Estas técnicas serán desarrolladas en mayor detalle posteriormente al centrarnos en las técnicas de detección acústica. Aunque ha de destacarse con anterioridad que el rango práctico de este tipo de sistemas puede ser reducido, como se demuestra en sistemas con detección eficaz hasta los 35-45 metros, aunque configuraciones más complejas han alcanzado hasta 290 metros [29].

No obstante, el centro de estudio de este trabajo es la detección en interiores, donde las distancias de operación se asumen reducidas, por lo que esto no supondrá una limitación para su desarrollo.

2.4.3 Detección óptica

Los sistemas de detección óptica utilizan cámaras de espectro visible e infrarrojo térmico (IR) para identificar drones. Su uso es bastante efectivo en la fase de identificación, es decir, en determinar si se

trata de un dron o no lo que se ha detectado. Las cámaras en el espectro visible son útiles para el reconocimiento visual, pero pueden ser menos efectivas bajo condiciones de baja iluminación o largas distancias. Las cámaras térmicas, por otro lado, detectan la radiación infrarroja generada por los drones, como el calor de los rotores o las baterías. En primer lugar, se estudia la huella térmica que dejan los UAVs, denominada “*huella dubitada*” y posteriormente esta se compara con otras previamente estudiadas y almacenadas llamadas “*huellas indubitadas*”, lo que permite identificaciones bastante precisas. Sin embargo, debe destacarse que las cámaras IR enfrentan desafíos en condiciones meteorológicas adversas, como niebla o lluvia [14].

2.4.4 Detección por radiofrecuencia

La detección por radiofrecuencia o RF (*Radiofrequency*) utiliza las señales emitidas entre los drones y sus controladores para identificarlos. La empresa tecnológica DJI (*Da Jiang Innovations*) en el 2017 desarrolla el sistema Aeroscope, un sistema que permite la detección precisa de drones comerciales al decodificar las tramas de comunicación, proporcionando datos como la ubicación, velocidad y modelo del dron. Además, la mayoría de los sistemas identifican las direcciones MAC del dron y de su controlador, ya que muchos de los drones comerciales utilizan WiFi para comunicarse [14].

Este enfoque es altamente efectivo a largas distancias y es relativamente económico, pero depende de que exista una comunicación activa entre el dron y su controlador, ya sea por señales radioeléctricas o por fibra óptica. Además, investigaciones recientes han explorado la detección de drones mediante “huellas” únicas en el espectro de RF, lo que permite clasificaciones más avanzadas [29]. Estas huellas, conocidas también como *RF fingerprints*, capturan las características intrínsecas de las señales emitidas por los drones y sus controladores, como el espectro de frecuencia y las transiciones de amplitud. Estas características son procesadas mediante algoritmos de aprendizaje automático, que analizan y clasifican las señales con base en patrones únicos, lo que permite una clasificación precisa incluso en entornos con interferencias como WiFi y Bluetooth [30].

2.4.5 Detección multisensorial

Por último, la fusión de datos de múltiples sensores es una estrategia clave para superar las limitaciones individuales de cada tecnología. Estas combinaciones integran sensores térmicos, visibles, acústicos y de RF para mejorar la detección y reducir falsas alarmas. Los sensores térmicos permiten identificar drones basándose en su huella de calor, siendo útiles incluso en condiciones de baja visibilidad, aunque enfrentan limitaciones ante condiciones climáticas adversas. Los sensores visibles proporcionan datos detallados para el reconocimiento visual, pero su eficacia disminuye a mayores distancias. Los sensores acústicos capturan el sonido característico de los rotores de los drones, destacando en entornos de baja visibilidad, aunque son sensibles al ruido ambiental y tienen un alcance limitado. Por último, los sensores de RF detectan las señales de comunicación de los drones y sus controladores, siendo especialmente efectivos para largas distancias, pero dependientes de la actividad comunicativa del dron [29], [31].

Estos sistemas emplean plataformas giratorias y algoritmos de aprendizaje automático para analizar simultáneamente datos de varios sensores, lo que permite una detección y seguimiento más robustos en tiempo real. La combinación de cámaras térmicas y visibles junto con micrófonos direccionales y radares ha demostrado ser particularmente eficaz para identificar drones en una variedad de escenarios [29], [31].

En la Tabla 2-2 se puede observar un resumen de las ventajas y limitaciones de cada uno de los sistemas descritos en este apartado:

Sistema	Distancia Detección	Información Aportada	Evita Obstáculos	Entorno Urbano	Entorno No urbano	Calidad- Precio	Observación
Acústico (pasivo)	Escasa	Alta	Muy bien	Regular	Bien	Alto	Muy corta distancia
Óptico (pasivo)	Escasa - media	Alta	Mal	Regular	Bien	Media	Bueno para identificar, malo para detectar
Radar (activo)	Alta	Alta	Regular	Mal	Muy bien	Alto	Entornos no urbanos
RF (pasivo)	Alta	Muy alta	Regular	Bien	Muy bien	Escasa	Mejor opción para entornos no urbanos

Tabla 2-2 Características de los sistemas de detección (tomada de [14])

Para este estudio, se ha optado por el uso de sistemas de detección acústica debido a sus ventajas específicas dentro de entornos cerrados. Aunque este tipo de detección tiene un alcance limitado en espacios abiertos, esta restricción es irrelevante en escenarios urbanos o interiores, donde la proximidad entre el dron y los sensores es suficiente para una detección efectiva. Además, este método ofrece una gran cantidad de información relevante para la identificación y clasificación de drones, aunque su desempeño puede verse afectado por el ruido ambiental en entornos de guerra o urbanos densamente poblados.

Por otro lado, la detección acústica presenta la ventaja de no depender de la línea de visión (LOS), lo que permite identificar drones incluso cuando se encuentran detrás de obstáculo o en entornos donde la luz es escasa. Finalmente, su bajo costo y alta efectividad lo convierten en una opción viable, como ha quedado demostrado en conflictos recientes, como el de Ucrania, donde estos sistemas han sido empleados con éxito.

2.5 Técnicas y sensores empleados en la detección sonora de UAVs

Una vez seleccionada la principal técnica que se va a utilizar, en este apartado se detallará el modo en el que va a ser empleada de manera que tenga un alto grado de efectividad y un reducido número de errores. Para ello, se describirán en primer lugar los principios básicos del sonido aplicados a la detección acústica, para posteriormente entrar en detalle con la obtención de datos, su procesamiento, las técnicas de detección y clasificación y finalmente la salida de dicho sonido.

2.5.1 Principios básicos de la detección acústica de drones

El sonido es una perturbación mecánica que se propaga a través de un medio elástico, como el aire, en forma de ondas longitudinales. Estas ondas se generan debido a las variaciones de presión provocadas por una fuente vibratoria, lo que da lugar a zonas alternas de compresión y expansión en el medio circundante. En términos de propagación, el sonido se transmite esféricamente desde su fuente y puede ser captado por sensores como micrófonos, los cuales convierten las ondas mecánicas en señales eléctricas procesables [32]. Además, su propagación está determinada por parámetros como la frecuencia, la amplitud y características del medio que afectan a la velocidad de propagación.

Los drones centro de este estudio, los cuadricópteros (drones multirrotores), generan un espectro acústico característico derivado de la interacción del aire con sus hélices en rotación y de las vibraciones de sus motores eléctricos. Este ruido es el resultado de la combinación de una frecuencia fundamental o *pitch*, relacionada con la velocidad de giro de los rotores, y de armónicos adicionales producidos por turbulencias en el flujo de aire. Dependiendo de la velocidad de los motores y la carga transportada, estas frecuencias pueden variar, situándose normalmente en el rango de los cientos de hercios hasta varios kilohercios [32].

Por otro lado, debe destacarse que el sonido generado por los drones es una onda no estacionaria, lo que significa que su comportamiento varía en el tiempo [32]. Este carácter no estacionario se debe a los constantes ajustes en la velocidad de rotación de los motores del dron, que dependen de factores como su desplazamiento, la dirección del viento y las maniobras realizadas. En ciertos intervalos de tiempo, cuando los rotores mantienen una velocidad estable, es posible identificar características de periodicidad en la señal, concepto conocido como “pseudo-periódico” o “aperiódico” [32]. Sin embargo, en situaciones de cambio de velocidad, la señal se vuelve altamente variable, dificultando su modelado como una señal periódica convencional.

Debido a esta naturaleza no estacionaria de las ondas generadas por el dron, un análisis en el dominio del tiempo no es suficiente para su correcta identificación. Para ello, se pasa al dominio frecuencial, siendo posible de esta manera conocer la contribución de cada frecuencia a la señal total, lo que permite distinguir las componentes más relevantes para la detección. Este enfoque facilita la obtención de parámetros clave como la frecuencia fundamental del dron, su timbre (definido por la cantidad y amplitud de los armónicos) y su altura tonal (diferenciando sonidos agudos y graves). Gracias a este análisis, es posible clasificar y detectar drones de manera más eficaz [32].

Para llevar a cabo este análisis, primero es necesario obtener las señales acústicas con la mejor calidad posible. En la siguiente sección, se describen los sensores utilizados y su disposición en los sistemas de detección acústica.

2.5.2 Adquisición de datos

Para la detección acústica de drones, es fundamental la correcta captura de los sonidos generados por sus motores y hélices. Esto se logra a través del uso de diversos sensores acústicos con características y disposiciones específicas que influyen en la calidad y precisión del proceso de detección.

Pero antes de entrar en detalle con los sensores utilizados, es necesario saber qué factores influyen en la captura del sonido, aumentando o disminuyendo la calidad con la que dicho sensor recibe la información:

- Distancia entre el dron y el sensor: A mayor distancia, mayor atenuación del sonido debido a la dispersión geométrica y la absorción atmosférica.
- Ruido ambiental: Fuentes de ruido externas, como disparos o explosiones (previsibles en un entorno de guerra) pueden reducir la relación señal-ruido o SNR (*Signal to Noise Ratio*) afectando a la detección efectiva.
- Interferencias y ecos: Superficies reflectantes generan ecos que pueden distorsionar la señal acústica, dificultando la identificación precisa de la firma acústica del dron. No obstante, el centro de estudio no es la identificación de un dron determinado si no la mera presencia de uno, por lo que este aspecto no influirá.
- Condiciones meteorológicas: Factores como el viento, humedad o temperatura pueden alterar la propagación del sonido, afectando a la captación de los sensores. En nuestro caso el viento tendrá menor impacto debido a que el estudio se realiza en interiores.
- Ubicación y orientación de los sensores: La colocación estratégica de micrófonos en entornos urbanos o abiertos influye en la capacidad del sistema para capturar sonidos de drones sin contaminación acústica externa.

- Frecuencia de operación de los sensores: La respuesta en frecuencia de los micrófonos y su sensibilidad a distintas bandas de frecuencia afecta a la eficacia en la detección de drones con distintos perfiles acústicos.

2.5.2.1 Tipos de sensores utilizados

El principal sensor utilizado para la detección de un sonido es el micrófono. Como se ha mencionado anteriormente, este es el que se encarga de transformar las ondas mecánicas longitudinales de la fuente sonora en ondas eléctricas que puedan ser posteriormente procesadas. A continuación, se describen las principales técnicas de uso de micrófonos.

- Micrófonos individuales: Capturan sonido en un solo punto de medición y son útiles para análisis simples o en entornos controlados. Sin embargo, presentan limitaciones para la localización del dron y la reducción del ruido ambiental [27].
- Arrays de micrófonos: Conjunto de micrófonos dispuestos en configuraciones geométricas específicas que mejoran la precisión en la detección y localización. Permiten estimar la dirección de llegada de la señal acústica o DOA (*Direction of Arrival*) y aplicar técnicas de *beamforming* para reducir interferencias [32], [33]. En la Figura 2-10 podemos observar un array de micrófonos.



Figura 2-10 Ejemplo de array de micrófonos [34]

- Micrófonos MEMS (*Microelectromechanical Systems*): Son micrófonos de pequeño tamaño que integran un sistema de conversión analógica-digital. Se caracterizan por su bajo consumo energético y alta sensibilidad, lo que los hace adecuados para sistemas embebidos y dispositivos portátiles. Su uso ha crecido en aplicaciones de detección acústica debido a su capacidad para captar sonidos en entornos con alto nivel de ruido [35].
- Micrófonos de condensador: Presentan una alta respuesta en frecuencia y son utilizados en entornos donde se requiere captación de señales débiles con alta fidelidad [34].

2.5.2.2 Distintas configuraciones de sensores

Antes veíamos cómo la ubicación y orientación de los micrófonos influye en su capacidad de captar el sonido. A continuación, veremos distintas formas de abordar este desafío:

- Arrays lineales o ULA (*Uniform Lineal Array*): Los micrófonos están alineados en una sola dimensión, proporcionando buena resolución en el plano horizontal o vertical, pero sin capacidad de localización tridimensional [32].
- Arrays circulares o UCA (*Uniform Circular Array*): Disposición en forma de círculo que permite detección omnidireccional (misma directividad en todas las direcciones), mejorando la estimación de la posición del dron sin necesidad de una orientación previa [32], [34]. La Figura 2-10 es un ejemplo de UCA.
- Arrays rectangulares o URA (*Uniform Rectangular Array*): Disposición en rejilla que permite obtener información tanto en azimut como en elevación, facilitando la localización tridimensional del UAV [32].

- Arrays irregulares y redes distribuidas: Combinaciones no uniformes de micrófonos que pueden mejorar la detección en entornos complejos, aunque requieren procesamiento avanzado para extraer información espacial precisa [33].

2.5.3 Procesamiento de datos

Una vez captados los datos mediante sensores adecuados, estos deben ser analizados para extraer información relevante que permita la identificación y clasificación del dron. Este proceso incluye la transformación de la señal en distintos dominios, la reducción de ruido e interferencias, así como la extracción de características acústicas relevantes

Las señales acústicas pueden analizarse en el dominio temporal o en el dominio frecuencial. En el dominio del tiempo, se observa la evolución de la amplitud de la señal a lo largo del tiempo, pero esta representación no es suficiente para identificar patrones específicos de los drones debido a la complejidad y variabilidad de su firma acústica. En la Figura 2-11 no se puede observar un *pitch* claro para poder identificar la firma acústica. Por ello, se requiere un análisis en el dominio de la frecuencia, donde se descompone la señal en sus componentes espectrales para identificar las frecuencias predominantes y sus armónicos [32], [34].

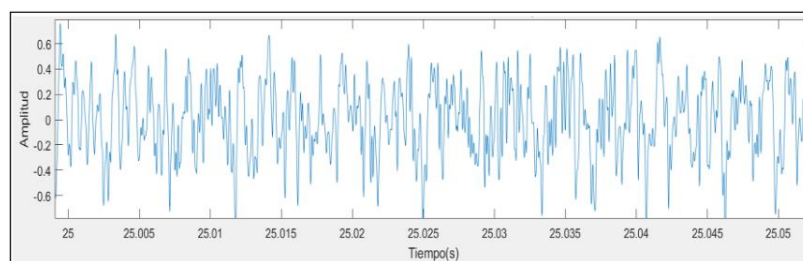


Figura 2-11 Análisis en el dominio del tiempo de un dron [32]

Para pasar del dominio temporal al dominio en frecuencia se emplea la transformada de Fourier o FT (*Fourier Transform*). Esta técnica permite descomponer una señal en una combinación de ondas sinusoidales de diferentes frecuencias [32]. Las dos variantes más utilizadas en procesamiento de señales acústicas son la Transformada de Fourier Rápida o FFT (*Fast Fourier Transform*) y la Transformada de Fourier de Tiempo Corto o STFT (*Short-Time Fourier Transform*).

La FFT es un algoritmo optimizado para calcular la Transformada de Fourier Discreta (DFT) de manera eficiente, reduciendo significativamente el tiempo de cálculo [34]. Es especialmente útil cuando se analiza una señal estacionaria, es decir, cuando sus características espectrales no varían en el tiempo [36].

Por su parte, la STFT se utiliza para el análisis de señales no estacionarias, como las generadas por los drones, ya que permite examinar cómo cambia el espectro de frecuencia a lo largo del tiempo mediante el uso de ventanas deslizantes. Esta técnica proporciona información temporal y frecuencial simultáneamente, aunque introduce una compensación entre resolución en el tiempo y en la frecuencia [34], [37].

La FT, en su forma convencional, no permite analizar cómo varían las frecuencias en el tiempo, ya que descompone la señal en sus componentes frecuenciales sin preservar la evolución temporal [32]. Para solucionar esto, se emplea la STFT, que divide la señal en segmentos de corta duración donde las características de la señal pueden considerarse estacionarias. A cada uno de estos segmentos se le aplica la FFT, obteniendo así la evolución espectral en función del tiempo [32], [34], [36]. En la Figura 2-12 se puede observar este proceso.

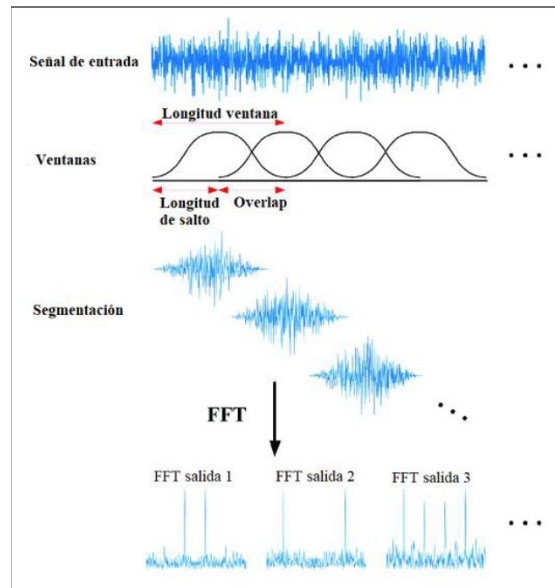


Figura 2-12 Uso de la técnica de STFT [32]

2.5.4 Extracción y representación de las características acústicas

Una vez procesada la información debemos tener la capacidad de extraer sus características principales.

Para ello, la información se representa en un espectrograma, una herramienta que permite visualizar cómo cambian las frecuencias de la señal a lo largo del tiempo. En la detección de drones, el espectrograma es clave para identificar patrones acústicos característicos, como la frecuencia fundamental y sus armónicos, diferenciándolos del ruido ambiental y otras señales no deseadas [38].

En la Figura 2-13 se ilustra un ejemplo tanto del espectro de frecuencia, como del espectro tiempo-frecuencia de las señales acústicas generadas por un UAV y ruido ambiental. Las subfiguras de arriba tratan del sonido producido por los drones, mientras que las de abajo representan el sonido ambiente. En las imágenes de la derecha se pueden observar variaciones en la intensidad de las señales acústicas representadas por distintos colores. Al analizar las de la izquierda, se identifica que una de las principales diferencias entre el sonido de los drones y el ruido de fondo es la presencia de armónicos y frecuencia fundamental (*pitch*), lo que indica que estas características pueden ser aprovechadas como un criterio para la detección de drones, de ahí la importancia de los espectrogramas [39].

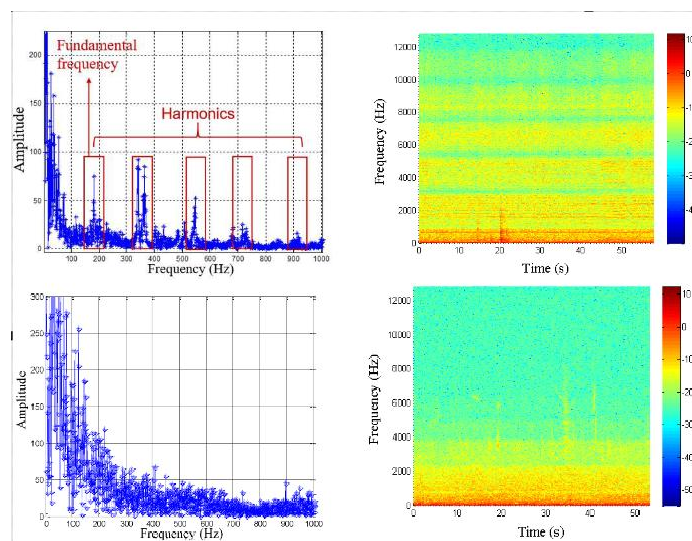


Figura 2-13 Ejemplo de espectrograma de sonido de un dron y de sonido ambiente [39]

Otras técnicas de extracción de datos sonoros son los Coeficientes Cepstrales de la Frecuencia de Mel o MFCC (*Mel-Frequency Cepstral Coefficients*) y los Modelos Ocultos de Markov o HMM (*Markov Hidden Models*).

2.5.4.1 MFCC

Los MFCC son una técnica de extracción de características ampliamente utilizada en el reconocimiento de patrones acústicos, ya que modelan la percepción humana del sonido. Estos coeficientes se calculan aplicando una transformación de la señal al dominio cepstral después de pasar por una escala de frecuencia basada en la percepción auditiva, conocida como la escala de Mel. El proceso de obtención de los MFCC implica la aplicación de la Transformada de Fourier para obtener el espectro de potencia de la señal, seguido de un filtrado con un banco de filtros de Mel y, finalmente, una Transformada Discreta del Coseno o DCT (*Discrete Cosine Transform*) para reducir la dimensionalidad de los datos. Estos coeficientes permiten representar la envolvente espectral de la señal de manera compacta [38]. En la Figura 2-14 se puede observar un esquema del proceso de obtención.

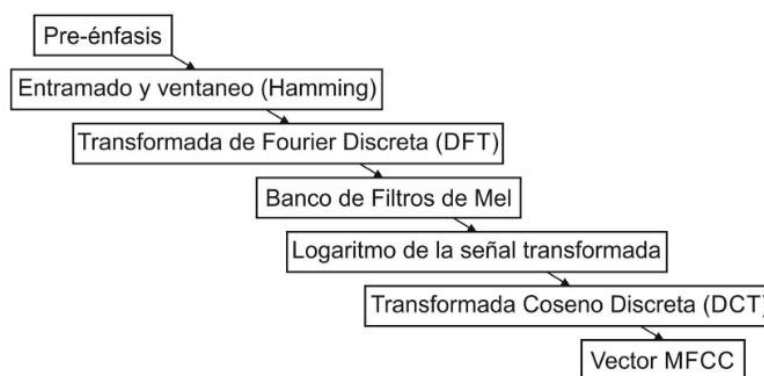


Figura 2-14 Proceso de obtención de los MFCC [38]

2.5.4.2 HMM

El funcionamiento de un HMM se basa en la representación de la señal como una secuencia de estados ocultos, donde cada estado tiene una probabilidad de transición y una distribución de emisión asociada. Esto permite que el modelo aprenda la relación entre los cambios en la señal de audio y las posibles clasificaciones de drones. Una de las ventajas de los HMM es su capacidad para manejar señales con variabilidad temporal y ruido ambiental, lo que los hace adecuados para entornos no controlados [40].

La combinación de MFCC y HMM es una de las estrategias más efectivas en la detección acústica de drones. Mientras que los MFCC transforman la señal de audio en un conjunto de características que reflejan su estructura espectral, los HMM modelan cómo evoluciona esta señal en el tiempo. De esta manera, los MFCC proporcionan una representación eficiente de la envolvente espectral del dron, mientras que los HMM analizan cómo cambia esta información con el tiempo, facilitando la clasificación de las señales incluso en entornos con ruido o variabilidad acústica [38], [40].

2.5.5 Técnicas de identificación

Anteriormente, en la Figura 2-5 definíamos la fase de identificación como el proceso de saber si un ruido existente se trata de un dron o no. En este apartado vamos a desarrollar las diferentes técnicas que existen para abordar esta fase.

2.5.5.1 Correlación de patrones e identificación de armónicos

La detección acústica de drones puede realizarse mediante la correlación de patrones entre una señal captada y una base de datos de firmas acústicas previamente registradas. Este método es eficaz cuando

se dispone de un conjunto de referencias bien definido, permitiendo comparar la señal en tiempo real con muestras almacenadas [34].

Otro enfoque común es la identificación de armónicos, donde se analizan las frecuencias fundamentales generadas por los rotores del dron y sus múltiplos enteros. Los drones de tipo cuadricóptero, por ejemplo, generan un espectro acústico característico debido a la interacción de sus hélices con el aire. La detección de estas frecuencias permite diferenciar un dron de otras fuentes sonoras ambientales [27].

2.5.5.2 Modelos estadísticos

Los modelos estadísticos han sido empleados en la detección de drones debido a su capacidad para modelar la variabilidad de las señales acústicas en distintos entornos. Un ejemplo de estos métodos es el Modelo de Mezcla Gaussiana o GMM (*Gaussian Mixture Model*).

Los GMM son modelos estadísticos utilizados para representar distribuciones de datos complejas mediante la combinación de múltiples funciones de densidad de probabilidad gaussianas. En la detección acústica de drones, los GMM se emplean para modelar la distribución probabilística de las características extraídas de las señales acústicas, facilitando la distinción entre drones y ruido ambiental [41].

El funcionamiento de un GMM se basa en la estimación de parámetros que describen cada una de las distribuciones gaussianas involucradas. Mediante el algoritmo EM o *Expectation-Maximization*, se optimizan estos parámetros para mejorar la precisión del modelo en la clasificación de señales acústicas. Además, la técnica de GMM Supervectors permite representar la señal en un espacio de características más robusto, facilitando la detección de patrones en entornos con alta variabilidad sonora [41].

Otro método estadístico utilizado es el Filtro Bayesiano, basado en el Teorema de Bayes. Este método se utiliza para mejorar la precisión en la estimación de la ubicación del dron al integrar mediciones actuales con estimaciones previas. Permite actualizar la posición estimada del dron de manera continua, reduciendo la incertidumbre generada por ecos y ruidos en el ambiente [39].

2.5.5.3 Aprendizaje máquina o Machine Learning

El aprendizaje máquina o *Machine Learning* (ML) es una rama de la inteligencia artificial que permite a los sistemas aprender patrones a partir de datos sin ser programados explícitamente. En el contexto de la detección de drones, esta técnica se emplea para reconocer firmas acústicas específicas de drones a partir de grandes volúmenes de datos sonoros, diferenciándolos de ruidos ambientales y otras fuentes sonoras [42].

La detección acústica de UAVs mediante *Machine Learning* se basa en la extracción de características relevantes del sonido y el entrenamiento de modelos de clasificación que pueden diferenciar entre sonidos de drones y ruido ambiental. No obstante, debe destacarse que su implementación requiere un alto poder computacional y grandes volúmenes de datos etiquetados. Algunas técnicas de ML son las Máquinas de Vector Soporte o SVM (*Support Vector Machines*) y RF o *Random Forest* [42].

En la misma rama, el *Deep Learning* es “un subconjunto del *Machine Learning* que utiliza redes neuronales multicapa, llamadas redes neuronales profundas, para simular el complejo poder de toma de decisiones del cerebro humano” [43]. En el caso de la detección acústica de drones, permite la identificación precisa de UAVs basándose en su huella acústica sin requerir la extracción manual de características, como ocurre en los métodos tradicionales [44].

En el contexto del procesamiento de audio, algunas técnicas del *Deep Learning* son las Redes Neuronales Recurrentes o RNN (*Recurrent Neural Network*), las Redes Neuronales Convolucionales o CNN (*Convolutional Neural Network*) y las CRNN, combinación de ambas:

- CNN: Utilizan capas convolucionales para procesar espectrogramas de audio, extrayendo patrones espaciales relevantes. Son particularmente efectivas en la detección de drones

debido a su capacidad de identificar características acústicas específicas en grandes volúmenes de datos [44].

- RNN: Se especializan en el análisis de secuencias temporales, lo que les permite modelar la evolución del sonido en el tiempo. Son adecuadas para la detección de drones en señales acústicas continuas, aunque pueden presentar problemas con largas dependencias temporales [44].
- CRNN: Combinan CNN y RNN, permitiendo aprovechar tanto la extracción de características espaciales como la modelización de la evolución temporal de la señal. Esto las convierte en una opción robusta para la detección de drones en entornos dinámicos [44].

Finalmente, en la Tabla 2-3 se puede observar un resumen con todas las técnicas de identificación descritas.

Técnica	Ventajas	Limitaciones
Correlación de patrones	Eficaz si hay una base de datos de referencia sólida	Requiere una base de datos extensa y bien definida
Identificación de armónicos	Permite diferenciar drones de otras fuentes sonoras con precisión	Sensibilidad al ruido y a interferencias en el entorno
GMM	Modela la variabilidad de las señales acústicas en distintos entornos	Requiere ajuste fino de parámetros y entrenamiento previo
Filtro Bayesiano	Reduce la incertidumbre en la localización de drones	Depende de mediciones previas, afectado por ecos y ruido
<i>Machine Learning</i>	Alta capacidad de clasificación con datos etiquetados	Alto poder computacional y grandes volúmenes de datos necesarios
<i>Deep Learning</i>	Precisión superior en detección, no requiere extracción manual de características	Requiere hardware especializado y alto poder de procesamiento

Tabla 2-3 Resumen técnicas de identificación

Después de analizar las distintas metodologías disponibles, hemos decidido apostar por la correlación de patrones como estrategia para la identificación de drones. Aunque este enfoque suele asociarse con la necesidad de contar con una base de datos extensa, nuestra intención no es abarcar todos los modelos desde el inicio, sino construir un procedimiento sólido a partir de un único dron. Una vez definido y optimizado el proceso, la incorporación de nuevas referencias será una cuestión de ampliación progresiva, no de rediseño.

Lo que hace atractiva esta técnica es su capacidad para comparar señales en tiempo real, cotejando la firma acústica captada con registros previos. Esto nos permite trabajar con un enfoque más directo y estructurado, evitando la dependencia de modelos estadísticos o sistemas de aprendizaje que requieren grandes volúmenes de datos y un procesamiento más costoso. En nuestro caso, lo fundamental es sentar las bases de un sistema eficiente, escalable y de bajo coste, que pueda desplegarse en escenarios reales sin requerir una infraestructura compleja.

Otro punto clave es la flexibilidad que ofrece la correlación de patrones. Una vez consolidado el proceso de detección con un dron, la integración de otros modelos será relativamente sencilla: bastará con registrar sus firmas acústicas y ampliar la base de comparación. Esto significa que, en lugar de

centrarnos en desarrollar un sistema generalista desde el principio, adoptamos una estrategia escalonada, asegurándonos de que cada fase funcione correctamente antes de dar el siguiente paso.

En última instancia, nuestra prioridad es la efectividad en la identificación, no la sobrecomplicación del sistema. Apostar por la correlación de patrones nos permite avanzar con una metodología clara, con la confianza de que cada nueva incorporación fortalecerá el modelo sin comprometer su funcionamiento.

2.6 Redes de sensores

Tras haber abordado en los apartados anteriores las distintas técnicas de detección y clasificación de drones, es fundamental analizar cómo se estructuran y operan las redes de sensores que permiten la implementación efectiva de estos sistemas.

2.6.1 Concepto general

Una red de sensores es un conjunto de dispositivos interconectados que capturan, procesan y transmiten información del entorno a través de diferentes protocolos de comunicación [45]. Esta red de sensores está compuesta por múltiples nodos sensores que trabajan de manera cooperativa para monitorear condiciones específicas del entorno. Estos nodos suelen incluir [46]:

- Sensores: Dispositivos encargados de la detección de variables como el sonido, temperatura, presión o radiación electromagnética.
- Microcontroladores: Procesadores encargados de la gestión de datos capturados.
- Módulos de comunicación: Interfaz que permite la transmisión de la información hacia otros nodos o centros de procesamiento.
- Fuentes de energía: Generalmente baterías recargables portátiles o energía recolectada del ambiente.
- Sistema central: Parte clave de la red de sensores. Se trata del nodo principal encargado de recibir, procesar y gestionar los datos de la red de sensores. En un sistema centralizado, este nodo actúa como el punto de control donde se almacenan y analizan los datos, permitiendo la toma de decisiones en tiempo real. Esto garantiza una gestión eficiente de la información, aunque también representa un punto crítico, pues su ausencia podría afectar el funcionamiento global del sistema.

2.6.2 Módulos de comunicación

Para garantizar la interconectividad de los nodos y la transmisión de datos dentro de la red, se emplean distintos métodos de comunicación inalámbrica, cada uno con características específicas en cuanto a alcance, consumo de energía y velocidad de transmisión. Entre las principales tecnologías utilizadas se encuentran WiFi, Bluetooth, ZigBee y LoRa. También existen otros métodos de comunicación que, aunque no se detallarán con profundidad, son relevantes en ciertos contextos. Entre ellos, las comunicaciones por cable ofrecen conexiones estables y seguras, ya que están menos expuestas a interferencias y ataques electrónicos. Sin embargo, su despliegue puede ser costoso y menos flexible en entornos dinámicos, lo que limita su uso en escenarios tácticos móviles [47].

Por otro lado, en entornos de guerra electrónica o EW (*Electronic Warfare*) se emplean técnicas avanzadas para interrumpir, interceptar o engañar las comunicaciones enemigas, utilizando el espectro electromagnético como medio de ataque o de defensa. En el ámbito de drones, técnicas como el *jamming* permiten interrumpir las señales de comunicación y control entre el operador y el UAV, impidiendo su navegación y obligándolo a aterrizar o desviarse de su ruta [48]. Aunque este tipo de medidas no son el centro de estudio de este trabajo, pero pueden servir como medida para la posterior fase de neutralización del dron.

2.6.2.1 WiFi

WiFi, o *Wireless Fidelity* es una tecnología que permite la conexión de dispositivos a través de una infraestructura de red inalámbrica. Se caracteriza por su alta velocidad de transmisión de datos y una cobertura extensa en entornos urbanos. WiFi opera en bandas de 2,4 GHz y 5 GHz, lo que permite

diferentes niveles de alcance y rendimiento. Aunque es ideal para aplicaciones que requieren grandes volúmenes de datos, su consumo energético elevado lo hace menos eficiente para redes de sensores con recursos limitados. Se emplea principalmente en entornos donde la transmisión de datos en tiempo real es esencial, como en videovigilancia y monitoreo de infraestructuras críticas [49].

2.6.2.2 Bluetooth

Bluetooth es una tecnología de comunicación inalámbrica de corto alcance diseñada para la transmisión de datos entre dispositivos cercanos. Funciona en la banda de 2,4 GHz y emplea un esquema de salto de frecuencia para minimizar interferencias. Su principal ventaja es su bajo consumo energético, especialmente en su variante *Bluetooth Low Energy* (BLE), lo que lo hace ideal para dispositivos con baterías pequeñas. Se utiliza en aplicaciones de redes de sensores donde se requiere una comunicación eficiente en entornos cerrados, como en hogares inteligentes y dispositivos portátiles [50].

2.6.2.3 ZigBee

ZigBee es un protocolo de comunicación inalámbrica diseñado para redes de sensores con bajo consumo energético y larga duración de batería. Su principal característica es la capacidad de formar redes en malla, donde cada nodo puede retransmitir datos y extender la cobertura sin necesidad de aumentar la potencia de transmisión. Esto lo hace altamente eficiente en entornos con múltiples dispositivos interconectados. ZigBee opera en las bandas de 2,4 GHz, 915 MHz y 868 MHz, lo que le permite adaptarse a distintos escenarios. Se utiliza ampliamente en aplicaciones de automatización industrial, monitoreo ambiental y hogares inteligentes [49].

2.6.2.4 LoRa

Long-Range es una tecnología de comunicación inalámbrica de largo alcance que permite la transmisión de datos a grandes distancias con un consumo de energía extremadamente bajo. Se basa en la modulación de espectro ensanchado y permite la transmisión de datos en bandas de frecuencia sin licencia, lo que facilita su despliegue en diversas aplicaciones sin restricciones regulatorias significativas [46], [51].

En la Tabla 2-4 se puede observar un resumen de las metodologías de comunicación inalámbrica comentadas con las ventajas, limitaciones y aplicaciones de cada una.

Módulo de comunicación	Ventajas	Limitaciones	Aplicaciones
WiFi	Alta velocidad de transmisión y amplio ancho de banda	Alto consumo energético, cobertura limitada en presencia de obstáculos	Videovigilancia, monitoreo de infraestructuras críticas
Bluetooth	Bajo consumo energético y conexión sencilla entre dispositivos cercanos	Alcance limitado, velocidad de transmisión inferior a WiFi	Hogares inteligentes, dispositivos portátiles
ZigBee	Bajo consumo energético, largo alcance en interiores	Baja tasa de datos, mayor latencia en la transmisión	Automatización industrial, monitoreo ambiental
LoRa	Gran alcance, bajo consumo de energía, ideal para entornos rurales	Baja velocidad de transmisión, no apto para grandes volúmenes de datos	Monitoreo ambiental y comunicaciones en zonas remotas

Tabla 2-4 Métodos de comunicación inalámbrica

Para la implementación del sistema de detección, hemos optado por WiFi como tecnología de comunicación inalámbrica. Aunque su consumo energético es superior al de otras opciones como Bluetooth o ZigBee, su mayor velocidad de transmisión y capacidad para manejar grandes volúmenes de datos lo convierten en la alternativa más adecuada para nuestro propósito. En un sistema de detección

acústica, donde es crucial enviar y procesar señales en tiempo real, una conexión con suficiente ancho de banda es prioritaria.

Además, la infraestructura de WiFi es accesible y fácil de integrar, lo que permite desplegar la red de sensores sin necesidad de módulos adicionales o configuraciones complejas. Este enfoque no solo simplifica la implementación inicial, sino que también facilita la escalabilidad del sistema, permitiendo la incorporación de nuevos nodos sin comprometer el rendimiento de la comunicación.

2.6.3 Aplicaciones militares de las redes de sensores.

Las redes de sensores inalámbricas, también conocidas como WSN (*Wireless Sensor Networks*), han demostrado ser una tecnología fundamental en operaciones militares modernas, facilitando la recopilación de información en tiempo real y optimizando la toma de decisiones en el campo de batalla. Diversas aplicaciones han sido desarrolladas para mejorar la seguridad, vigilancia y eficiencia logística de las tropas. En la Figura 2-15 se puede observar un esquema general de un escenario de guerra en el que se contemplan múltiples aplicaciones de las WSN.

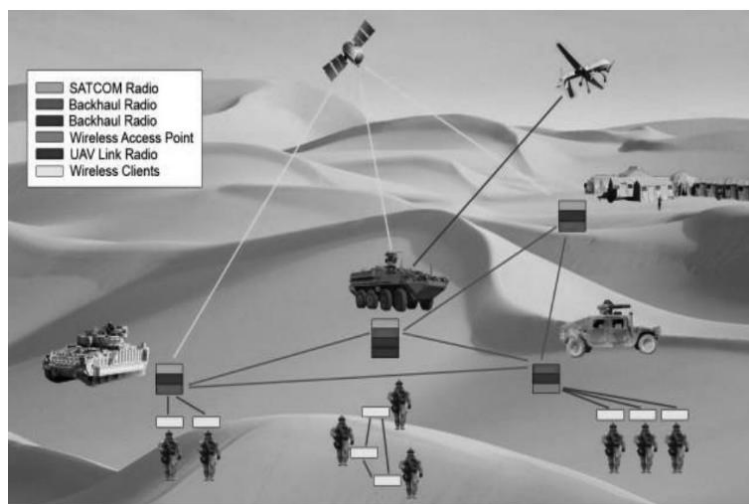


Figura 2-15 WSN en escenario de guerra [45]

Uno de los usos clave de estas redes es la guerra de minas (*Mine Warfare*), concretamente en el SHLM o *Self-Healing Land Mines*, un sistema de minas terrestres antitanque autónomas que pueden detectar riesgos y reaccionar automáticamente ante amenazas para maximizar su efectividad en la defensa de perímetros. En el ámbito de la detección de disparos y explosiones, el AAP o *Aerostat Acoustic Payload for Transient Detection* consiste en sensores acústicos suspendidos en aerostatos que detectan disparos de morteros, artillería y armas pequeñas, permitiendo una rápida respuesta de las unidades en el terreno. Por otro lado, el *Soldier Detection and Tracking* (SDT) emplea sensores acústicos y visuales para localizar la posición de soldados enemigos en entornos urbanos y de combate cerrado. En la misma línea, otra aplicación significativa es el *Early Attack Reaction Sensor* (EARS), un sistema portátil para soldados que detecta disparos y determina su dirección, proporcionando información crítica para la protección de unidades en combate [45].

Para la protección de infraestructuras y bases militares, el sistema *Perimeter Protection* (PP) integra redes de sensores que combinan cámaras, sensores térmicos e infrarrojos para una vigilancia continua y detección de intrusos en zonas críticas. Finalmente, el *Acoustic Threatening Sound Recognition System* (ATS) permite identificar y clasificar sonidos de amenazas en entornos de combate asimétrico, optimizando la capacidad de respuesta táctica [45].

Dado el éxito probado de estas aplicaciones en operaciones militares, la adaptación de las redes de sensores para la detección de drones representa una evolución natural. La integración de estos sistemas con el análisis de datos permitiría una identificación más precisa de UAVs hostiles, fortaleciendo las capacidades defensivas y la seguridad operativa en el campo de batalla.

3 DESARROLLO DEL TFG

3.1 Metodología

Este apartado describe la metodología que se seguirá en el desarrollo del proyecto, estructurada en distintas fases que permitirán alcanzar el objetivo de diseñar un sistema eficiente de detección de drones en tiempo real y en un espacio confinado. Se establecerá un flujo de trabajo claro y organizado, desde la recopilación de datos hasta la implementación del sistema de detección.

3.1.1 Propósito

Se proyecta diseñar e implementar un sistema de detección de drones en entornos interiores mediante el análisis en tiempo real de su huella acústica. Se busca desarrollar un método efectivo que permita identificar la presencia de un dron basándose en su firma sonora, utilizando una red de sensores conectados inalámbricamente a un sistema central de procesamiento.

El desarrollo del proyecto se enfocará en varias etapas clave. En primer lugar, se capturará audio mediante distintos tipos de sensores acústicos conectados a varias Raspberry Pi, asegurando que la señal capturada sea clara y fiable. La Raspberry Pi, haciendo su función de microcontrolador, transmitirá el audio en tiempo real al sistema de procesamiento, el portátil. Una vez en este, se procesarán los datos recopilados para identificar patrones característicos de la señal emitida con los de un dron ya conocido, dando a conocer al operario si el audio recibido se trata de este dron o no.

Para ello, se ha realizado un estudio de las cualidades de dicho UAV, similar a la obtención de inteligencia como si de la firma del radar de un buque o la hélice de un submarino se tratase. En dicho estudio, se ha hecho un análisis en el que se han generado espectrogramas que permiten visualizar la distribución de frecuencias en función del tiempo, así como la intensidad de sonido.

En este análisis, se implementa un filtro paso-banda que permite aislar esa banda de frecuencias de interés, estableciéndose posteriormente un umbral de detección basado en la energía acústica en decibelios, con el objetivo de diferenciar de manera efectiva entre un dron y otros sonidos ambientales. De esta manera, con un filtro aplicado a una banda de frecuencias de interés y un umbral de intensidad definido, se podrá identificar si efectivamente el sonido recibido se trata del UAV que conocemos o no.

Finalmente, gracias a este análisis, el filtrado y umbral definido se aplicarán en tiempo real a los audios recibidos, permitiendo discernir la presencia del dron.

Este enfoque permitirá evaluar la viabilidad del sistema en entornos reales y su capacidad de adaptarse a diversas condiciones operativas. La implementación de este método sentará las bases para futuros desarrollos en detección de drones en contextos y de guerra urbana, donde la identificación temprana de estos dispositivos puede ser crucial.

3.1.2 Desarrollo de fases

Para estructurar el desarrollo del proyecto de manera clara y organizada, se ha definido un conjunto de fases que permitirán avanzar de forma progresiva desde la captura de datos hasta la implementación del sistema de detección final. La Figura 3-1 muestra un esquema con las fases del desarrollo del TFG y las tareas asociadas a cada una de ellas.

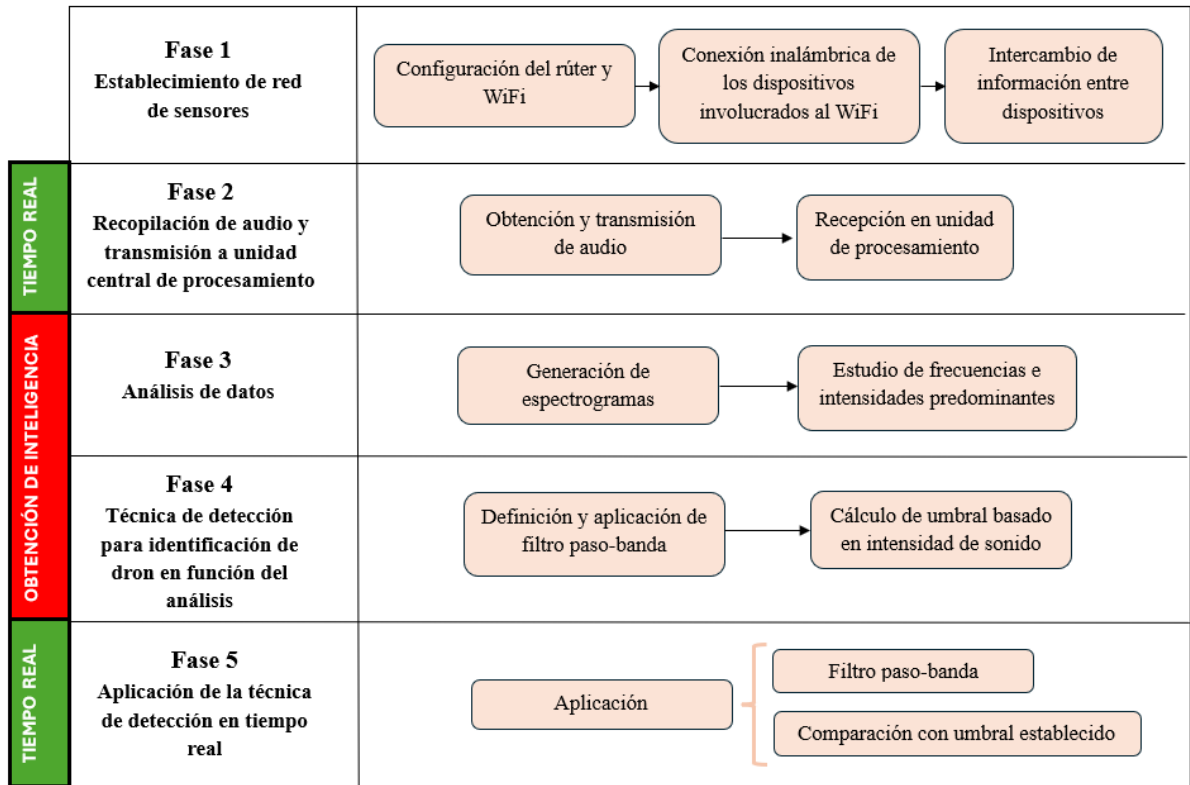


Figura 3-1 Esquema de fases

El desarrollo del sistema se divide en dos grandes bloques: la obtención de inteligencia, realizada de manera pasiva, y su aplicación en tiempo real, de manera activa. En primer lugar, se establece la red de sensores y se recopilan los datos de audio para su análisis. A partir de estos datos, se estudian las características acústicas del dron, obteniendo la información necesaria para su detección.

Una vez definido el método de identificación, este se implementa en tiempo real, permitiendo que el sistema procese continuamente los audios recibidos y determine si corresponden al dron. De este modo, la detección se basa en un conocimiento previo estructurado, asegurando un funcionamiento adecuado.

3.2 Consideraciones previas

Para la implementación del sistema de detección de drones, se ha seleccionado cuidadosamente una combinación de hardware y software que garantiza un equilibrio óptimo entre costo, facilidad de implementación y compatibilidad con los requisitos del proyecto, así como un espacio en interiores para llevar a cabo las simulaciones y pruebas que sustentan el desarrollo de este trabajo.

La arquitectura sigue una topología en estrella, en la que los sensores acústicos están conectados a microcontroladores, los cuales transmiten los datos a un nodo central (PC). En este nodo, se ejecuta el código de análisis encargado de detectar drones en entornos interiores. En la Figura 3-2 se puede observar esta arquitectura:

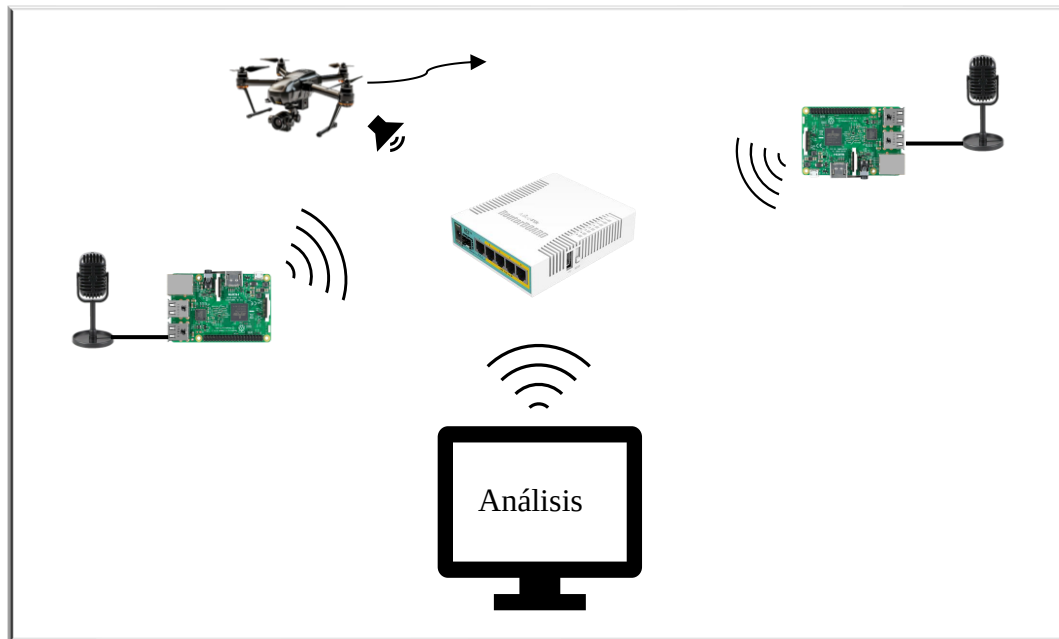


Figura 3-2 Arquitectura del sistema

A continuación, se detallan los componentes utilizados, su función en el sistema y las características técnicas que justifican su elección, además de la descripción del espacio utilizado.

3.2.1 Hardware empleado

3.2.1.1 R ter MikroTik

Para facilitar la comunicaci n inal mbrica entre los dispositivos del sistema, se ha seleccionado el r ter *MikroTik hEX PoE*, ilustrado en la Figura 3-3. Este dispositivo act a como el n cleo de la red, gestionando la conectividad entre las Raspberry Pi y el port til utilizado para el procesamiento central.



Figura 3-3 R ter MikroTik [52]

El *MikroTik hEX PoE* ofrece una cobertura WiFi simult nea en las bandas de 2,4 GHz y 5 GHz, lo que permite una mayor flexibilidad en la conexi n de dispositivos. Las Raspberry Pi se conectan a la banda de 2,4 GHz debido a su mayor cobertura y mejor penetraci n en interiores, mientras que el port til utiliza la banda de 5 GHz para aprovechar velocidades m s altas y menor interferencia, optimizando as  la comunicaci n con la unidad central de procesamiento.

Equipado con cinco puertos Ethernet de 10/100/1000 Mbps, el r ter asegura conexiones cableadas de alta velocidad. Adem s, incluye un puerto USB que puede utilizarse para almacenamiento externo o m dems 3G/4G, y admite alimentaci n a trav s de Ethernet (PoE). [53]. En la Tabla 3-1 se puede observar un resumen de las sus caracter sticas principales.

Especificación	Detalle
CPU	QCA9557, 1 núcleo, 800 MHz
RAM	128 MB
Almacenamiento	16 MB Flash
Puertos Ethernet	5 × 10/100/10000 Mbps
Bandas inalámbricas	2,4 GHz (300 Mbps), 5 GHz (867Mbps)
Dimensiones	114 × 137 × 29 mm
Consumo máximo de energía	54 W

Tabla 3-1 Características router MicroTik (tomada de [53])

Estas características hacen del router seleccionado una opción versátil y económica para gestionar la red de sensores en el proyecto de detección de drones.

3.2.1.2 Micrófono de condensador TONOR TC30

El TONOR TC30 será el micrófono principal utilizado en este trabajo para la captura de audio en las pruebas de detección de drones. Su elección se debe a la disponibilidad de este y a la facilidad de su uso (conexión por USB). Además, su patrón polar cardioide permite una captación direccional del sonido minimizando el ruido ambiental [54]. En la Figura 3-4 se puede observar este micrófono.



Figura 3-4 TONOR TC30 [54]

Para su uso en este proyecto, el TC30 se conectará a la Raspberry Pi 3, desde donde se capturará el audio para su posterior análisis. Gracias a su diseño cardioide, este micrófono permite registrar sonidos provenientes de una dirección específica, lo que facilitará la detección de drones minimizando interferencias de ruido externo. Además, su rango de frecuencia de 50 Hz a 20 kHz cubre el espectro sonoro en el que operan los drones utilizados en las pruebas [54].

3.2.1.3 Raspberry Pi 3 modelo B

En el sistema de detección de drones, las Raspberry Pi 3 Modelo B actúan como nodos de adquisición y transmisión de datos. Su función principal es recibir el audio captado por el micrófono y enviarlo a través de la red al servidor central, donde se realiza el análisis de los datos. No realiza procesamiento avanzado en esta etapa, sino que simplemente retransmite toda la información recogida.

En la Figura 3-5 se muestra el modelo utilizado.



Figura 3-5 Vista general Raspberry Pi 3 Modelo B [55]

La elección de la Raspberry Pi 3 Modelo B se debe a su conectividad integrada, ya que cuenta con WiFi 802.11n y Bluetooth 4.1, lo que permite la transmisión inalámbrica de los datos sin necesidad de adaptadores adicionales ([55]). Además, su bajo costo y eficiencia energética la convierten en una opción ideal para un sistema de monitoreo distribuido con múltiples sensores. Su comunidad de desarrollo es extensa, lo que facilita el acceso a documentación y soporte técnico en caso de problemas. Por otro lado, la presencia de múltiples puertos USB permite la conexión de diferentes micrófonos sin necesidad de *hubs* adicionales. Por último, su reducido tamaño la convierte en un elemento estratégico discreto a la hora de establecer un sistema de seguridad dentro de un edificio o una habitación.

En general, cuenta con los siguientes puertos [55]:

- 4 USB 2.0: Pueden usarse para conectar micrófonos, teclado y ratón, entre otros.
- HDMI: Para poder acceder a su interfaz gráfica en una pantalla u ordenador (de esta manera podemos configurar parámetros como su conexión WiFi)
- Ethernet 10/100 Mbps: Permite conexión cableada a un router.
- WiFi 802.11n: Permite la transmisión inalámbrica de los datos capturados al servidor central.
- Bluetooth 4.1: En este trabajo no se emplea pero también podría aplicarse para la conexión con otros dispositivos.
- GPIO de 40 pines: Para conectar distintos sensores.
- Micro USB (alimentación): Para alimentar la Raspberry con una placa de 5 V/2,5 A.
- Ranura para MicroSD: Contiene el sistema operativo y software necesario para la transmisión de datos.

La Raspberry Pi 3 Modelo B utiliza como sistema operativo Raspberry Pi OS (anteriormente conocido como Raspbian), una distribución basada en Debian Linux y optimizada para el hardware disponible [55]. Este sistema operativo se carga desde una tarjeta microSD y ofrece un entorno de escritorio ligero, adecuado para las capacidades de la Raspberry Pi.

Cabe destacar que el sistema operativo se ejecuta desde una tarjeta microSD y permite el acceso remoto mediante el comando *ssh*, lo que simplifica la configuración sin necesidad de conectar un monitor. Además, incluye soporte para librerías de procesamiento de señales, lo que podría facilitar futuras mejoras en el análisis de audio dentro del propio sistema.

3.2.1.4 Dron

Para la realización de las pruebas, se ha utilizado un mini dron plegable con cámara HD 4 K. Este dispositivo se utiliza para generar las señales acústicas representativas de su comportamiento en entornos interiores, permitiendo así el desarrollo y la prueba del sistema de detección. Este modelo ha sido escogido por su portabilidad, facilidad de uso y costo accesible, lo que permite realizar múltiples pruebas sin incurrir en grandes gastos.

El dron cuenta con diversas funcionalidades que permiten simular condiciones de vuelo reales. Dispone de modo sin cabeza, lo que facilita su control sin necesidad de ajustar su orientación, y de mantenimiento de altitud, lo que le permite mantenerse estable en el aire sin necesidad de ajustes

constantes por parte del operador. Además, incorpora un sistema de retorno automático, lo que garantiza que pueda volver a su punto de origen con facilidad, reduciendo el riesgo de pérdida o accidentes durante las pruebas. Su control puede realizarse tanto mediante un mando a distancia como a través de una aplicación móvil, lo que proporciona flexibilidad en su manejo. En cuanto a especificaciones técnicas, su batería modular proporciona una autonomía de aproximadamente 15 minutos por carga, lo que es suficiente para la mayoría de las pruebas previstas. Su alcance máximo de control es de hasta 100 metros, lo que permite evaluar su ruido en diferentes distancias y condiciones de vuelo. Además, cuenta con luces LED, lo que facilita su visibilidad en entornos con poca iluminación. En la Figura 3-6 se ve puede observar el UAV con su mando de control remoto [56].



Figura 3-6 Dron profesional E88 4K empleado en las pruebas [56]

3.2.1.5 Portátil

Un ordenador portátil actúa como sistema central de procesamiento, siendo el encargado de recibir y analizar los datos acústicos transmitidos por las Raspberry Pi. Su elección se debe a su capacidad de procesamiento suficiente para manejar algoritmos de filtrado y análisis espectral, asegurando una detección eficiente de drones en el entorno. Además, su conectividad inalámbrica permite la recepción de datos sin necesidad de conexiones físicas, facilitando la implementación del sistema en diferentes ubicaciones. Gracias a su potencia y versatilidad, el ordenador permite realizar pruebas y ajustes en el modelo de detección, optimizando su precisión y rendimiento.

3.2.2 Software, editor gráfico y librerías empleadas

El desarrollo del sistema de detección de drones ha requerido el uso de herramientas de software especializadas para la programación, análisis y visualización de datos. Se ha optado por Python como lenguaje principal debido a su versatilidad y su amplio ecosistema de librerías para el procesamiento de señales. Se ha utilizado la versión 3.11.9, instalada desde la página oficial de dicho lenguaje ([57]).

Para la escritura y depuración del código, se ha utilizado el entorno de desarrollo Visual Studio Code, elegido por su facilidad de integración con herramientas externas y su compatibilidad con múltiples plataformas. Dicho entorno también se ha instalado desde su página oficial ([58]).

Una vez instalados el lenguaje de programación y el entorno de desarrollo, se han desarrollado una serie de *scripts* diseñados para gestionar todo el flujo de trabajo, desde la captura y transmisión de audio hasta su análisis espectral y la definición de un umbral de detección fiable. Para su desarrollo, ha sido fundamental el uso de diversas bibliotecas que han permitido un procesamiento eficiente de los datos acústicos. A continuación, se detallan las principales herramientas empleadas en el proyecto:

- *NumPy* ha sido indispensable para la manipulación de datos numéricos, en especial en el cálculo de la Transformada Rápida de Fourier (FFT), empleada en el análisis de las frecuencias predominantes en las grabaciones. Su capacidad para realizar operaciones vectorizadas ha optimizado significativamente el rendimiento en el procesamiento de señales. Se instala con el comando:

```
pip install numpy
```

- *SciPy* ha jugado un papel clave en distintas fases del análisis de señal. La función *butter* se ha empleado para diseñar filtros paso-banda Butterworth, lo que ha permitido aislar con precisión la banda de frecuencias característica del dron. Asimismo, *filtfilt* ha sido utilizada para aplicar el filtrado sin generar distorsión de fase, garantizando la fidelidad de la señal. Además, *spectrogram* ha sido una herramienta fundamental para la representación temporal de la energía acústica, permitiendo evaluar la presencia del dron en cada instante de la grabación. Se instala con:

```
pip install scipy
```

- *Matplotlib* ha sido esencial en la visualización de datos, facilitando la representación gráfica de los distintos análisis. Desde los *boxplots* utilizados para evaluar la distribución de intensidades hasta los espectrogramas filtrados, esta biblioteca ha permitido interpretar con claridad los resultados obtenidos. Se instala con:

```
pip install matplotlib
```

- *PyAudio* ha sido la pieza clave en la captura y reproducción de audio en tiempo real. Gracias a esta biblioteca, se ha podido configurar la adquisición de sonido desde los micrófonos conectados a las Raspberry Pi y su posterior transmisión a través de la red. Se instala con:

```
pip install pyaudio
```

- *Wave*, incluida en la instalación estándar de Python, se ha empleado para la lectura y almacenamiento de archivos de audio en formato WAV, facilitando el registro de los sonidos capturados para su posterior análisis.
- *Socket* ha gestionado la comunicación en red mediante el protocolo UDP (*User Datagram Protocol*), permitiendo la transmisión de los datos de audio desde las Raspberry Pi hasta el ordenador central. Esta librería viene preinstalada en Python y ha sido fundamental para garantizar una comunicación rápida y eficiente entre los dispositivos.
- *Logging*, incluida en Python por defecto, ha sido utilizada en algunos *scripts* para la gestión de mensajes y depuración del sistema. Permite registrar eventos importantes, advertencias y errores durante la ejecución del código, facilitando el monitoreo del funcionamiento del sistema y la detección de posibles fallos.
- *Threading*, también incorporada en la biblioteca estándar de Python, ha hecho posible la ejecución simultánea de múltiples procesos dentro del sistema. Ha sido utilizada principalmente para gestionar la recepción de audio en paralelo desde varias Raspberry Pi, evitando bloqueos y asegurando un flujo continuo de datos.
- *Time*, disponible de forma nativa en Python, ha sido empleada para la gestión de tiempos y sincronización dentro del sistema. Se ha utilizado tanto en la segmentación de ventanas de grabación como en la asignación de marcas temporales a los archivos generados.
- *Signal*, también integrada en Python, ha sido esencial para gestionar las interrupciones manuales del programa mediante “Ctrl + C”. Su uso ha permitido implementar una finalización segura del sistema, cerrando correctamente sockets y liberando recursos antes de finalizar la ejecución.
- *Sys*, otra de las bibliotecas estándar de Python, ha sido utilizada para manejar argumentos del sistema y controlar la salida del programa. Se ha empleado en conjunto con *signal* para garantizar una terminación ordenada del *script*, evitando bloqueos inesperados.
- Finalmente, *os*, presente también en Python, ha sido utilizada para la gestión de archivos y directorios, facilitando la automatización de tareas como la carga de audios y el almacenamiento de gráficos generados en las distintas fases del análisis.

Todas estas librerías se han instalado o venían instaladas ya en la versión de Python del ordenador. Debe destacarse que en las Raspberry Pi solo han sido necesarias dos librerías principales: *socket* y *pyaudio*. La primera, *socket*, ya viene incluida en la instalación estándar de Python, por lo que no requiere ninguna instalación adicional. En cambio, *pyaudio* sí debe instalarse manualmente, ya que es la encargada de gestionar la captura y transmisión del audio desde los micrófonos conectados a las Raspberry Pi. Para instalarla en el sistema operativo de la Raspberry Pi, se puede utilizar el siguiente comando, que difiere un poco del utilizado para el ordenador:

```
sudo apt install python3-pyaudio
```

Este comando descarga e instala *pyaudio* junto con sus dependencias en la Raspberry Pi, permitiendo la adquisición del sonido en tiempo real para su posterior envío al sistema de procesamiento.

Gracias a la combinación de todas estas herramientas, ha sido posible desarrollar un sistema robusto y eficiente para la detección acústica de drones, garantizando la correcta adquisición, filtrado y análisis de los datos capturados.

3.2.2.1 Uso de la IA generativa

Este trabajo también ha integrado herramientas de inteligencia artificial generativa, como GitHub Copilot y ChatGPT (OpenAI), para optimizar la memoria y el desarrollo del código en Python. Todas las aportaciones generadas por IA fueron revisadas, modificadas y ajustadas para alinearse con los objetivos del estudio y cumplir con las normativas académicas. El uso de estas tecnologías se llevó a cabo de forma ética y responsable, asegurando la originalidad del contenido.

3.2.3 Escenario de pruebas

La sala de estar del cuartel “Marqués de la Victoria”, ubicada la primera planta, ha sido el espacio elegido para llevar a cabo las pruebas del sistema de detección de drones. Su amplitud y distribución abierta facilitan la realización de experimentos a diferentes distancias sin obstáculos significativos. Además, al tratarse de un recinto cerrado y resguardado, se minimiza la interferencia de ruidos externos, lo que garantiza un entorno controlado para la captura y análisis de las señales acústicas del dron. Las condiciones de esta sala permiten evaluar el rendimiento del sistema en un escenario representativo de un entorno urbano interior, donde la detección acústica puede verse afectada por la presencia de ecos y ruido de fondo. Se ve ilustrada en la Figura 3-7.

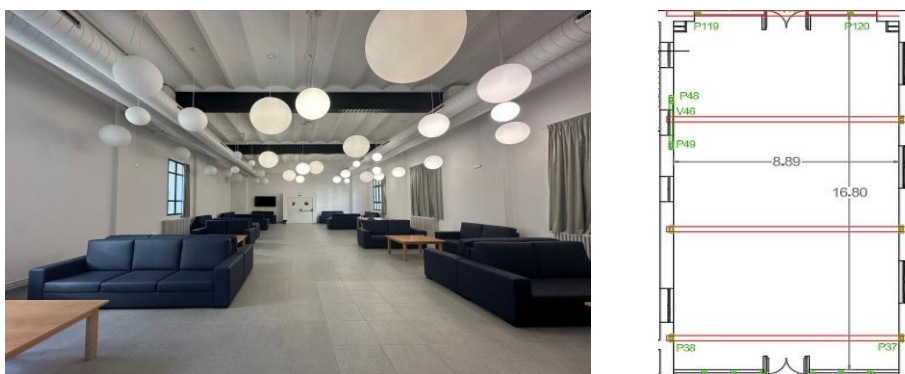


Figura 3-7 Sala de estar del cuartel Marqués de la Victoria

3.3 Primera fase: Establecimiento de red de sensores

El primer paso en el desarrollo del sistema de detección de drones ha consistido en la configuración de la red de sensores, asegurando la comunicación entre las Raspberry Pi y el ordenador central. Para ello, se ha establecido una red inalámbrica gestionada por el router MikroTik, a la que se han conectado las Raspberry Pi 3 Modelo B mediante la banda de 2,4 GHz y el portátil a la banda de 5 GHz, como se ha mencionado ya anteriormente.

3.3.1 Configuración del router

Una vez conectado a alimentación, se ha accedido a la IP del router y se ha establecido una contraseña para que los dispositivos puedan acceder a él sin problema. Para ello, se introdujo la dirección IP (192.168.88.1) en el navegador y se accedió a la configuración mediante la interfaz de MikroTik. Este paso se puede observar en la Figura 3-8.

Figura 3-8 Configuración del router

Además, en la Figura 3-9 se muestra el resto de configuración del router, destacándose los siguientes parámetros:

- Dirección IP: 192.168.88.1. Se trata de la puerta de enlace para los dispositivos que se conecten a la red.
- Máscara de subred: 255.255.255.0 / 24. Lo que significa que la red puede albergar hasta 254 dispositivos con direcciones IP dentro del rango 192.168.88.1 – 192.168.88.254.
- Servidor DHCP activado: El router asigna direcciones IP a los dispositivos conectados dentro del rango 192.168.88.10 – 192.168.88.254, asegurando que cada uno de ellos obtenga una dirección IP sin conflicto.
- Cortafuegos activado: Esto proporciona seguridad adicional al controlar qué tráfico de red puede atravesar el router.

Figura 3-9 Configuración avanzada del router

Esta configuración garantiza que la red de sensores funcione correctamente, permitiendo que las Raspberry Pi y el ordenador se conecten de manera estable y puedan comunicarse entre sí. Además, la asignación automática de direcciones IP mediante DHCP facilita la configuración inicial, evitando la necesidad de configurar manualmente cada dispositivo.

3.3.2 Configuración de dispositivos

Una vez configurado el router, el siguiente paso ha sido preparar los dispositivos que forman parte del sistema: las Raspberry Pi 3 Modelo B y el ordenador HP que actúa como unidad central de procesamiento. Esta configuración ha permitido que cada dispositivo se integre correctamente en la red y pueda intercambiar datos de manera eficiente.

3.3.2.1 Configuración de ordenador central

El ordenador ha sido configurado para gestionar y recibir la información proveniente de las Raspberry Pi. Para garantizar una conexión estable y de alto rendimiento, se ha conectado a la banda de 5 GHz del router, lo que ha permitido una menor latencia y una transmisión de datos más rápida.

Para obtener la dirección IP del ordenador, confirmando de esta manera su integración en la red, se ha utilizado el siguiente comando:

```
ipconfig
```

En la Figura 3-10 se puede observar el resultado de este comando, observándose, tanto la IP asignada al ordenador, como la puerta de enlace del router MikroTik.

```
Adaptador de LAN inalámbrica Wi-Fi:

Sufijo DNS específico para la conexión. . . :
Vínculo: dirección IPv6 local. . . : fe80::d6b8:f1ad:3350:ae28%24
Dirección IPv4. . . . . : 192.168.88.247
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 192.168.88.1
```

Figura 3-10 Obtención de IP del ordenador

3.3.2.2 Raspberry Pi Modelo 3 B

Cada una de las Raspberry Pi 3 Modelo B se ha preparado para su conexión a la red accediendo a su interfaz gráfica, para ello se han conectado a un monitor, teclado y ratón en los respectivos puertos habilitados para ello, lo que ha permitido una configuración inicial manual. En la Figura 3-11 se ve ilustrada la interfaz gráfica mencionada y su conexión manual a la banda de 2,4 GHz a través del gestor de redes del sistema operativo.

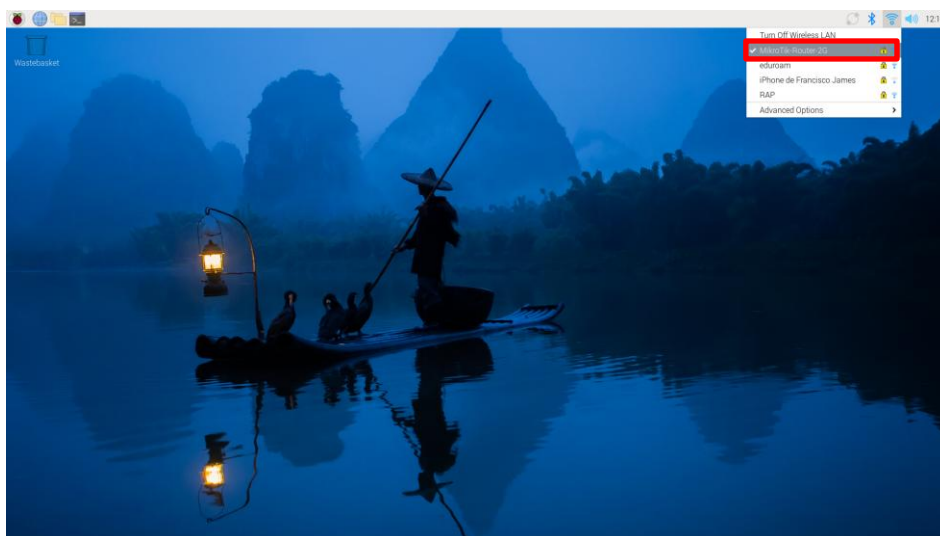
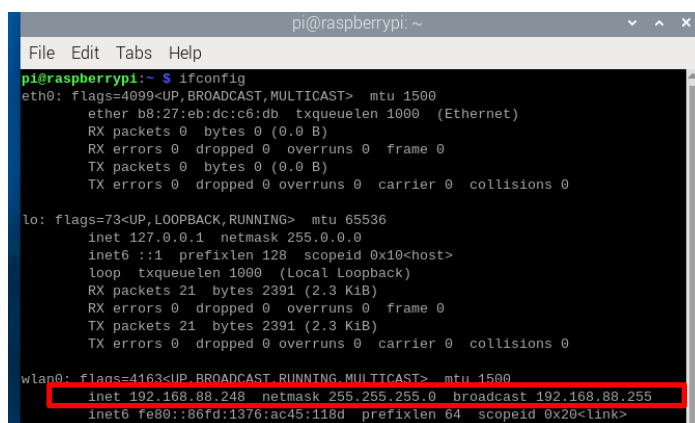


Figura 3-11 Interfaz gráfica Raspberry Pi Modelo 3 B y conexión a WiFi

Una vez establecida la conexión, se ha comprobado la asignación de direcciones IP por parte del servidor DHCP del router, verificando que cada Raspberry Pi ha obtenido una dirección única dentro del

rango definido en la configuración del router. Para ello, se ha ejecutado el comando *ifconfig*, de esta manera obtenemos la dirección IP de cada Raspberry, como se puede ver en la Figura 3-12.



```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether b8:27:eb:dc:c6:db txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 21 bytes 2391 (2.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 21 bytes 2391 (2.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.88.248 netmask 255.255.255.0 broadcast 192.168.88.255
    inet6 fe80::86fd:1376:ac45:118d prefixlen 64 scopeid 0x20<link>
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
  
```

Figura 3-12 Comando *ifconfig* para resaltar características de red

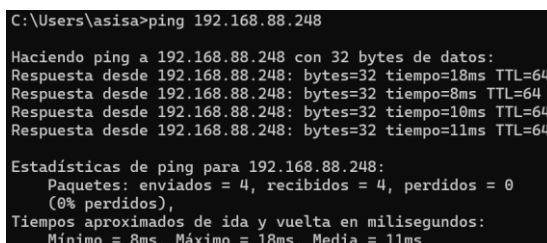
Realizando este comando en ambas Raspberry (accediendo a su terminal) sabemos que sus IP son:

- Raspberry 1: 192.168.88.248
- Raspberry 2: 192.168.88.246

Para asegurar la comunicación con el ordenador central y entre las propias Raspberry Pi, se ha utilizado el siguiente comando:

```
ping direcciónIP
```

Este comando se trata de una herramienta fundamental para diagnosticar y comprobar conexiones en una red. Se envían paquetes de prueba entre dispositivos y mide el tiempo que tardan en recibir una respuesta, lo que permite verificar si la conexión es estable y si hay pérdidas de datos. En la Figura 3-13 se puede observar un ejemplo de este comando.



```

C:\Users\asisa>ping 192.168.88.248

Haciendo ping a 192.168.88.248 con 32 bytes de datos:
Respuesta desde 192.168.88.248: bytes=32 tiempo=18ms TTL=64
Respuesta desde 192.168.88.248: bytes=32 tiempo=8ms TTL=64
Respuesta desde 192.168.88.248: bytes=32 tiempo=10ms TTL=64
Respuesta desde 192.168.88.248: bytes=32 tiempo=11ms TTL=64

Estadísticas de ping para 192.168.88.248:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 8ms, Máximo = 18ms, Media = 11ms
  
```

Figura 3-13 Comando *ping* desde ordenador a Raspberry Pi para comprobar conexión

La captura del resultado obtenido muestra como los cuatro paquetes enviados han sido recibidos correctamente sin pérdidas, lo que confirma que la conexión es estable. Además, los tiempos de respuesta oscilan entre 8 ms y 18 ms, con una media de 11 ms, lo que indica una latencia baja, adecuada para la comunicación dentro de la red local. El valor TTL (*Time To Live*) igual a 64 refleja que el dispositivo de destino está accesible sin atravesar múltiples nodos intermedios.

Al recibir respuestas sin errores ejecutando este comando sobre ambas direcciones IP de las Raspberry, se ha confirmado que la red está correctamente establecida y que cada dispositivo puede enviar y recibir datos de manera fluida.

3.3.2.3 Acceso remoto mediante comando *ssh*

Hasta ahora hemos manejado las Raspberry desde su interfaz, teniendo que conectar monitor, teclado y ratón, cosa que no es eficiente si queremos realizar movimientos rápidos entre dispositivos. Para evitarlo, se ha habilitado el acceso remoto mediante SSH (*Secure Shell*). SSH es un protocolo que

permite conectarse a otro dispositivo y controlarlo a través de una terminal, lo que resulta esencial en el proyecto, ya que permite administrar las Raspberry Pi directamente desde el ordenador central.

En primer lugar, el servicio ha de instalarse en las Raspberry Pi, pues en el portátil ya viene preinstalado. Se instala mediante el siguiente comando:

```
sudo apt update && sudo apt install openssh-server -y.
```

Posteriormente, se ha comprobado su funcionamiento en las Raspberry Pi mediante el comando:

```
systemctl status ssh
```

Así, se ha comprobado que efectivamente el servicio está habilitado, como se puede observar en la Figura 3-14. Además, se pueden observar las últimas conexiones realizadas.

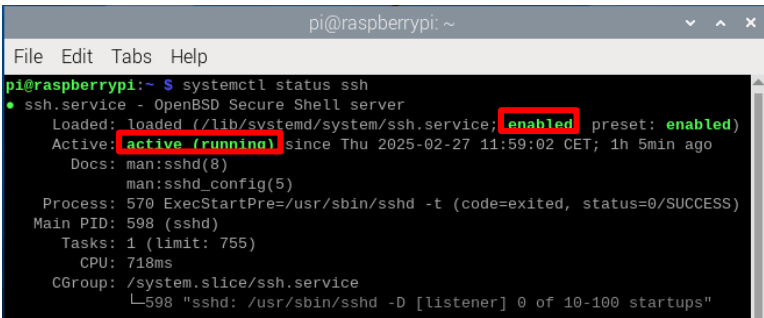


Figura 3-14 Comando ssh

Para acceder al servicio se utiliza el comando:

```
ssh usuario@dirección IP
```

Por ejemplo, si queremos acceder a la Raspberry Pi 1, usaremos el comando `ssh pi@192.168.88.248`. Tras introducir la contraseña de acceso, se consigue acceder a la terminal de la Raspberry Pi sin necesidad de periféricos adicionales, permitiendo ejecutar cualquier comando o *script* directamente desde el sistema de procesamiento (el portátil).

Esta configuración ha optimizado el flujo de trabajo, ya que todas las tareas pueden ser gestionadas de manera remota sin interferencias ni necesidad de cambiar constantemente de dispositivo.

De esta manera, podemos afirmar la configuración exitosa de nuestra red de sensores, tal y como se puede ver en la Figura 3-15, en donde accediendo nuevamente a la dirección IP del router se confirma que cada dispositivo ha obtenido una dirección IP válida dentro del rango definido por el servidor DHCP del router y que la conexión inalámbrica se mantiene estable. Además, la intensidad de la señal indica que los dispositivos están dentro de un rango adecuado de cobertura, garantizando una transmisión de datos sin interrupciones. Con esta infraestructura establecida, la red está lista para avanzar a la siguiente fase, donde se implementará la transmisión y almacenamiento del audio capturado por los sensores acústicos.

	MAC Address	In ACL	Last IP	Uptime		Signal Strength
	10:68:38:05:1B:22	no	192.168.88.247	00:12:42	-27	
	B8:27:EB:89:93:8E	no	192.168.88.248	00:17:11	-27	
	B8:27:EB:E8:8F:48	no	192.168.88.246	00:16:57	-45	

Figura 3-15 Estado de conexión de los dispositivos en la red de sensores

3.4 Segunda fase: Recopilación, transmisión y recepción de datos

El siguiente paso en el desarrollo del sistema es la implementación de un mecanismo de captura, transmisión y recepción de audio en tiempo real, asegurando que los datos acústicos sean enviados desde las Raspberry Pi hasta el ordenador central sin interrupciones. Este proceso será fundamental, ya que permitirá que el sistema de procesamiento reciba información de manera continua y pueda analizarla posteriormente sin depender de archivos de audio previamente almacenados.

3.4.1 Captura y transmisión de sonido desde las Raspberry Pi

Cada Raspberry Pi contará con un micrófono conectado a uno de sus puertos USB, encargado de captar las ondas sonoras del entorno y convertirlas en una señal digital. Antes de gestionar la captura de audio, se debe configurar este micrófono y asegurarse de que la entrada de sonido está realizándose correctamente. Para comprobarlo, se utiliza el siguiente comando:

```
arecord -l
```

Como se puede observar en la Figura 3-16. Este comando nos muestra una lista de los dispositivos conectados a la Raspberry Pi. Así, verificamos que el audio está entrando por el micrófono que tenemos conectado por USB, el *TONOR TC30*.

```
pi@raspberrypi:~$ arecord -l
**** List of CAPTURE Hardware Devices ****
card 2: Device [TONOR TC30 Audio Device], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

Figura 3-16 Configuración micrófono

Para gestionar esta captura de sonido, se implementará el *script* denominado “Send_audio.py” (Anexo III: Código de envío de audio desde raspberry pi), distinguiéndose una Raspberry Pi de otra en función del puerto escogido para la recepción de audio en el sistema de procesamiento (ordenador). Es decir, cada una tendrá un *script* distinto en el que lo único que cambia es el puerto al que envían la información. El uso de puertos distintos para recibir datos en el ordenador permitirá gestionar de manera separada los datos provenientes de cada sensor acústico, facilitando su identificación y procesamiento.

El *script* mencionado consta de las características mostradas en la Tabla 3-2.

Parámetro	Valor
Formato de audio	16 bits por muestra
Canal	Mono
Frecuencia de muestreo	44,1 kHz (estándar en procesamiento de audio)
Fragmentos de audio	1024 muestras por fragmento
Puertos	Raspberry Pi 1 (192.168.88.248): 5000
	Raspberry Pi 2 (192.168.88.246): 5001

Tabla 3-2 Parámetros captura de audio

Cabe destacar que dicho *script* ha sido realizado empleando Visual Studios desde el ordenador. Para pasarlo de un dispositivo a otro, se ha utilizado el siguiente comando:

```
scp send_audio.py usuario@DirecciónIP: /home/pi/Desktop
```

El uso de este comando consiste en poner la dirección IP y el nombre del archivo, compartiéndose este por medio de la conexión WiFi, un ejemplo más de la eficiente red que se ha montado y que permite el intercambio y comunicación rápida de información entre dispositivos.

Los *scripts* están diseñados para capturar estos fragmentos de audio y enviarlos mediante el protocolo UDP, un protocolo de transporte que no establece conexión previa y permite la transmisión rápida de datos, lo que lo hace ideal para aplicaciones en tiempo real como la transmisión de audio. Aplicamos este protocolo mediante el uso de la ya mencionada herramienta de *socket*. La elección de estos parámetros garantiza que el sistema pueda operar con un equilibrio entre calidad de audio y eficiencia en la transmisión de datos.

3.4.1.1 Flujo de ejecución del script

El *script* sigue el siguiente flujo de ejecución:

1. Inicialización del micrófono y configuración de audio: Se crea una instancia de la librería *pyaudio* y se abre un flujo de audio con los parámetros previamente definidos, garantizando la captura del sonido en tiempo real.

```
audio=pyaudio.PyAudio()
stream=audio.open(format=FORMAT, channels=CHANNELS, rate=RATE,
                  input=True, frames_per_buffer=CHUNK)
```

Siendo *format* los 16 bits por muestra, *channels* el canal mono, *rate* la frecuencia de muestreo y *chunk* el tamaño del búfer, que como ya hemos comentado, es 1024 muestras por fragmento

2. Configuración del *socket* UDP: Se establece un socket de comunicación mediante el protocolo UDP, asignando la dirección IP del ordenador central (192.168.88.247) y el puerto correspondiente, asegurando la correcta transmisión de datos.

```
SERVER_IP = "192.168.88.247"
PORT = 5000
```

Después de configurarlo, se crea el *socket* UDP mediante:

```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

A partir de este momento, aparecerá un mensaje en pantalla indicando “Transmitiendo audio, presione Ctrl + C para detener”

3. Inicio y mantenimiento de la transmisión: El sistema entra en un bucle continuo donde se capturan fragmentos de audio en bloques de 1024 muestras. Cada fragmento es enviado de inmediato al ordenador receptor, manteniendo una transmisión fluida y sin retrasos significativos.

```
try:
    while True:
        data = stream.read(CHUNK)
        sock.sendto(data, (SERVER_IP, PORT))
except KeyboardInterrupt:
```

4. Detención y liberación de recursos: Si el usuario interrumpe el proceso con Ctrl + C, el sistema finaliza correctamente cerrando el flujo de audio, liberando los recursos de *pyaudio* y cerrando el *socket*, evitando bloqueos o errores en futuras ejecuciones. Se mostrará en pantalla un mensaje indicando “Deteniendo transmisión”.

```
stream.stop_stream()
stream.close()
audio.terminate()
sock.close()
```

De esta manera, al ejecutar el *script* en ambas Raspberry Pi, a las cuales hemos accedido mediante el comando *ssh*, se comienza a “escuchar” y a transmitir lo “escuchado” al ordenador. Cuando queremos terminar de transmitir, intervenimos con el teclado del ordenador, presionando “Ctrl + C”. En la Figura 3-17 se puede ver ilustrado el mensaje que aparece en el terminal de las Raspberry Pi para indicar tanto el inicio como el final de un audio transmitido.

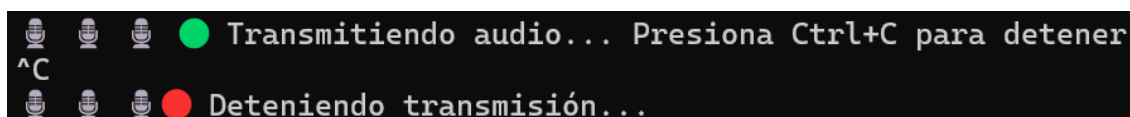


Figura 3-17 Captura y transmisión de audio en las Raspberry Pi

3.4.2 Recepción del audio en el sistema de procesamiento

El ordenador central desempeña un papel clave en la recepción y gestión del audio transmitido en tiempo real por las Raspberry Pi. Para lograr esta tarea, se ha desarrollado el *script* “Receive_tiempo_real”, diseñado para recibir y gestionar el audio transmitido en tiempo real. Este *script* se puede observar detalladamente en el Anexo IV: Código de recepción de audio en tiempo real.

El sistema está diseñado para recibir audio desde dos Raspberry Pi de manera simultánea, utilizando el protocolo UDP y la librería *socket*. Para ello, se emplea otra librería, *threading*, la cual permite procesar la recepción de cada dispositivo en paralelo sin afectar el rendimiento. Además, como era de esperar, cuenta con los mismos parámetros de recepción de audio que los presentados en la Tabla 3-2, que eran de captura de audio.

Para gestionar la transmisión en tiempo real, se han establecido ventanas temporales de 5 segundos. Dentro de cada ventana, los datos recibidos son almacenados y, una vez finalizado el tiempo definido, se genera un archivo en formato WAV con un nombre estructurado que permite identificar la fuente y la temporalidad del audio.

Esto permite que posteriormente se pueda analizar el audio recibido con referencia temporal exacta. De esta manera, el proceso de almacenamiento incluye:

1. Recepción del búfer de audio desde el *socket* asignado a cada Raspberry Pi.
2. Acumulación de datos en una lista de *frames* hasta completar la ventana de 5 segundos.
3. Escritura del archivo de audio en la carpeta de almacenamiento (el operario crea una manualmente con el día correspondiente) con el formato RaspberryPiX_HH-MM-SS.wav

Así, un archivo con el nombre *RaspberryPi1_19-23-45* indica que:

- Se origina desde la Raspberry Pi 1.
- Fue grabado a las 19:23:45 del día al que corresponda la carpeta.

Al ejecutar el *script*, el sistema inicia el conjunto de *sockets* que quedan a la espera de recibir audio desde diferentes Raspberry Pi a través de la red. Estos datos se capturan en tiempo real, se almacenan en archivos de audio y se gestionan en intervalos predefinidos. Además, cada 30 minutos el programa

se pausa para que el operario decida si desea continuar o finalizar la recepción. El funcionamiento del sistema se organiza en varias secciones clave, que vamos a desglosar paso a paso.

3.4.2.1 Definición de variables y configuración inicial

El primer paso en el *script* es definir varias configuraciones clave para la captura de audio y la gestión de la red. Se establecen los siguientes parámetros: la calidad y formato del audio, la frecuencia de muestreo, el tamaño de cada fragmento, la duración de cada sesión y los puertos de escucha.

```
FORMAT = pyaudio.paInt16
CHANNELS = 1
RATE = 44100
CHUNK = 1024
WINDOW_TIME = 5
SESSION_DURATION = 1800
```

Como se puede observar, son los mismos parámetros que los establecidos anteriormente en la Tabla 3-2.

Además, se define un diccionario *ports*, que almacena los puertos en los que se recibirán los datos y los asocia con el nombre de cada Raspberry Pi transmisora (*device_name*). Esto abre el camino a que se puedan integrar más Raspberry Pi en el futuro, pues simplemente se añadirían otros puertos.

```
ports = {5000: "RaspberryPi1", 5001: "RaspberryPi2"}
```

Otro elemento fundamental es la variable *output_folder*, donde se almacenarán los archivos de audio. Cada día tendrá su propia carpeta que debe ser creada manualmente por el operario antes de ejecutar el *script*. Por ejemplo, esta sería la carpeta dentro de “Audio Recibido” del día 12 de marzo de 2025.

```
output_folder = "Audio Recibido/12-03-2025"
```

3.4.2.2 Inicio en la recepción del audio.

Al ejecutar el *script*, el sistema inicializa los *sockets* correspondientes y entra en un estado de espera para recibir transmisión desde las Raspberry Pi. Esto se consigue mediante la función “*receive_audio*”. Esta función se encarga de recibir el audio transmitido desde una Raspberry Pi a través de la red, almacenarlo en un archivo WAV cada cierto tiempo y detectar si la transmisión sigue activa o se ha interrumpido. Funciona en paralelo para cada dispositivo que envía audio, manejando diferentes puertos de manera independiente.

Primero definimos la función, que recibe como parámetros el puerto del ordenador que va a recibir el audio y la Raspberry Pi que va a transmitirlo, variables descritas previamente.

```
def receive_audio(port, device_name):
```

Dentro de esta función, lo primero que se hace es definir un *socket* tipo UDP, utilizando la función

```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

Lo que indica que se trabajará con direcciones IPv4 y datagramas en la transmisión. Este *socket* se almacena en un diccionario global, permitiendo su acceso desde otros puntos del programa, mediante:

```
sockets[port] = sock
```

Luego, el *socket* se vincula al puerto especificado, quedando en espera de paquetes de audio entrantes. Además, se define un tiempo de espera de 1 segundo utilizando *sock.settimeout(1)*. Este ajuste es importante porque evita que el *socket* quede bloqueado indefinidamente si deja de recibir datos.

Esta secuencia se consigue mediante los siguientes comandos:

```
sock.bind(("", port))
sock.settimeout(1)
```

De esta manera, mientras no haya transmisión activa, se muestra un mensaje de “esperando transmisión” para cada puerto, como se puede observar en la Figura 3-18.

```
2025-03-07 10:51:53,577 - INFO - 🟢🟢 Sistema de recepción en tiempo real activo. Presiona Ctrl+C para detener.
2025-03-07 10:51:53,577 - INFO - 🟡 Esperando transmisión desde RaspberryPi1 en el puerto 5000...
2025-03-07 10:51:53,577 - INFO - 🟡 Esperando transmisión desde RaspberryPi2 en el puerto 5001...
```

Figura 3-18 Inicio de sistema de detección en tiempo real

Cuando iniciamos la transmisión de alguna de las Raspberry Pi (desde su correspondiente terminal), el sistema lo detecta y cambia el estado a “transmitiendo”, indicando en la consola que ha comenzado a recibir audio desde el dispositivo correspondiente.

A partir de este momento, se empiezan a acumular los paquetes de audio recibidos en memoria. Los fragmentos de audio se van almacenando en una lista:

```
frames = []
```

También se establece una variable para registrar el momento de inicio de la recepción para controlar el tiempo transcurrido y determinar cuándo guardar el audio en un archivo, además de para saber cuándo termina la sesión. Esto se hace con:

```
start_time = time.time()
```

A continuación, la función entra en su bucle principal, que se ejecutará mientras la variable *running* se mantenga activa.

Cuando un paquete de audio llega correctamente, el sistema lo almacena y marca la transmisión como activa:

```
try:
    data, addr = sock.recvfrom(CHUNK * 2)
    active_streams[port] = True
    if not started_transmitting:
        logging.info(f"{device_name} ha comenzado a transmitir en el puerto {port}.")
        started_transmitting = True
        frames.append(data)
```

La variable *started_transmitting* es clave en este proceso, ya que evita que el sistema muestre múltiples veces el mensaje de inicio de transmisión. Solo cuando el primer paquete de audio es recibido se indica en el log que la Raspberry Pi ha comenzado a transmitir.

Por otro lado, si después de un tiempo definido en `sock.settimeout(1)` no se recibe ningún paquete (si dejamos de transmitir intencionadamente con la Raspberry Pi), el sistema asume que la transmisión se ha interrumpido y emite una advertencia:

```
except socket.timeout:
    if active_streams[port]:
        logging.warning(f" No se detecta transmisión de {device_name}
        en {port}.")
        started_transmitting = False
        active_streams[port] = False
```

3.4.2.3 Almacenamiento del audio

Cada cierto tiempo, el audio recibido se guarda en un archivo de sonido en la carpeta definida por el operario. Para ello, se establece un intervalo de 5 segundos, tras los cuales los datos acumulados en la lista *frames* se guardan en un archivo WAV con el nombre descrito anteriormente. Esto se consigue mediante el siguiente condicional:

```
if time.time() - start_time >= WINDOW_TIME:
    timestamp = time.strftime("%H-%M-%S")
    filename=os.path.join(output_folder,f"{device_name}_{timestamp}
    .wav")
```

Para escribir el archivo WAV, se usa la librería *wave*, que permite especificar el formato del audio (canales, frecuencia de muestreo, tamaño de las muestras) y escribir los datos almacenados en *frames*.

```
with wave.open(filename, 'wb') as wf:
    wf.setnchannels(CHANNELS)
    wf.setsampwidth(pyaudio.PyAudio().get_sample_size(FORMAT))
    wf.setframerate(RATE)
    wf.writeframes(b''.join(frames))
```

Una vez guardado el archivo, se vacía la lista *frames* para comenzar a almacenar nuevos datos, y se reinicia el temporizador *start_time*.

```
frames = []
start_time = time.time()
```

3.4.2.4 Finalización y reinicio de la sesión

El sistema está diseñado para ejecutarse en sesiones de 30 minutos. Cuando este tiempo se cumple, el programa detiene la recepción de audio y solicita al operario que indique si desea continuar.

```
logging.info("Sesión finalizada. ¿Desea continuar? (Sí = Reiniciar,
No = Salir)")
user_input = input("Presiona 's' para continuar o cualquier otra
tecla para salir: ").strip().lower()
```

Si el operario decide continuar, el programa se reinicia por completo, ejecutando de nuevo el *script* desde el inicio con:

```
os.execv(sys.executable, ['python'] + sys.argv)
```

En caso contrario, el programa finaliza de manera ordenada con:

```
sys.exit(0)
```

De esta manera nos aseguramos de que todos los *sockets* abiertos se cierran correctamente, avisando al usuario por pantalla.

3.4.2.5 Ejemplo implementación

Para comprender mejor el funcionamiento, a continuación, se muestra un ejemplo práctico. Pongamos que la Raspberry Pi 1, después de ejecutar su código de *send_audio*, transmite durante 20 segundos. Esta sería la secuencia:

1. Nos informa de que comienza a recibir en el puerto 5000 (correspondiente a la Raspberry Pi 1).
2. Genera un archivo inicial de 0 segundos como inicio de las grabaciones y su almacenamiento.
3. Posteriormente guarda cuatro archivos de 5 segundos. Hasta que insertamos, 20 segundos después de su inicio y desde el terminal de la Raspberry, “Control + C” como finalización de la transmisión, como se ha visto anteriormente en el apartado de captura de audio.
4. Después de 1 segundo en el que el sistema no recibe paquetes en el puerto, advierte al operador.

Esta secuencia se puede observar en la Figura 3-19:

```
2025-03-12 18:13:18,263 - INFO - 🟢🟢🟢 Sistema de recepción en tiempo real activo. Se detendrá en 30 minutos.
2025-03-12 18:13:18,263 - INFO - 🟠 Esperando transmisión desde RaspberryPi2 en el puerto 5001...
2025-03-12 18:13:18,263 - INFO - 🟠 Esperando transmisión desde RaspberryPi1 en el puerto 5000...
2025-03-12 18:17:24,580 - INFO - 🟢 RaspberryPi1 ha comenzado a transmitir en el puerto 5000.
2025-03-12 18:17:24,984 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi1_18-17-24.wav
2025-03-12 18:17:29,995 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi1_18-17-29.wav
2025-03-12 18:17:35,012 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi1_18-17-35.wav
2025-03-12 18:17:40,023 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi1_18-17-40.wav
2025-03-12 18:17:45,078 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi1_18-17-45.wav
2025-03-12 18:17:47,661 - WARNING - ⚠ No se detecta transmisión de RaspberryPi1 en 5000.
```

Figura 3-19 Ejemplo de transmisión y recepción de audio en la Raspberry Pi 1

Probamos la misma secuencia con la Raspberry 2 para comprobar su funcionamiento. Probamos nuevamente con 20 segundos de transmisión. La única diferencia será apreciable en el nombre del archivo, pues indicará RaspberryPi2 en lugar de RaspberryPi1. Una vez transcurrido el tiempo establecido, anulamos la transmisión desde el terminal de la Raspberry. Este proceso se puede observar de nuevo en la Figura 3-20:

```
2025-03-12 18:21:35,282 - INFO - 🟢 RaspberryPi2 ha comenzado a transmitir en el puerto 5001.
2025-03-12 18:21:35,285 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi2_18-21-35.wav
2025-03-12 18:21:40,305 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi2_18-21-40.wav
2025-03-12 18:21:45,339 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi2_18-21-45.wav
2025-03-12 18:21:50,372 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi2_18-21-50.wav
2025-03-12 18:21:55,411 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi2_18-21-55.wav
2025-03-12 18:21:59,700 - WARNING - ⚠ No se detecta transmisión de RaspberryPi2 en 5001.
```

Figura 3-20 Ejemplo de transmisión y recepción de audio en la Raspberry Pi 2

Se procede ahora a la prueba más relevante hasta el momento: la transmisión simultánea de ambas Raspberry Pi, con el objetivo de comprobar que el sistema de procesamiento es capaz de recibir, gestionar y almacenar el audio procedente de ambas fuentes de manera continua, sin interrupciones. Para ello, se ejecutan los *scripts* de transmisión en ambas Raspberry Pi de forma simultánea, iniciando el envío de datos hacia el ordenador. La Figura 3-21 muestra esta secuencia.

```

2025-03-12 18:33:13,968 - INFO - 🟢 RaspberryPi1 ha comenzado a transmitir en el puerto 5000.
2025-03-12 18:33:13,983 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi1_18-33-13.wav
2025-03-12 18:33:16,347 - INFO - 🟢 RaspberryPi2 ha comenzado a transmitir en el puerto 5001.
2025-03-12 18:33:16,362 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi2_18-33-16.wav
2025-03-12 18:33:19,002 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi1_18-33-18.wav
2025-03-12 18:33:21,392 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi2_18-33-21.wav
2025-03-12 18:33:24,020 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi1_18-33-24.wav
2025-03-12 18:33:26,428 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi2_18-33-26.wav
2025-03-12 18:33:29,053 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi1_18-33-29.wav
2025-03-12 18:33:31,462 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi2_18-33-31.wav
2025-03-12 18:33:34,083 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi1_18-33-34.wav
2025-03-12 18:33:36,496 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi2_18-33-36.wav
2025-03-12 18:33:39,088 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi1_18-33-39.wav
2025-03-12 18:33:41,525 - INFO - 📁 Archivo guardado: Audio Recibido/12-03-2025\RaspberryPi2_18-33-41.wav
2025-03-12 18:33:43,742 - WARNING - ⚠️ No se detecta transmisión de RaspberryPi1 en 5000.
2025-03-12 18:33:44,972 - WARNING - ⚠️ No se detecta transmisión de RaspberryPi2 en 5001.
2025-03-12 18:43:18,278 - INFO - 🛑 Sesión finalizada. ¿Desea continuar? (Sí = Reiniciar, No = Salir)
Presiona 's' para continuar o cualquier otra tecla para salir:
k
2025-03-12 18:43:26,798 - INFO - ✅ Finalizando el programa.

```

Figura 3-21 Transmisión simultánea de ambas Raspberry Pi

Durante la prueba, se realiza una transmisión de 25 segundos. Como se observa en la Figura 3-21, el sistema guarda cinco archivos por cada Raspberry Pi, sin contar los dos primeros archivos generados al inicio, que sirven para marcar el comienzo del almacenamiento. Estos cinco archivos corresponden a las ventanas de cinco segundos grabadas durante ese periodo de tiempo. Después de esos 25 segundos, tras detener la transmisión, el sistema emite una advertencia indicando que ya no se detecta señal en los puertos correspondientes, lo que confirma el correcto funcionamiento del sistema en condiciones de transmisión simultánea. Finalmente, 30 minutos después de comenzar la transmisión (observar Figura 3-19 y su inicio para comprobar que efectivamente son 30 minutos después), el sistema nos pregunta si queremos continuar o reiniciar. Al introducir una tecla distinta a la s, finalizamos el programa.

Podemos comprobar el almacenamiento de los audios en la Figura 3-22, los cuales serán utilizados posteriormente para el análisis del sonido recibido en tiempo real.

- 📁 RaspberryPi2_18-33-41
- 📁 RaspberryPi1_18-33-39
- 📁 RaspberryPi2_18-33-36
- 📁 RaspberryPi1_18-33-34
- 📁 RaspberryPi2_18-33-31
- 📁 RaspberryPi1_18-33-29
- 📁 RaspberryPi2_18-33-26
- 📁 RaspberryPi1_18-33-24
- 📁 RaspberryPi2_18-33-21
- 📁 RaspberryPi1_18-33-18

Figura 3-22 Almacenamiento en formato WAV

El sistema ha sido optimizado para mejorar su estabilidad y eficiencia, asegurando que la recepción de audio sea continua y sin interrupciones. Gracias a la implementación de *threading*, es posible manejar ambas transmisiones de manera simultánea sin bloqueos, permitiendo también añadir nuevas Raspberry Pi en un futuro sin tener que realizar cambios significativos en el código. Además, se han incorporado mensajes informativos y de advertencia que permiten supervisar el estado del sistema en todo momento. Esta estructura asegura una recepción confiable de los datos y facilita el procesamiento posterior para la detección del dron en tiempo real. Ahora, el ordenador central recibirá continuamente la información acústica de las Raspberry Pi, permitiendo que el análisis se realice de manera más eficiente. En la siguiente fase del desarrollo, se abordará la integración del sistema de detección de drones, aplicando algoritmos de análisis sobre los segmentos de audio capturados en tiempo real.

3.5 Tercera fase: Análisis de datos

El propósito de esta fase es analizar en profundidad la huella acústica del dron para identificar los patrones más relevantes en su señal sonora. A partir de este análisis, se determinarán los elementos clave que permitirán definir, en la siguiente fase, los filtros de frecuencia y umbrales de detección adecuados. Para ello, se han aplicado diversas técnicas de procesamiento de señales, generando espectrogramas, distribuciones de frecuencia y análisis de intensidad en distintas condiciones de vuelo del dron.

Tras haberse comprobado el correcto funcionamiento del sistema de recepción de audio, para realizar este análisis se han hecho grabaciones del dron en distintas condiciones. Todas las grabaciones tienen una longitud aproximada de 30 segundos, lo que nos permite hacer un estudio profundo de sus características. Los audios obtenidos son los siguientes y se pueden escuchar accediendo al QR contenido en el Anexo XII: QR para audios del UAV grabados desde TONOR TC30.

- Dron a 1,5 metros del micrófono.
- Dron a 3 metros del micrófono.
- Dron a 5 metros del micrófono.
- Dron a 10 metros del micrófono.
- Dron a 15 metros del micrófono.
- Recorrido del dron por toda la sala.

Con estos audios se ha recopilado la información necesaria para analizar la huella acústica del dron. Se trata de la fase de “inteligencia”, en la que obtenemos los patrones característicos de un dron conocido. Se han seguido los siguientes pasos.

3.5.1 Análisis espectral

El análisis espectral permite visualizar la distribución de la energía de la señal en función del tiempo y la frecuencia. Esta técnica es esencial para comprender cómo se comporta la huella acústica del dron y determinar en qué rangos de frecuencia su presencia es más evidente. Para ello, se ha generado una representación visual del contenido espectral de los audios mediante espectrogramas, utilizando el *script* “Espectro.py” (Anexo V: Código de generación de espectrogramas)

3.5.1.1 Flujo de ejecución del script

En el inicio de este *script* se cargan los archivos de audio en formato WAV, asegurando que todos los datos sean procesados correctamente en un formato compatible con las herramientas de análisis. Dado que algunos archivos pueden estar en estéreo, se procede a convertirlos a una señal mono. La diferencia entre ambos formatos radica en la forma en que se capturan y reproducen los sonidos. Un archivo estéreo contiene dos canales de audio, permitiendo una percepción espacial del sonido, mientras que un archivo mono tiene un único canal, concentrando toda la información en una sola pista. En este análisis, la conversión a mono es necesaria para evitar redundancias y facilitar la interpretación de los datos sin duplicidad de información, lo que permite evitar redundancias y facilitar la interpretación de los datos sin duplicidad de información.

Posteriormente, se aplica la Transformada de Fourier de Corto Plazo (STFT), una técnica que permite analizar la evolución de la señal en el dominio de la frecuencia a lo largo del tiempo. Esta transformación segmenta la señal en intervalos pequeños, aplicando la Transformada de Fourier en cada uno de ellos para obtener una representación en el dominio tiempo-frecuencia.

Con los datos obtenidos, se genera el espectrograma, una representación gráfica en la que la intensidad del color indica la energía presente en cada frecuencia y momento. Finalmente, se interpretan los resultados, identificando los rangos de frecuencia con mayor energía y observando si estos patrones se mantienen constantes en distintas condiciones de vuelo del dron.

3.5.1.2 Transformada de Fourier de Corto Plazo (STFT)

Antes de entrar en detalle con los espectrogramas obtenidos, se ofrece una explicación de la base de su generación. La STFT es una herramienta fundamental en el análisis de señales, ya que permite estudiar cómo varía la frecuencia de una señal a lo largo del tiempo.

Para su cálculo, la señal original se divide en segmentos de tamaño fijo, en nuestro caso de 1024 muestras. Cada uno de estos segmentos se multiplica por una función ventana deslizante que suaviza los extremos y reduce las discontinuidades en la transformación. Una vez aplicada la ventana, se procede a calcular la Transformada de Fourier en cada uno de los segmentos, obteniendo así la distribución espectral de la señal en ese instante de tiempo. Al repetir este proceso para cada segmento, se construye un gráfico conocido como espectrograma, donde el eje horizontal representa el tiempo, el eje vertical la frecuencia y la intensidad del color indica la amplitud de la señal en cada frecuencia y momento.

Matemáticamente, la STFT se define como:

$$STFT(x(t)) = X(t, f) = \int_{-\infty}^{\infty} x(\tau)w(\tau - t)e^{-j2\pi f\tau}d\tau \quad (1)$$

En la ecuación (1), $x(t)$ es la señal de audio de entrada, τ es la variable de integración, $w(\tau - t)$ es la ventana deslizante centrada en t y $e^{-j2\pi f\tau}$ es la base de la Transformada de Fourier.

Después de calcular la STFT, tomamos su módulo al cuadrado para obtener la densidad espectral de potencia, tal y como se muestra en la ecuación (2).

$$S_{xx}(f, t) = |\mathcal{F}\{x(t)\}|^2 \quad (2)$$

De esta manera obtenemos un espectrograma definido por:

- Eje de frecuencias (f): Contiene las frecuencias analizadas, que dependen de la frecuencia de muestreo del audio. La frecuencia máxima en f es la frecuencia de Nyquist, definida como la mitad de la frecuencia de muestreo.
- Eje de tiempos (t): Contiene los instantes de tiempo en los que se calcula la STFT.
- Matriz S_{xx} : nos indica cuánta energía hay en cada frecuencia f y en cada instante de tiempo t . Cada fila de S_{xx} corresponde a una frecuencia f , y cada columna representa un instante t .

En Python, simplemente utilizamos la función:

```
f, t, Sxx = spectrogram(audio_data, fs)
```

En donde *audio_data* son los datos del audio en forma de array y *fs* es la frecuencia de muestreo. Esta función calcula el espectrograma de una señal aplicando internamente la STFT y devolviendo directamente la densidad espectral de potencia en valores reales. En lugar de aplicar la función *stft*, que retorna valores complejos incluyendo fase, *spectrogram* nos proporciona la distribución de energía en cada frecuencia y tiempo, lo que facilita su interpretación sin necesidad de cálculos adicionales. En este análisis, hemos optado por *spectrogram* en lugar de *stft* porque nuestro objetivo es extraer información sobre la intensidad del sonido del dron, sin necesidad de reconstrucción de la señal ni manipulación de la fase, simplificando así el procesamiento de datos y la generación del espectrograma.

Cabe destacar que, en la función de espectrograma, los valores de *audio_data* provienen de la conversión de los *frames* de audio en enteros de 16 bits. Esto significa que los valores de amplitud en S_{xx} están en unidades arbitrarias de amplitud de la señal relacionadas con la codificación PCM (*Pulse Code Modulation*) del archivo WAV. Para obtener la representación en dB, se aplica la ecuación (3):

$$Sxx_{dB} = 10 \log_{10} Sxx \quad (3)$$

Donde los valores originales de Sxx representan la densidad espectral de potencia en unidades proporcionales al cuadrado de la amplitud. Antes de aplicar el logaritmo, los valores de Sxx están en unidades de potencia relativa, y al convertirlos con la ecuación anterior, los pasamos a decibelios (dB), que es una escala logarítmica utilizada para representar la intensidad del sonido de manera más comprensible.

3.5.1.3 Espectrogramas

El espectrograma proporciona una primera visión global de cómo se comporta la señal del dron en términos de frecuencia y tiempo. Esto permite establecer hipótesis sobre qué bandas frecuenciales son más relevantes y cómo se podría proceder en las siguientes fases del análisis. A continuación, en la Figura 3-23, se pueden observar los resultados con el dron a distintas distancias.

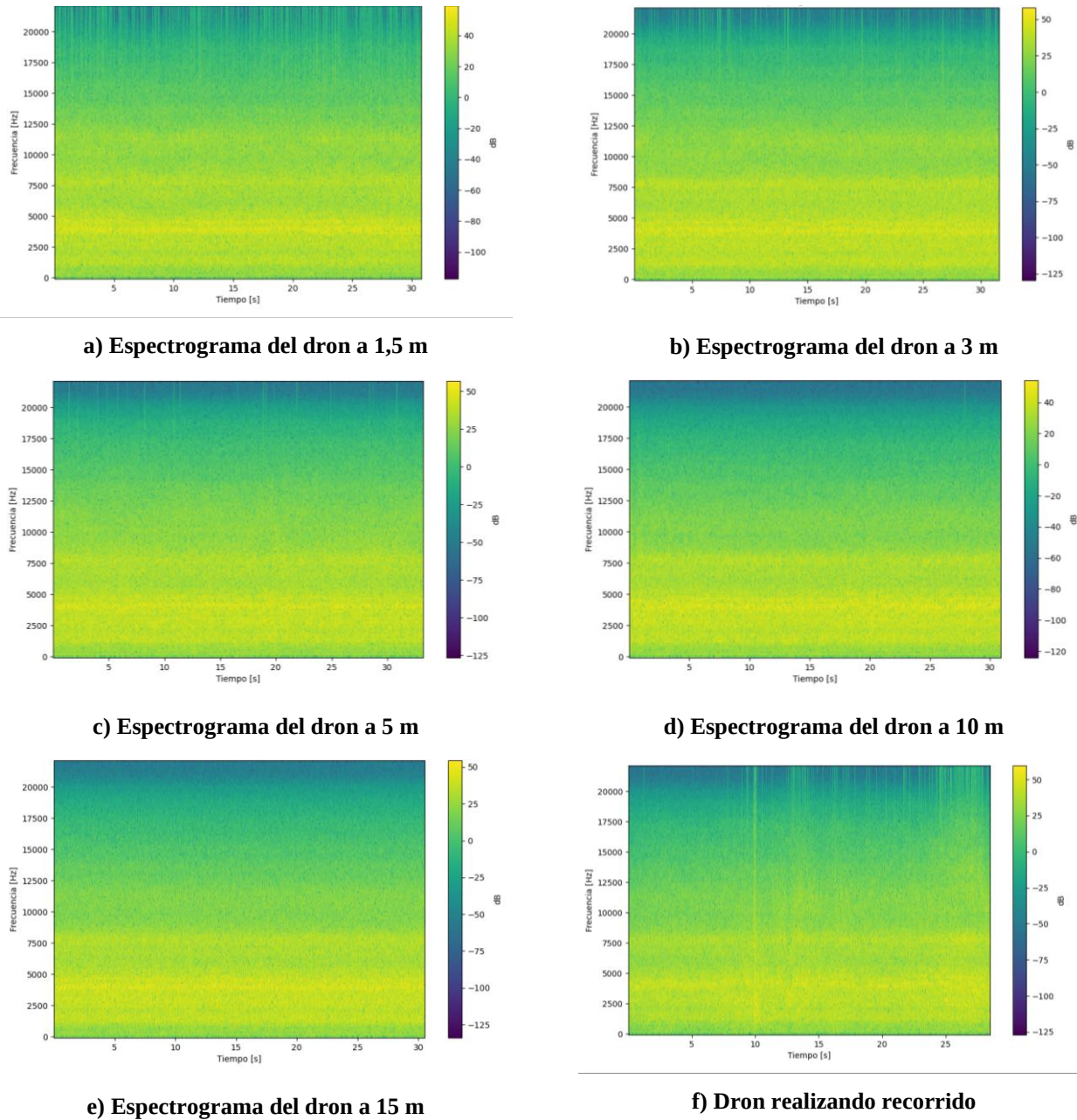


Figura 3-23 Espectrogramas del dron a distintas distancias del micrófono

Los espectrogramas obtenidos para distintas distancias muestran un patrón característico en el que las frecuencias predominantes se encuentran en un rango aproximado de 2500 Hz a 8000 Hz. A medida que la distancia entre el dron y el micrófono aumenta, se observa una atenuación en la intensidad de estas frecuencias, lo que sugiere que la señal del dron se debilita con la distancia. No obstante, incluso en distancias mayores, las frecuencias características del dron siguen siendo detectables, lo que indica la viabilidad del método de detección basado en la huella acústica.

Además, se observan armónicos adicionales en algunas grabaciones, lo que sugiere que el sonido del dron no se compone de una única frecuencia, sino de un conjunto de frecuencias que interactúan entre sí. También es notable la presencia de ruido ambiental en las grabaciones, lo que hace necesario el uso de técnicas de filtrado para mejorar la detección.

Para afinar aún más este análisis, pues un rango de 2500 a 8000 Hz es demasiado grande, se requiere realizar una evaluación más detallada de las frecuencias predominantes. Para ello, se procederá a realizar un análisis de distribución frecuencial en el que se representarán estas frecuencias en función de su presencia a lo largo de los distintos experimentos. Posteriormente, se generará un *boxplot* para visualizar la variabilidad de estas frecuencias y determinar con mayor precisión qué rangos son más representativos del sonido del dron. Este análisis permitirá establecer un filtro paso-banda óptimo en la siguiente fase del estudio.

3.5.2 Análisis frecuencial

El espectrograma permite visualizar la evolución de la señal en el tiempo, pero no ofrece un análisis cuantitativo sobre qué frecuencias tienen mayor presencia global en el audio. El análisis frecuencial de cada audio y la generación de un *boxplot* con un resumen general ayuda a identificar cuáles son las frecuencias con mayor concentración de energía en distintos momentos, proporcionando una base más objetiva para la selección de filtros en la fase siguiente. Para ello se han desarrollado dos enfoques implementados en dos nuevos *scripts*, “Análisis_frecuencia.py” (Anexo VI: Código de análisis frecuencial) y “Análisis_boxplot.py” (Anexo VII: Código de generación de *boxplot*). En la Figura 3-24 se puede observar el esquema de ejecución en cada uno.

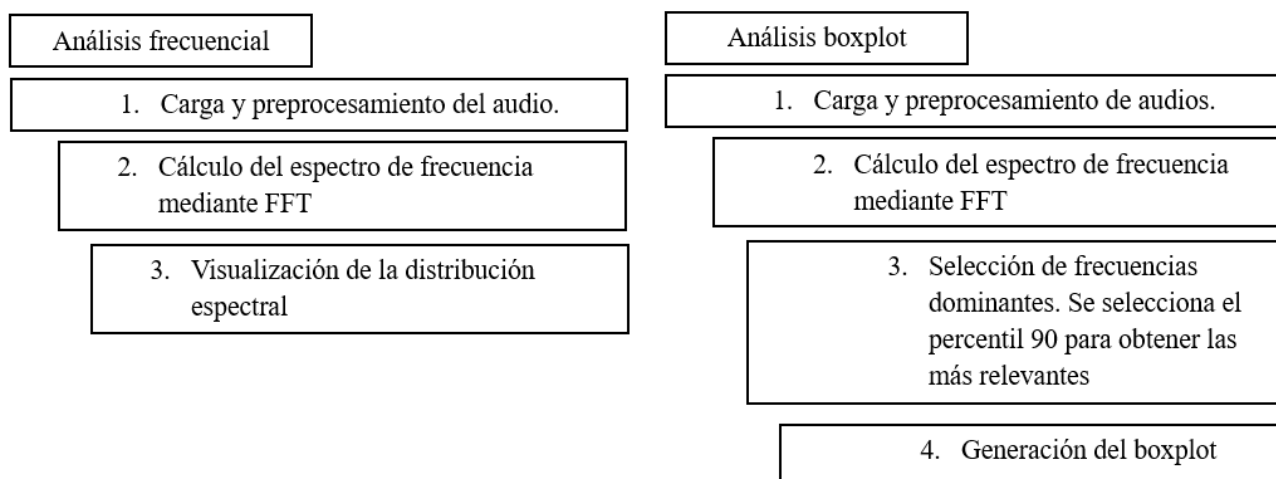


Figura 3-24 Secuencia del análisis frecuencial

En la Figura 3-25 se pueden observar los resultados obtenidos de los distintos audios del dron. Las gráficas presentadas muestran la distribución de frecuencias del sonido del dron a diferentes distancias, proporcionando información clave sobre cómo varía su huella acústica en función de la proximidad al micrófono. Al observar cada una de las gráficas, se pueden extraer algunas conclusiones importantes.

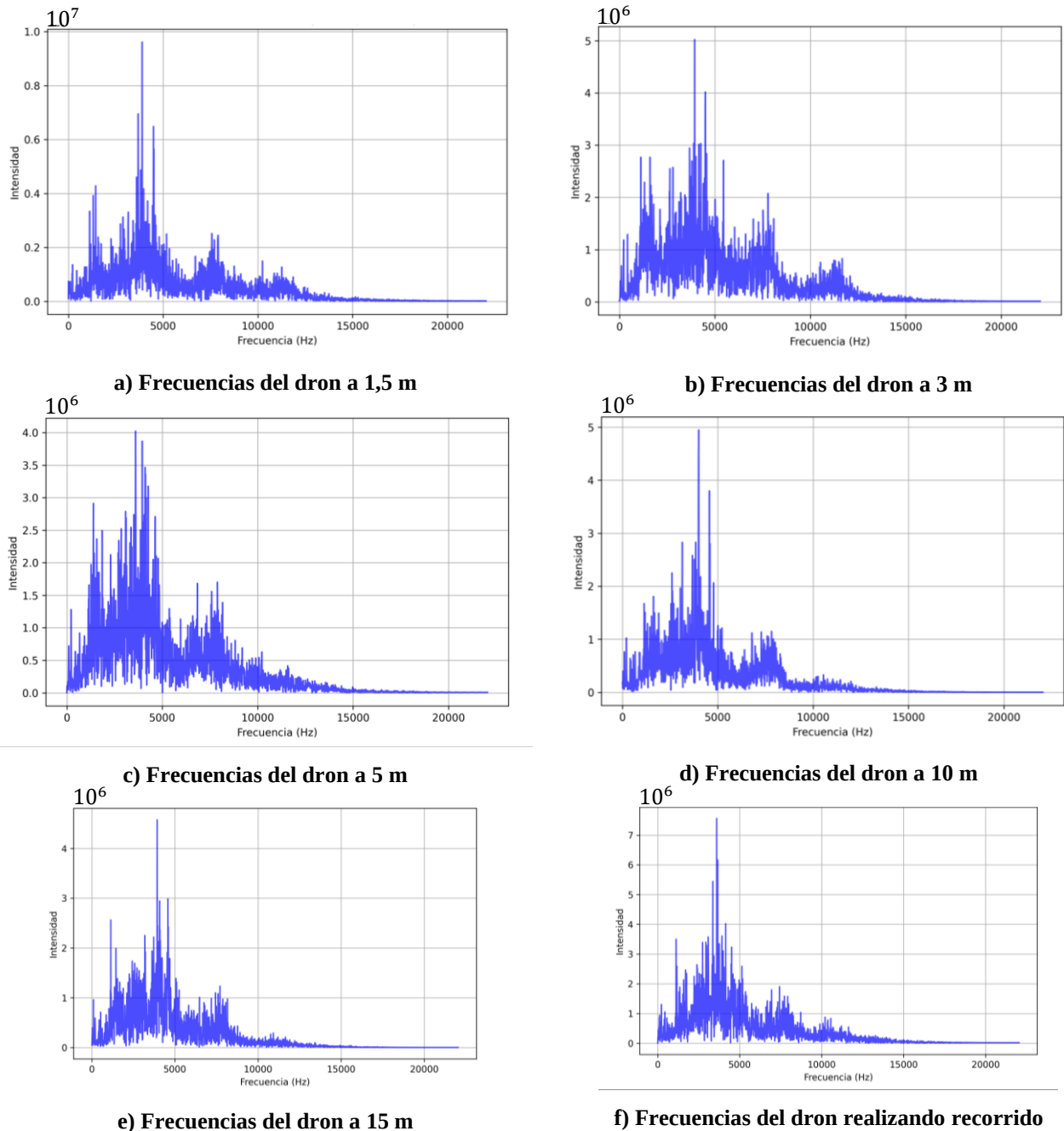


Figura 3-25 Distribución frecuencial de los distintos audios del dron

Las gráficas muestran con claridad cómo el sonido del dron evoluciona con la distancia, destacando patrones interesantes en la distribución de frecuencias. Desde los primeros metros, la señal es fuerte y bien definida, con picos claros en la región de 3000 Hz a 6000 Hz, una zona que parece ser clave en su huella acústica. A medida que el dron se aleja, la energía total de la señal se reduce notablemente, pero ese mismo rango de frecuencias sigue siendo el más persistente, lo que sugiere que podría utilizarse como una referencia estable para su detección.

Al comparar los registros tomados a diferentes distancias, se observa que las frecuencias dominantes no desaparecen, pero sí pierden intensidad y definición. En los audios más cercanos, los picos de energía están bien marcados, con una mayor diferencia entre las frecuencias más intensas y las que tienen menor presencia. Sin embargo, en los registros de 10 y 15 metros, la señal parece más difusa y mezclada con componentes de menor energía. Esto podría deberse no solo a la atenuación natural del sonido, sino también a la creciente influencia del ruido ambiental, que comienza a enmascarar parte de la señal.

original. Es especialmente llamativo cómo, aunque la estructura general del espectro se mantiene, la nitidez con la que se diferencian los armónicos se va perdiendo con la distancia.

El caso del audio tomado durante todo el recorrido del dron es particularmente revelador. Aquí, la intensidad de la señal es más baja en general, lo que sugiere que, al promediar diferentes posiciones en el espacio, los momentos en los que el dron estaba más cerca del micrófono quedan diluidos dentro del conjunto. Aun así, el patrón de frecuencias característico sigue presente, lo que refuerza la idea de que el dron deja una marca acústica reconocible, aunque con variaciones en intensidad según la posición y la distancia.

Esta distribución de frecuencias de cada audio la podemos juntar realizando el *boxplot* (Figura 3-26) comentado. Así, podemos hacer un análisis global de todos los audios juntos.

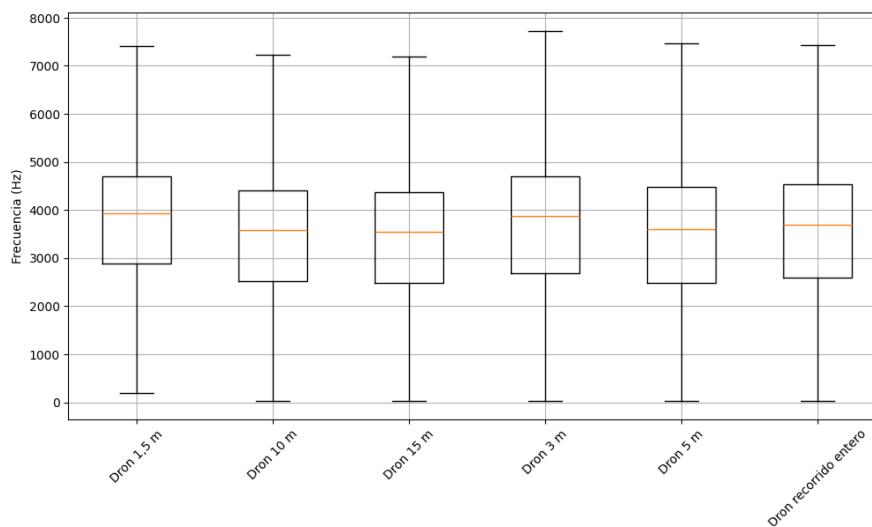


Figura 3-26 Distribución de frecuencias predominantes en los audios del dron

El *boxplot* generado permite visualizar con mayor claridad la distribución de las frecuencias predominantes en los distintos escenarios de prueba con el dron. Se puede observar que, en todas las condiciones, la mediana de las frecuencias predominantes se sitúa en torno a los 3500-4000 Hz, con una gran concentración de datos dentro del rango de 3000 a 6000 Hz. Esto coincide con los resultados obtenidos previamente en el análisis espectral y las distribuciones individuales de frecuencia, lo que refuerza la idea de que la mayor parte de la energía acústica del dron se encuentra dentro de este intervalo.

Con esta información, se puede concluir que el filtro paso-banda adecuado para la detección del dron debe abarcar aproximadamente el rango de 3000 a 6000 Hz. Este intervalo garantizará que la señal más característica del dron se conserve, mientras que se eliminarán componentes de baja frecuencia que pueden estar relacionadas con ruido de fondo y otras interferencias ambientales. Además, al comparar los distintos escenarios de distancia, se confirma que este patrón de frecuencias es estable, lo que indica que la detección será consistente independientemente de la posición del dron respecto al micrófono.

3.6 Cuarta fase: Técnica de detección para identificación de dron

Con la información extraída en la fase anterior, en la que se ha definido la huella acústica del dron y delimitado el rango de frecuencias más característico de su sonido, se procede ahora a la implementación del sistema de detección. Para ello, se ha desarrollado un nuevo *script*, “Deteccion_umbral.py” (Anexo VIII: Código de filtro paso-banda y definición de umbral), cuyo objetivo es aplicar un filtrado de la señal y establecer un umbral de detección basado en el análisis de la energía acústica en la banda de interés.

La finalidad de este procedimiento es clara: asegurar que el sistema pueda diferenciar de manera efectiva cuándo la señal recibida contiene la presencia del dron y cuándo no. Con este fin, se han

implementado dos pasos clave. Primero, la aplicación de un filtro paso-banda que permite aislar la franja de 3000-6000 Hz, donde se concentra la mayor parte de la energía relevante del dron. Segundo, la definición de un umbral de detección, estableciendo un criterio cuantitativo que permita decidir si la señal analizada corresponde efectivamente a la presencia del UAV.

3.6.1 Aplicación del filtro paso-banda

Para el filtrado de la señal, se ha optado por un filtro Butterworth de orden 5. Este tipo de filtro es conocido por su respuesta en frecuencia suave y sin ondulaciones en la banda pasante, lo que lo convierte en una elección adecuada cuando se busca preservar la integridad de la señal filtrada sin introducir distorsiones abruptas. En los filtros digitales, el orden hace referencia a la cantidad de coeficientes que definen su respuesta en frecuencia. Un filtro de orden superior presenta una pendiente más pronunciada en la región de transición, permitiendo una mayor atenuación fuera de la banda de paso, aunque también puede introducir más desfase y posibles distorsiones. Los filtros Butterworth pueden diseñarse con distintos órdenes, pero se ha seleccionado el orden 5 por ofrecer un equilibrio entre atenuación y distorsión. Un orden más bajo no filtraría eficazmente las frecuencias no deseadas, mientras que uno más alto podría generar efectos no deseados, como distorsión en la señal o inestabilidad numérica en la implementación del filtro.

El diseño del filtro se ha definido utilizando la siguiente función perteneciente a la librería *butter*:

```
def butter_bandpass(lowcut, highcut, fs, order=5):
    nyq = 0.5 * fs
    low = lowcut / nyq
    high = highcut / nyq
    b, a = butter(order, [low, high], btype='band')
    return b, a
```

Así, se definen los coeficientes del sistema en función del rango de frecuencias de interés. Para su correcta aplicación, se define la frecuencia de Nyquist como la mitad de la frecuencia de muestreo, lo que permite normalizar las frecuencias de corte antes de diseñar el filtro. Posteriormente, los coeficientes obtenidos, *b* y *a*, se utilizan al aplicar el filtro. Estos coeficientes representan los parámetros del filtro digital diseñado. *b* corresponde a los coeficientes del numerador de la función de transferencia del filtro, mientras que *a* representa los coeficientes del denominador. Juntos, definen la respuesta en frecuencia del sistema, determinando cómo se atenúan o preservan las distintas componentes de la señal.

Después de definirse el filtro, se aplica a la señal de audio recibida mediante la función:

```
def bandpass_filter(data, lowcut, highcut, fs, order=5):
    b, a = butter_bandpass(lowcut, highcut, fs, order=order)
    return filtfilt(b, a, data)
```

Cuando se aplica la función *filtfilt*, lo que ocurre es que la señal *data* se filtra dos veces: primero en el sentido directo y luego en el inverso. Esto asegura que no haya desfase en la señal resultante, ya que el filtrado causal tradicional suele generar retardos en la respuesta de la señal. Gracias a este procedimiento, el filtrado se realiza de manera eficiente sin comprometer la alineación temporal de la señal acústica, lo cual es esencial en aplicaciones de detección.

3.6.2 Generación de los espectrogramas filtrados

Tras la aplicación del filtro, la señal resultante es sometida al cálculo del espectrograma, empleando el mismo procedimiento utilizado en fases anteriores. La diferencia en este punto radica en que ahora la

señal analizada conserva fundamentalmente la información contenida en la banda de 3000-6000 Hz, eliminando componentes ajenos a esta región.

La representación en dB se obtiene mediante la transformación logarítmica estándar, permitiendo visualizar la intensidad de la señal en cada instante y facilitando el análisis cuantitativo posterior. En este punto, la información del espectrograma se almacena en una matriz, donde cada fila representa una frecuencia específica y cada columna un instante de tiempo. Esta matriz nos permite extraer con precisión la energía contenida en la banda de interés y analizar su comportamiento en función del tiempo. La conversión a dB es fundamental, ya que proporciona una escala más adecuada para interpretar la distribución de energía en la señal, mejorando la capacidad del sistema para diferenciar eventos acústicos relevantes de ruido residual. Podemos observar estos espectrogramas filtrados en la Figura 3-27

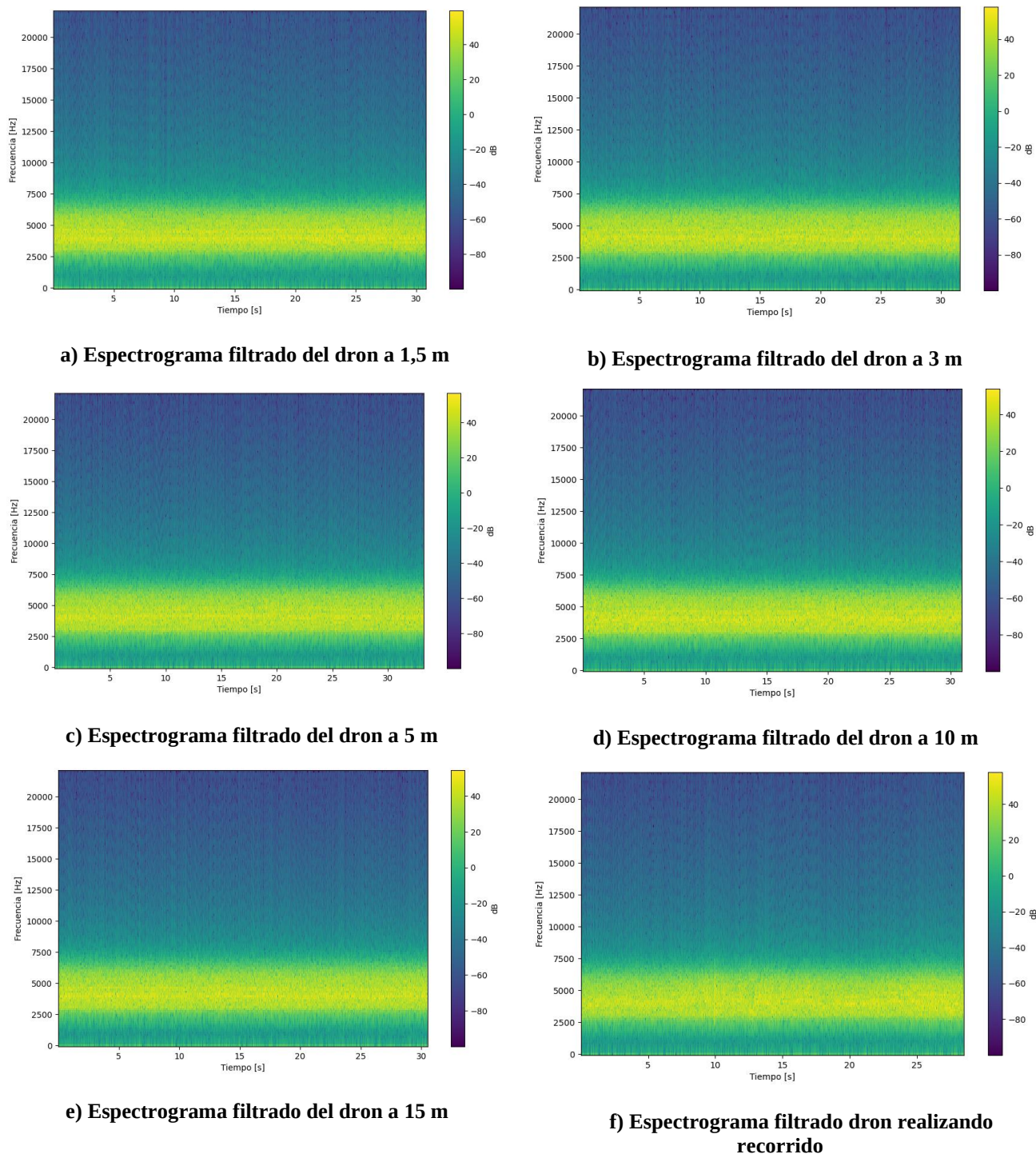


Figura 3-27 Espectrogramas filtrados de los audios del dron a distintas distancias

Al observar la Figura 3-27, se aprecia con claridad cómo las frecuencias ajenas a la banda de 3000-6000 Hz han sido muy atenuadas por el filtrado, lo que resalta aún más las regiones donde el dron concentra su energía acústica. En comparación con los espectrogramas originales, ahora la señal está mucho más definida en el rango de interés, evitando la interferencia de componentes de baja y alta frecuencia que podrían dificultar la detección. Además, se observa cómo la intensidad de la señal dentro de esta banda varía en función de la distancia, lo que resalta la importancia de establecer un umbral adecuado que permita una detección fiable del dron sin depender de la distancia específica.

3.6.3 Extracción de energía en las bandas de interés

El siguiente paso en el proceso consiste en aislar la información contenida en la banda seleccionada y calcular su energía promedio a lo largo del tiempo. Para ello, se trabaja con la matriz del espectrograma previamente generada, seleccionando las filas que corresponden a las frecuencias dentro del rango de 3000 a 6000 Hz. La extracción de energía se realiza promediando los valores de amplitud en dB dentro de estas filas a lo largo de todas las columnas temporales, de manera que se obtiene una estimación estable del contenido energético en la banda de interés. Esta operación nos proporciona un valor medio de energía en la banda de interés, representativo de la presencia del dron en la señal. El proceso garantiza que cualquier fluctuación en la señal se tenga en cuenta de manera global y no dependa de un instante aislado, asegurando así una medida más estable y menos susceptible a variaciones puntuales.

Esta extracción de energía se realiza con los siguientes comandos:

```
freq_low_idx = np.searchsorted(f, low_freq)
freq_high_idx = np.searchsorted(f, high_freq)
energy_band = Sxx_dB[freq_low_idx:freq_high_idx, :]
avg_energy = np.mean(energy_band)
energy_values.append(avg_energy)
```

Primero se identifican los índices de las frecuencias de corte en la matriz del espectrograma utilizando las funciones de la librería *numpy*, *np.searchsorted(f, low_freq)* y *np.searchsorted(f, high_freq)*. Estas funciones permiten localizar en la variable de frecuencias *f* las posiciones correspondientes a *low_freq* y *high_freq*, asegurando que seleccionamos el intervalo correcto dentro de la matriz espectral.

Una vez identificados los límites de la banda, se extraen los valores de energía de la matriz *Sxx_dB* utilizando *Sxx_dB[freq_low_idx:freq_high_idx, :]*, lo que nos proporciona únicamente la información contenida en la franja de interés a lo largo del tiempo. Posteriormente, se calcula el valor medio de la energía en dB dentro de esta banda mediante *np.mean(energy_band)*, asegurando que la estimación obtenida representa de manera global el comportamiento de la señal en este rango. Finalmente, este valor promedio se almacena en una lista (*energy_values.append(avg_energy)*) para su posterior análisis y comparación con otros datos de referencia. Este enfoque permite obtener una métrica más estable y representativa del contenido energético de la señal en la banda relevante. La elección del promedio en lugar de otros estadísticos responde a la necesidad de reducir el impacto de fluctuaciones puntuales en la señal, evitando que valores atípicos puedan distorsionar la detección.

3.6.4 Definición del umbral de detección

Una vez obtenidos los valores de energía en la banda filtrada, se procede a definir un umbral de detección basado en su distribución estadística. Para ello, se calculan la media ($\bar{E}$) y la desviación estándar (σ) de los valores obtenidos, utilizando las ecuaciones (4) y (5):

$$\bar{E} = \frac{1}{M} \sum_{i=1}^M E_i \quad (4)$$

$$\sigma = \sqrt{\frac{1}{M} \sum_{i=1}^M (E_i - \bar{E})^2} \quad (5)$$

Posteriormente, se establece un umbral utilizando la ecuación (6):

$$\text{Umbral de detección inferior} = \bar{E} - k \cdot \sigma \quad (6)$$

$$\text{Umbral de detección superior} = \bar{E} + k \cdot \sigma$$

El parámetro k actúa como un factor de tolerancia, permitiendo ajustar la sensibilidad del sistema. En este caso, se ha determinado empíricamente que un valor de 1,5 ofrece un equilibrio adecuado entre precisión y robustez, minimizando tanto los falsos positivos como los falsos negativos en la detección.

Para garantizar la reproducibilidad del análisis y evitar recalcularlo en cada ejecución, los valores obtenidos se guardan en un archivo. Este enfoque permite utilizar el mismo criterio de detección en futuras evaluaciones sin necesidad de repetir el proceso de análisis cada vez que se incorpora una nueva señal. Esta implementación representa el paso final en el desarrollo del sistema de detección, sentando las bases para la evaluación de su rendimiento en condiciones reales. En la Figura 3-28 se puede observar el resultado de nuestro umbral de detección basado en los audios grabados del dron a varias distancias.

```
C:\Users\asisa\Desktop\PROGRAMACION\PC>python3 Deteccion_umbral_DB.py

=====
UMBRAL DE DETECCIÓN (dB)
=====
Umbral inferior: 35.10 dB
Umbral superior: 40.38 dB
=====
✅ Umbral guardado en: Pruebas\umbral_deteccion.txt
```

Figura 3-28 Umbral de detección del dron

Como resultado del análisis, se ha definido un umbral inferior de 35,10 dB y un umbral superior de 40,38 dB. Estos valores representan los límites dentro de los cuales se considera que la señal analizada contiene la presencia del dron, recordando que estos límites están establecidos dentro de una banda de frecuencias de interés, por lo que, al analizar un audio, como veremos en el apartado 4, se buscará únicamente en esa banda de frecuencias y solo lo que esté dentro de ese umbral se considerará un dron. Este umbral se guarda en un archivo aparte llamado “umbral_deteccion.txt”, que posteriormente será cargado para compararlo con un audio recibido en tiempo real.

3.7 Quinta fase: Aplicación de la técnica de detección en tiempo real

Tras definir la técnica de detección en la fase anterior, en la que se estableció el filtro paso-banda y el umbral basado en la energía acústica en decibelios, se procede ahora a la implementación del sistema en tiempo real. En esta fase, el método desarrollado se aplica a los audios capturados de manera continua en la fase 2, evaluando en cada ventana temporal de 5 segundos si la señal contiene las características acústicas del dron. Para ello, cada fragmento de audio recibido es sometido al mismo procesamiento: se filtra en la banda de interés de 3000-6000 Hz y se calcula su nivel de energía. Este valor se compara con

el umbral predefinido, permitiendo determinar en tiempo real si el dron está presente en la escena o no. Esta fase representa la transición del análisis pasivo de datos a un sistema funcional de detección operando en condiciones reales, con el objetivo de validar la eficacia del modelo en entornos urbanos y de conflicto simulados.

Para ello, se ha desarrollado un nuevo *script* denominado “Detección_tiempo_real.py” (Anexo X: Código de detección en tiempo real), el cual integrará la captura, procesamiento y análisis del audio en tiempo real. Este *script* permitirá la detección automatizada del dron a medida que se reciben los datos, ejecutando el filtrado y la comparación con el umbral sin intervención manual.

A continuación, desglosaremos el *script* paso a paso.

3.7.1 Configuración del sistema y carga del umbral de detección

Antes de iniciar la detección, el programa establece la configuración del sistema y carga el umbral de detección desde un archivo externo.

```
log_folder = "Archivos diarios de detección"
log_filename=os.path.join(log_folder,f"deteccion_{datetime.now().strftime('%Y-%m-%d')}.log")
logging.FileHandler(log_filename, encoding="utf-8"),
logging.StreamHandler()
```

Este código inicializa el sistema de *logging*, que permite registrar eventos clave en un archivo de *log* y mostrarlos en la consola en tiempo real. Este archivo *log* se almacenará en una carpeta llamada “Archivos diarios de detección”.

Posteriormente, se carga el umbral de detección, guardado en el archivo *umbral_detección.txt*, el cual contiene los valores mínimo y máximo de energía en la banda de interés. Para hacerlo usamos:

```
umbral_file = "umbral_deteccion.txt"
with open(umbral_file, "r") as f:
    lines = f.readlines()
    umbral_inferior=float(lines[0].split(":")[1].strip().replace("dB", ""))
    umbral_superior=float(lines[1].split(":")[1].strip().replace("dB", ""))
```

Este fragmento de código extrae los valores numéricos del umbral desde el archivo de texto, eliminando los caracteres innecesarios como “dB”. Para comprender estas líneas de código, en la Figura 3-29 se ve ilustrado el formato que tiene el archivo TXT.

```
Umbral inferior: 35.10 dB
Umbral superior: 40.38 dB
```

Figura 3-29 Archivo TXT que contiene el umbral de detección

3.7.2 Definición y aplicación de filtro paso-banda

El sistema emplea un filtro paso-banda para aislar las frecuencias características de un dron, eliminando aquellas que no son relevantes para la detección.

Dado que esta técnica ya se explicó previamente, no se volverá a detallar aquí. En este punto del código, simplemente se aplica el filtro a la señal para extraer únicamente las frecuencias dentro del rango de interés (3000-6000 Hz), lo que permitirá posteriormente el análisis de energía en el espectrograma.

3.7.3 Procesamiento de archivos de audio

Cuando un nuevo archivo de audio llega a la carpeta de entrada, el sistema lo procesa para determinar si contiene señales de dron.

```
input_folder = "Audio Recibido/12-03-2025"
output_folder = "Resultados Detección"
```

Los archivos de audio están almacenados en la carpeta del día correspondiente dentro de "Audio Recibido". Los resultados del análisis se guardan en la carpeta "Resultados Detección", donde se genera un archivo de texto con la energía promedio detectada y la conclusión sobre la posible presencia de un dron en el audio analizado.

Para procesar el audio, se utiliza la siguiente secuencia:

```
def process_audio_file(file_path):
    with wave.open(file_path, "rb") as wav_file:
        fs = wav_file.getframerate()
        frames = wav_file.readframes(-1)
        audio_data = np.frombuffer(frames, dtype=np.int16)
```

Aquí se abre el archivo WAV, se extrae su frecuencia de muestreo (fs) y se convierte el contenido a un array *numpy* para su posterior análisis.

El siguiente paso es aplicar el filtro paso-banda con un comando que ya conocemos:

```
filtered_audio=bandpass_filter(audio_data, low_freq, high_freq, fs)
```

3.7.4 Generación de espectrogramas, extracción de energía y comparación con umbral

Dado que la generación del espectrograma, la extracción de energía y la comparación con el umbral ya se explicaron en detalle anteriormente, no se vuelve a explicar en detalle, sino que se aplica directamente en esta fase del análisis. El espectrograma se utiliza únicamente para analizar la distribución de la energía en la banda de interés (3000-6000 Hz), sin ser guardado ni visualizado. Posteriormente, se calcula la energía promedio en esta banda y se compara con los umbrales predefinidos para determinar si se trata de un posible dron.

Antes de esta comprobación, se asegura que el audio no se ha analizado previamente, utilizando la variable *result_path* y un condicional:

```
result_path=os.path.join(output_folder, os.path.basename(file_path)
+ ".txt")
    if os.path.exists(result_path):
        return
```

Esto asegura que si ya existe un archivo de resultados correspondiente a un audio recibido, ese audio no se vuelva a analizar, incluso si el script se reinicia.

Para esta comparación, se verifica si el valor de energía promedio obtenido se encuentra dentro del rango definido en el archivo de configuración. Si el valor está dentro del umbral establecido, se marca el archivo como una posible detección de dron y se registra en el sistema de *logging* para su análisis posterior.

```
if umbral_inferior <= avg_energy <= umbral_superior:
    logging.info(f"Posible DRON detectado en el audio
{os.path.basename(file_path)} ({avg_energy:.2f} dB)")
```

3.7.5 Almacenamiento de resultados y monitoreo en tiempo real

Una vez realizada la comparación con el umbral, el sistema almacena los resultados en la carpeta de salida. Se genera un archivo de texto por cada audio procesado, el cual contiene información relevante sobre la detección:

```
result_path=os.path.join(output_folder, os.path.basename(file_path)
+ ".txt")
with open(result_path, "w", encoding="utf-8") as res_file:
    res_file.write(f"Archivo: {os.path.basename(file_path)}\n")
    res_file.write(f"Energía promedio: {avg_energy:.2f} dB\n")
    res_file.write(f"Resultado: {'Dron detectado' if umbral_inferior
<= avg_energy <= umbral_superior else 'No es un dron'}\n")
```

Aquí, el programa crea un archivo de texto en la carpeta *output_folder* con el mismo nombre que el archivo de audio analizado, añadiendo la extensión TXT. En este archivo, se registran el nombre del audio, la energía promedio detectada y la conclusión del análisis (si se ha detectado un dron o no).

A continuación, el sistema entra en un bucle de monitoreo continuo, en el cual verifica cada 5 segundos si han llegado nuevos archivos de audio a la carpeta de entrada. Si no se encuentran archivos nuevos, se muestra una advertencia en el *log* para alertar de que no se están recibiendo datos.

```
while True:
    audio_files=[f for f in os.listdir(input_folder) if
f.endswith('.wav')]
    new_files = [f for f in audio_files if f not in processed_files]
    if not audio_files:
        logging.warning("No se encontraron archivos para analizar.
Verifica si el sistema de recepción está funcionando.")
```

En caso contrario, el audio se envía a *process_audio_file* y se realiza la secuencia explicada anteriormente.

```
else:
    for audio_file in audio_files:
        file_path = os.path.join(input_folder, audio_file)
        process_audio_file(file_path)
```

Este bucle se ejecuta de manera continua, en el cual se revisa la carpeta de entrada cada 5 segundos en busca de nuevos archivos de audio

```
time.sleep(5)
```

Este tiempo de espera permite que el sistema no consuma recursos constantemente verificando archivos en la carpeta de entrada. Un intervalo de 5 segundos es suficiente para mantener un equilibrio entre eficiencia y rendimiento.

Finalmente, si el usuario decide detener la ejecución manualmente con “Ctrl+C”, se captura la señal de interrupción y se muestra un mensaje indicando que la detección ha sido detenida:

```
except KeyboardInterrupt:  
    logging.info("Detección detenida por el usuario.")
```

Desde la configuración inicial hasta el monitoreo constante de la carpeta de entrada, el programa opera de forma autónoma, asegurando que cada nuevo archivo recibido sea procesado sin necesidad de intervención manual.

El proceso comienza con la carga de los umbrales de detección, los cuales establecen el rango energético en el que se considera que una señal podría corresponder al sonido característico de un dron. Luego, el audio es filtrado mediante un filtro pasa-banda, que elimina frecuencias irrelevantes y deja únicamente la banda de interés entre 3000 y 6000 Hz. Con esta señal filtrada, el sistema genera un espectrograma, cuyo único propósito es extraer la energía dentro del rango de análisis. Este valor se compara con los umbrales predefinidos y, en función del resultado, se clasifica el archivo como una posible detección de dron o no.

Una vez completado el análisis, los resultados se almacenan en un archivo de texto dentro de la carpeta de salida. Esto permite llevar un registro claro de cada audio procesado, con información detallada sobre la energía detectada y el resultado del análisis. Además, el sistema está diseñado para operar de manera continua, verificando cada cinco segundos si han llegado nuevos archivos a la carpeta de entrada. Si el usuario decide detener la ejecución, el programa maneja la interrupción de manera controlada y finaliza sin afectar los registros previos.

Este enfoque permite un equilibrio entre automatización y supervisión manual. Mientras que el operario solo debe asegurarse de que las carpetas de entrada y salida estén correctamente configuradas, el sistema se encarga del procesamiento sin requerir una supervisión constante. La combinación de procesamiento de señales, análisis espectral y monitoreo en tiempo real garantiza una detección de drones rápida, precisa y adaptable a distintos entornos operativos.

4 RESULTADOS, DISCUSIÓN Y PRUEBA DE VALIDACIÓN

Para comprobar la eficacia de la red de sensores en la detección de drones en interiores, se llevaron a cabo una serie de pruebas diseñadas para evaluar su rendimiento en distintas condiciones. El objetivo es no solo verificar que el sistema funciona correctamente en un entorno controlado, sino también someterlo a escenarios más exigentes, acercándolo lo máximo posible a situaciones reales.

El primer paso será analizar su desempeño sin transmisión en tiempo real, utilizando archivos de audio, aplicando el filtro y el umbral de detección previamente definido. Esto permitirá entender cómo responde el sistema en condiciones ideales antes de introducir factores que puedan comprometer su precisión.

A continuación, se incorporará ruido ambiental para simular el contexto de un entorno urbano o un escenario de conflicto, donde la presencia de otras fuentes sonoras podría interferir con la detección, permitiendo concluir si existen falsos positivos. También se probará un modelo distinto de micrófono para determinar cuál ofrece una mejor captura de la señal, teniendo en cuenta variables como la sensibilidad, la respuesta en frecuencia y la capacidad de discriminar sonidos relevantes en medio del ruido.

Finalmente, se realizará una prueba de validación en un entorno simulado que replicará la entrada de un dron en un edificio con varias salas. En esta última fase, el sistema operará en tiempo real, permitiendo evaluar su capacidad para detectar la presencia del dron con la inmediatez y precisión que exige una aplicación práctica.

4.1 Prueba 1: Prueba general de funcionamiento (sin transmisión en tiempo real)

4.1.1 Procedimiento

Para validar el sistema de detección de drones, la primera prueba se centra en verificar su correcto funcionamiento bajo condiciones controladas, utilizando los seis audios previamente registrados del dron a distintas distancias. En este punto, no se aplicará procesamiento en tiempo real; el análisis se realizará sobre archivos de audio almacenados, con el objetivo de comprobar que el sistema es capaz de detectar la presencia del dron de manera fiable antes de introducir nuevas variables como ruido ambiental o transmisión en red.

Para ello, se utilizará una única Raspberry Pi conectada a un micrófono *TONOR TC30*, la cual registrará el sonido ambiente y almacenará los archivos de audio en un formato adecuado para su análisis posterior. En cada ensayo, se reproducirá el sonido característico del dron y se aplicará el filtro paso-banda correspondiente antes de generar el espectrograma.

El procedimiento consiste en cargar cada uno de los seis audios en un *script* de detección desarrollado específicamente para esta prueba, denominado “Prueba_detección.py” (Anexo IX: Código

de pruebas del dron). Este *script* analiza el contenido del audio y determina si la señal registrada se encuentra dentro del rango de frecuencias e intensidades características del dron, basándose en los umbrales previamente definidos.

4.1.2 Flujo de ejecución

El proceso de análisis se divide en varias etapas:

4.1.2.1 Carga y filtrado de audio

Se toma el archivo de audio correspondiente y se aplica un filtro paso-banda en el rango de 3000-6000 Hz, eliminando las frecuencias que no pertenecen a la señal del dron. Para ello se utilizan los comandos previamente estudiados en el desarrollo

Primero definición del filtro Butterworth:

```
def butter_bandpass(lowcut, highcut, fs, order=5):
```

Posteriormente, aplicación del filtro y uso de la librería *filtfilt*:

```
def bandpass_filter(data, lowcut, highcut, fs, order=5):
    b, a = butter_bandpass(lowcut, highcut, fs, order=order)
    return filtfilt(b, a, data)
```

Y finalmente, aplicación del filtro al audio seleccionado:

```
filtered_audio=bandpass_filter(audio_data, low_freq, high_freq, fs)
```

4.1.2.2 Generación de espectrograma

Tras el filtrado, se genera un espectrograma para visualizar la distribución de energía en las distintas frecuencias a lo largo del tiempo. La energía espectral se convierte a decibelios (dB) para poder evaluar la intensidad de la señal dentro del rango de interés.

Esto se hace con los comandos ya estudiados y que se muestran a continuación:

Primero, la generación del espectrograma:

```
f, t, Sxx = spectrogram(filtered_audio, fs)
```

Después, la conversión a decibelios:

```
Sxx_dB = 10 * np.log10(Sxx + 1e-10)
```

4.1.2.3 Comparación con el umbral de detección

Se calcula el promedio de la intensidad (en dB) dentro de la banda de 3000-6000 Hz y se compara con el umbral de detección definido previamente en un archivo de configuración (*umbral_deteccion.txt*).

Para la extracción de energía se utilizan los mismos comandos que cuando definimos el umbral:

```
freq_low_idx = np.searchsorted(f, low_freq)
freq_high_idx = np.searchsorted(f, high_freq)
dB_mean = np.mean(Sxx_dB[freq_low_idx:freq_high_idx, :])
```

Y posteriormente lo comparamos con ese umbral definido previamente, el cual hemos cargado con anterioridad en el *script*.

```
deteccion = umbral_low <= dB_mean <= umbral_high
```

Si la intensidad del audio filtrado se encuentra dentro del rango establecido por los umbrales, el sistema detecta la presencia del dron y lo señala como una posible intrusión. Si el nivel de dB está por debajo del umbral inferior o por encima del superior, se considera que no se ha detectado un dron en el audio.

4.1.3 Resultados y discusión

Los resultados obtenidos se pueden observar en la Figura 4-1 y sus distintas subfiguras:

```
=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 40.19 dB
Umbral inferior: 35.10 dB
Umbral superior: 40.38 dB
=====
🚨🚨🚨 Posible DRON detectado en el audio.
=====
Espectrograma guardado en: Detecciones Pruebas\Dron 1,5 m_detected.png
```

a) Dron a 1,5 metros

```
=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 38.24 dB
Umbral inferior: 35.10 dB
Umbral superior: 40.38 dB
=====
🚨🚨🚨 Posible DRON detectado en el audio.
=====
Espectrograma guardado en: Detecciones Pruebas\Dron 3 m_detected.png
```

b) Dron a 3 metros

```
=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 37.11 dB
Umbral inferior: 35.10 dB
Umbral superior: 40.38 dB
=====
🚨🚨🚨 Posible DRON detectado en el audio.
=====
Espectrograma guardado en: Detecciones Pruebas\Dron 5 m_detected.png
```

c) Dron a 5 metros

```
=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 35.57 dB
Umbral inferior: 35.10 dB
Umbral superior: 40.38 dB
=====
🚨🚨🚨 Posible DRON detectado en el audio.
=====
Espectrograma guardado en: Detecciones Pruebas\Dron 10 m_detected.png
```

d) Dron a 10 metros

```

=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 35.10 dB
Umbral inferior: 35.10 dB
Umbral superior: 40.38 dB
=====
XXX No se ha detectado un dron en el audio.
=====
Espectrograma guardado en: Detecciones Pruebas\Dron 15 m_detected.png

```

e) Dron a 15 metros

```

=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 38.03 dB
Umbral inferior: 35.10 dB
Umbral superior: 40.38 dB
=====
🚀🚀🚀 Posible DRON detectado en el audio.
=====
Espectrograma guardado en: Detecciones Pruebas\Dron recorrido entero_detected.png

```

f) Dron realizando recorrido por la sala

Figura 4-1 Resultados de detección

Los resultados obtenidos confirman el correcto funcionamiento del sistema de detección en las condiciones de prueba establecidas, permitiendo identificar la presencia del dron hasta una distancia de 10 metros. Sin embargo, al llegar a 15 metros, el sistema ya no logra detectarlo, concretamente, se sitúa justo en el valor del umbral inferior, lo que indica que esta es la distancia máxima de detección bajo los umbrales actuales. Cabe destacar que no se presentan los espectrogramas filtrados obtenidos porque son los mismos que los analizados en la sección 3.6.2.

Como era de esperar, a medida que el dron se aleja, la intensidad de la señal dentro de la banda 3000-6000 Hz disminuye progresivamente. Esto es un comportamiento lógico, ya que la energía sonora se atenúa con la distancia debido a la dispersión y la absorción del sonido en el ambiente.

Entrando en detalle con cada audio:

- Distancia corta (1,5 metros): En esta posición cercana, la intensidad medida es de 40,19 dB, un valor elevado dentro del umbral permitido. La detección es clara y sin margen de error.
- Distancia intermedia (3 y 5 metros): Se observa una reducción progresiva de la intensidad, con valores de 38,24 dB y 37,11 dB, respectivamente. Ambos siguen dentro del umbral de detección, confirmando que el sistema mantiene su efectividad en esta franja.
- Distancia límite (10 metros): El valor cae a 35,57 dB, aún dentro del umbral, pero muy cercano al límite inferior. Esto sugiere que a partir de esta distancia el margen de detección se vuelve más reducido.
- Distancia máxima (15 metros): La energía del dron en la banda de interés cae exactamente a 35,10 dB, coincidiendo con el umbral inferior. Como resultado, el sistema ya no es capaz de identificar su presencia, marcando esta distancia como el límite máximo de detección bajo estas condiciones.
- Recorrido completo del dron: En la prueba más realista, donde el dron se desplaza a lo largo del espacio de prueba, el sistema lo detecta correctamente con una intensidad media de 38,03 dB, lo que refuerza la estabilidad del método dentro de su rango operativo.

Debe resaltarse un aspecto: Si bien esta prueba se centró en la detección a diferentes distancias, es importante considerar otro escenario: ¿qué pasaría si el dron estuviera justo encima del micrófono, a 0 metros de distancia?

Dado que la intensidad de la señal disminuye con la distancia, el razonamiento inverso sugiere que, a una distancia extremadamente corta, la energía sonora en la banda de interés podría superar el umbral

superior establecido. En este caso, el sistema podría clasificar el sonido como fuera de rango y, paradójicamente, no detectar el dron.

Sin embargo, este escenario no representa un problema real en la práctica. Antes de estar a 0 metros, el dron necesariamente habría pasado por 1, 2 o más metros, donde sí habría sido detectado. Es decir, la detección ya se habría producido antes de alcanzar una proximidad extrema.

4.2 Prueba 2: Análisis de falsos positivos

4.2.1 Propósito

Para garantizar que el sistema de detección de drones no solo sea preciso en la identificación de su objetivo, sino también robusto frente a interferencias o ruido de fondo, se ha diseñado una prueba específica para evaluar la posible aparición de falsos positivos. Un falso positivo ocurriría si el sistema identificara erróneamente como dron un sonido que en realidad no lo es.

Para esta prueba, se han grabado distintos audios desde un altavoz a 1,5 metros y 3 metros de distancia del micrófono. Cada uno de estos sonidos representa un tipo de ruido que podría estar presente en un escenario real, especialmente en contextos urbanos o de guerra. Los audios seleccionados son los siguientes y se pueden escuchar accediendo al QR contenido en el Anexo XIII: QR para audios sonido externo.

- Sonido de ambiente de guerra
- Disparos (ráfagas de armas de fuego)
- Ataque aéreo con bombas (explosiones y sirenas)
- Sonido de lluvia (ruido constante y extendido en el tiempo)

Cada uno de estos audios se analizará utilizando el mismo procedimiento y *script* aplicado al dron. Esta evaluación permitirá verificar si el sistema es capaz de distinguir correctamente un dron de otros sonidos del entorno.

4.2.2 Resultados y discusión

En las siguientes figuras se ilustrarán y se analizarán los resultados obtenidos:

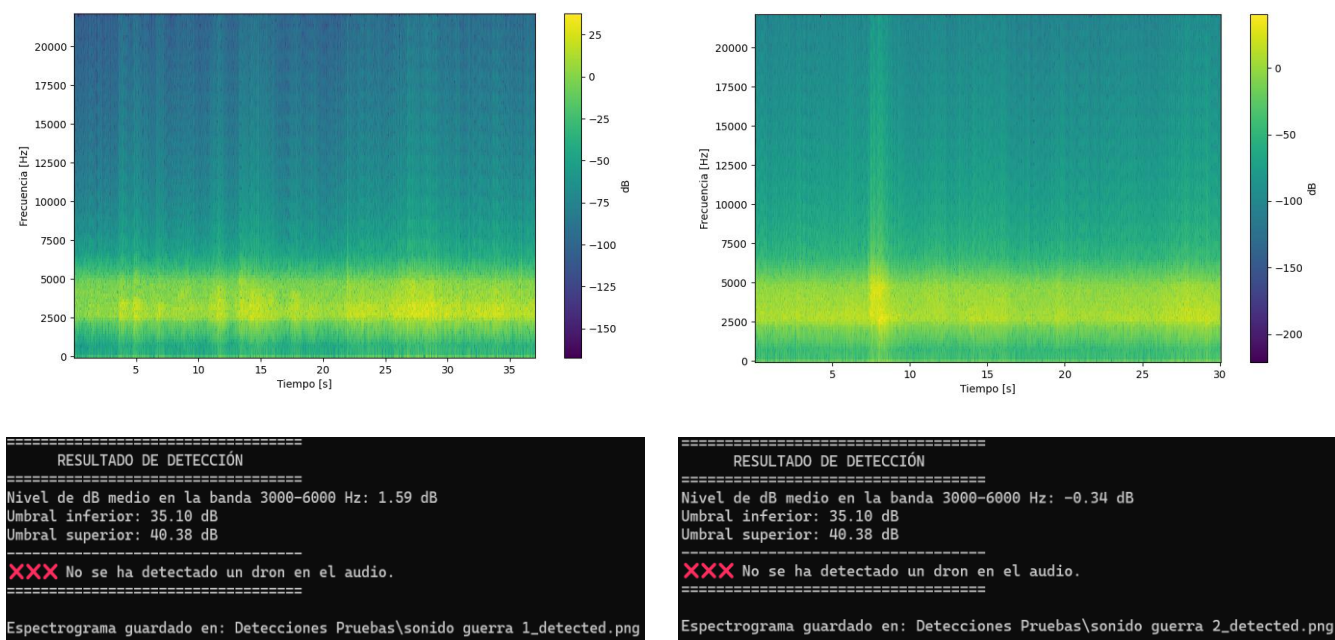


Figura 4-2 Análisis de sonido de guerra a varias distancias

En la Figura 4-2 se analiza el sonido de ambiente de guerra a dos distancias cercanas, confirmando que el sistema de detección ha respondido de manera correcta, descartándolo sin generar falsos positivos. A 1,5 metros, la intensidad medida en la banda de 3000-6000 Hz es de apenas 1,59 dB, y a 3 metros disminuye aún más hasta -0,34 dB, valores muy por debajo del umbral mínimo de detección de 35,10 dB. Esta diferencia es evidente también en los espectrogramas, donde la presencia de energía en la banda de interés es prácticamente nula. Aunque el sonido de ambiente de guerra contiene múltiples fuentes acústicas, su distribución de energía se concentra mayormente en frecuencias más bajas, lo que evita cualquier confusión con el patrón característico del dron.

Procedemos ahora con el audio de disparos, cuyos resultados se ven ilustrados en la Figura 4-3:

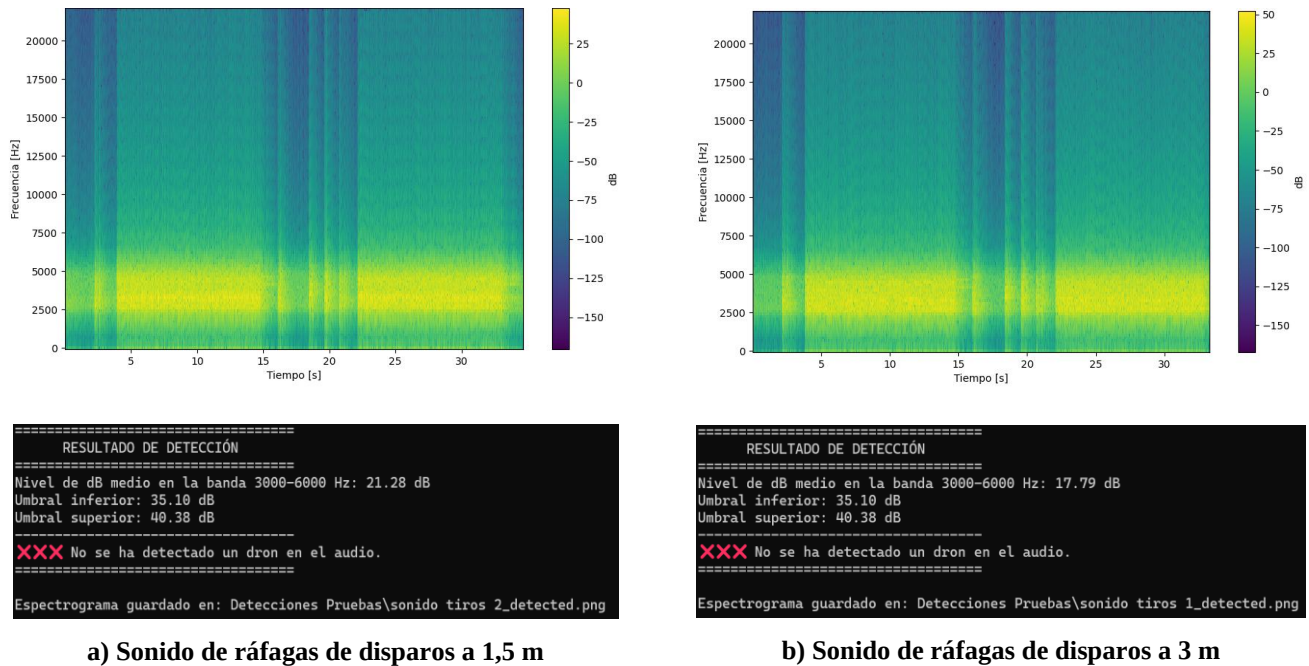
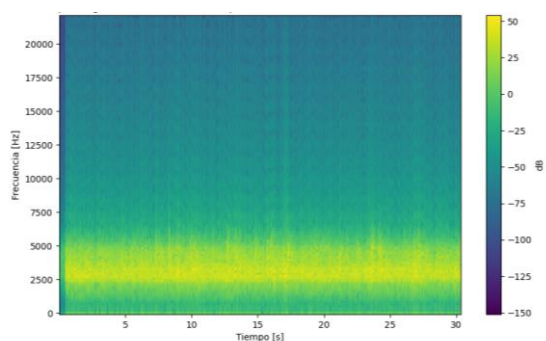


Figura 4-3 Análisis de sonido de disparos a varias distancias

El análisis del sonido de ráfagas de disparos muestra que, al igual que en la prueba anterior, el sistema ha sido capaz de diferenciarlo del ruido característico de un dron, evitando falsos positivos. A 1,5 metros, la intensidad media en la banda de 3000-6000 Hz es de 21,28 dB, mientras que a 3 metros el valor registrado desciende a 17,79 dB, un comportamiento esperado ya que la intensidad debería disminuir con la distancia.

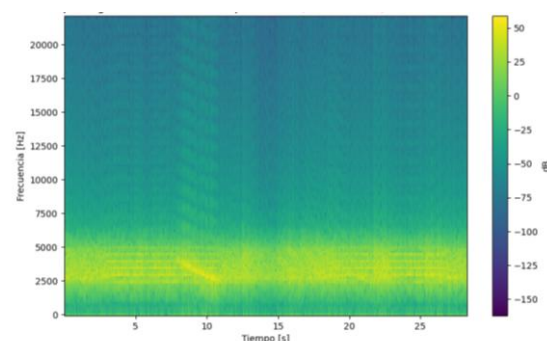
Lo fundamental es que en ambas mediciones los niveles de intensidad permanecen por debajo del umbral mínimo de detección de 35,10 dB, por lo que el sistema ha descartado correctamente estos sonidos sin confundirlos con un dron. En los espectrogramas se observa que, aunque los disparos generan energía en un amplio rango de frecuencias, su distribución en la banda específica de 3000-6000 Hz no es suficiente para activar la detección.

A continuación, se muestran los resultados del audio simulando un ataque aéreo, con sirenas y bombas. Se puede observar en la Figura 4-4:



```
=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 20.73 dB
Umbral inferior: 35.10 dB
Umbral superior: 40.38 dB
=====
XXX No se ha detectado un dron en el audio.
=====
Espectrograma guardado en: Detecciones Pruebas\sonido bombas 1_detected.png
```

a) Sonido bombardeo aéreo a 1,5 m



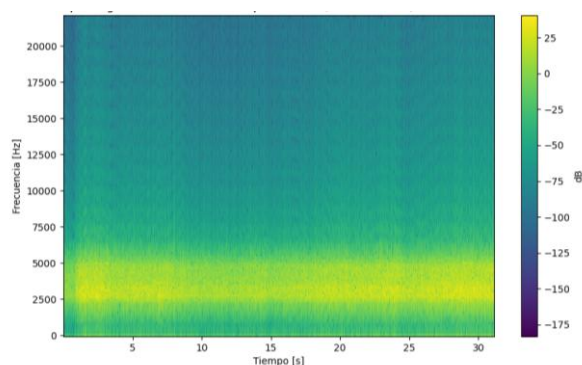
```
=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 19.81 dB
Umbral inferior: 35.10 dB
Umbral superior: 40.38 dB
=====
XXX No se ha detectado un dron en el audio.
=====
Espectrograma guardado en: Detecciones Pruebas\sonido bombas 2_detected.png
```

b) Sonido de bombardeo aéreo a 3 m

Figura 4-4 Análisis de sonido de bombardeo aéreo a varias distancias

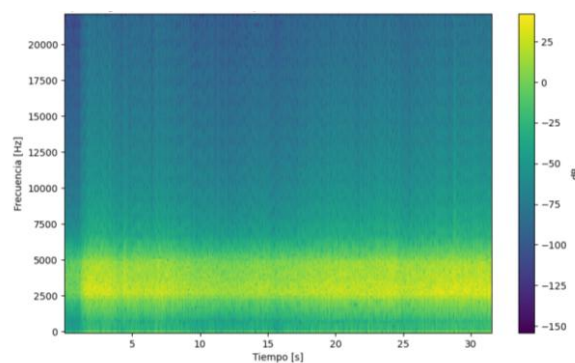
El análisis del sonido de bombardeo aéreo confirma nuevamente la solidez del sistema en la discriminación de sonidos, ya que en ningún caso se ha generado un falso positivo. A 1,5 metros, la intensidad medida en la banda de 3000-6000 Hz es de 20,73 dB, mientras que a 3 metros disminuye ligeramente hasta 19,81 dB, siguiendo una tendencia coherente con la atenuación esperada por la distancia. Aunque este tipo de sonido contiene explosiones y sirenas, su distribución de energía en el espectro no es suficiente para activar la detección, lo que sugiere que la señal del dron tiene una estructura más definida en la banda de interés. Los espectrogramas refuerzan esta idea, mostrando que la mayor parte de la energía se concentra en frecuencias más bajas, con una presencia mínima en el rango de detección del dron. La ligera variación en los valores de intensidad entre 1,5 y 3 metros podría deberse a la naturaleza variable del bombardeo, donde algunos picos de energía pueden diferir entre mediciones. Sin embargo, lo importante es que el sistema ha descartado correctamente estos sonidos, manteniendo su capacidad de distinguir entre ruidos del entorno y la señal característica del dron.

Finalmente, analizamos ahora un audio de lluvia constante. Se pueden ver los resultados en la Figura 4-5:



```
=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 10.19 dB
Umbral inferior: 35.10 dB
Umbral superior: 40.38 dB
=====
XXX No se ha detectado un dron en el audio.
=====
Espectrograma guardado en: Detecciones Pruebas\sonido lluvia 2_detected.png
```

a) Sonido lluvia constante a 1,5 m



```
=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 7.38 dB
Umbral inferior: 35.10 dB
Umbral superior: 40.38 dB
=====
XXX No se ha detectado un dron en el audio.
=====
Espectrograma guardado en: Detecciones Pruebas\sonido lluvia 1_detected.png
```

b) Sonido de lluvia constante a 3 m

Figura 4-5 Análisis de sonido de lluvia a varias distancias

Este análisis del sonido de lluvia refuerza aún más la precisión del sistema, ya que tampoco ha generado falsos positivos en ninguna de las mediciones. A 1,5 metros, la intensidad media en la banda de 3000-6000 Hz es de 10,19 dB, mientras que a 3 metros aumenta ligeramente a 7,38 dB. Ambos valores se encuentran muy por debajo del umbral mínimo de detección, lo que impide que el sistema lo identifique erróneamente como un dron.

Por su parte, los espectrogramas muestran un patrón de ruido constante, con mayor presencia de energía en frecuencias bajas y una disminución progresiva hacia las frecuencias más altas. En la banda de detección del dron, la energía es escasa, lo que evita que el sistema active una alerta. Estos resultados confirman que el método empleado es capaz de diferenciar sonidos ambientales prolongados y difusos de las señales características de un dron. La estabilidad de la detección ante ruido de fondo continuo como la lluvia es especialmente relevante, ya que demuestra que el sistema puede operar en entornos al aire libre sin verse afectado por condiciones meteorológicas adversas.

Esta tendencia se mantiene en el resto de sonidos evaluados, consolidando la capacidad del sistema para evitar falsos positivos incluso en escenarios ruidosos. En todos los casos analizados (sonidos de ambiente de guerra, disparos, bombardeos y lluvia) los niveles de intensidad en la banda de 3000-6000 Hz se mantuvieron significativamente por debajo del umbral mínimo de detección, lo que impidió que fueran identificados erróneamente como un dron. Aunque algunos sonidos impulsivos, como los disparos y las explosiones, mostraron una ligera variación en la intensidad con la distancia, su distribución espectral no presentó una concentración de energía suficiente en la banda de interés, permitiendo que el sistema los descartara de manera efectiva. En el caso del sonido de lluvia, la prueba resultó especialmente relevante, ya que confirma que el sistema puede operar de manera estable en entornos con ruido ambiental constante sin comprometer su rendimiento. Estos resultados reafirman que la selección del filtro paso-banda y los umbrales de detección han sido adecuados para discriminar con éxito la presencia de un dron, incluso en escenarios sonoros complejos, lo que representa un avance crucial en la robustez del sistema y su aplicabilidad en condiciones reales.

4.3 Prueba 3: Análisis de consumo energético

Hasta este punto, las Raspberry Pi 3 Modelo B han contado con una alimentación eléctrica estable mediante conexión a la red. Sin embargo, en un entorno de conflicto o en escenarios donde el acceso a la infraestructura eléctrica no esté garantizado, es esencial evaluar la autonomía del sistema cuando se alimenta mediante baterías portátiles.

Para ello, se ha llevado a cabo un análisis del consumo energético del microcontrolador, ya que es el elemento que está desplegado por el espacio a monitorizar, mientras que los otros componentes (ordenador y router) estarían en una ubicación con mejores condiciones. Además, añadir esta batería mejoraría la portabilidad del sistema, cambiándolo de ubicación según las necesidades de investigación. Su consumo depende de múltiples factores, incluyendo el uso de la red WiFi para la transmisión de audio, el procesamiento en tiempo real de las señales y los periféricos conectados (como el micrófono USB).

4.3.1 Batería portátil

Para determinar la autonomía del sistema, se ha utilizado una batería externa nueva denominada *VEEKTOMX Mini Power Bank 10000mAh*, cuyas características se detallan a continuación en la Tabla 4-1:

Especificación	Valor
Capacidad	10000 mAh
Potencia de salida máxima	22,5 W
Puertos de salida	2 USB-A, 1 USB-C
Dimensiones	10,6 × 6,86 × 1,6 cm ³
Peso	169 gramos
Funciones adicionales	Pantalla LED que muestra nivel de batería

Tabla 4-1 Características batería portátil VEEKTOMX Mini Power Bank (tomada de [59])

Esta batería se conecta a la Raspberry Pi tal y como se muestra en la Figura 4-6, y se mide el tiempo que permanece operativa mientras transmite audio de manera continua al sistema de procesamiento.



Figura 4-6 Batería portátil conectada a Raspberry Pi

4.3.2 Modificación de script “Send_audio.py”

Este *script*, encargado de la transmisión de audio desde la Raspberry Pi al sistema de procesamiento, se ha modificado para poder registrar la hora de inicio y la hora de finalización de la transmisión en un archivo de registro (“log.txt”). De esta manera, cuando la batería se agote y la Raspberry Pi se apague, será posible calcular el tiempo total de funcionamiento del sistema en condiciones realistas.

Para ello, se han añadido dos secuencias, además de importar la librería *datetime*, que permite registrar la fecha y hora exacta.

```
import datetime
```

Al iniciar la transmisión, se almacena la hora exacta en “log.txt”. Para ello, se ha añadido la siguiente línea de código al inicio del *script*:

```
start_time = datetime.datetime.now()
with open("log.txt", "a") as log_file:
    log_file.write(f"Inicio de transmisión: {start_time}\n")
```

Cuando la transmisión se detiene (ya sea porque se interrumpe manualmente o porque la batería se agota), se almacena la hora de finalización y se calcula la duración total de funcionamiento.

```
end_time = datetime.datetime.now()
with open("log.txt", "a") as log_file:
```

```
log_file.write(f"Fin de transmisión: {end_time}\n")
duration = end_time - start_time
log_file.write(f"Duración total: {duration}\n\n")
```

El resultado final es que el sistema registra el tiempo de operación sin interferir con la funcionalidad principal de envío de datos.

4.3.3 Resultado

Con la Raspberry Pi conectada a la batería portátil, se ejecuta el *script* de transmisión de audio, iniciando el envío de datos hacia el sistema de procesamiento. Se han realizado dos mediciones, obteniéndose los siguientes resultados en términos de autonomía energética:

Inicio de transmisión: 2025-03-14 10:14:47.247917	Inicio de transmisión: 2025-03-20 22:21:20.33256
Fin de transmisión: 2025-03-15 07:35:52.356712	Fin de transmisión: 2025-03-21 19:39:34.871237
Duración total: 21:20:5.108795	Duración total: 22:18:14.538677

a) Duración total primera prueba

b) Duración total segunda prueba

Figura 4-7 Duración batería portátil

Los resultados obtenidos indican que la Raspberry Pi, conectada a la batería portátil seleccionada, logra mantener la transmisión de datos de manera continua en la medida menos favorable durante aproximadamente 21 horas y 20 minutos antes de agotar su carga. Esto sugiere que, en un entorno de operación real, un operario debería reemplazar o recargar la batería cada 21 horas para garantizar la continuidad del microcontrolador. Este tiempo de autonomía es significativo y sugiere que el sistema puede mantenerse operativo durante largos periodos sin intervención constante, lo que mejora su viabilidad en escenarios tácticos donde el acceso a alimentación convencional es limitado, además de poder moverse de posición sin dificultad.

4.4 Prueba 4: Cambio de sensor acústico

La detección del dron hasta ahora se ha realizado con un único micrófono, el *TONOR TC30*, pero la elección del dispositivo puede influir en el rendimiento del sistema. Cada micrófono tiene una respuesta en frecuencia diferente, lo que puede alterar la forma en que la señal del dron es captada y, por tanto, su detección con el umbral previamente definido. Para evaluar esta variación, se ha incorporado un segundo micrófono, repitiendo las pruebas realizadas anteriormente y comparando los resultados.

El primer paso es comprobar si el umbral de detección original sigue funcionando con el nuevo micrófono. Si la detección falla, se analizará el espectro de frecuencias captado por el nuevo micrófono para ajustar el filtro paso-banda y calcular un nuevo umbral de detección.

Esta prueba no solo busca validar la adaptabilidad del sistema a distintos dispositivos, sino también preparar su integración en la siguiente y última prueba. Ambos micrófonos, cada uno con su propio umbral ajustado, se utilizarán en la prueba final de validación, donde estarán distribuidos en distintas salas para evaluar su rendimiento en un entorno más realista.

4.4.1 Modelo utilizado

Para evaluar la adaptabilidad de nuestro sistema de detección de drones a diferentes dispositivos de captura de audio, hemos incorporado la *Logitech C615*, una cámara web que incluye un micrófono integrado. En la Tabla 4-2 se pueden observar sus principales características acústicas:

Especificación	Detalle
Tipo de micrófono	Mono integrado
Características	Cancelación de ruido
Longitud del cable	95 cm

Tabla 4-2 Especificaciones Logitech C615 (tomada de [60])

Al tratarse de una cámara web, el micrófono integrado podría no ofrecer la misma calidad que dispositivos de grabación de audio dedicados. Esta diferencia podría afectar la capacidad del sistema para detectar la señal del dron. En caso de que la detección no sea efectiva con el umbral y el filtro actuales, ajustaremos estos parámetros en función de las características específicas del micrófono de la Logitech C615.

Además, la integración de una cámara web en nuestro sistema abre la posibilidad de futuras mejoras que combinen las señales de audio y video a la hora de hacer la detección, lo que podría enriquecer las capacidades del sistema en escenarios más complejos. En la Figura 4-8 se ve ilustrada el modelo utilizado.



Figura 4-8 Cámara de web Logitech C615 [60]

4.4.2 Grabación de audios y comprobación con umbral original

Para comprobar si el nuevo micrófono es capaz de detectar el dron con el umbral ya establecido, se han realizado grabaciones siguiendo exactamente las mismas condiciones que en el apartado de “Análisis de datos”. Se han registrado los audios a diferentes distancias, manteniendo el entorno y los parámetros de captación inalterados para que la única variable sea el cambio de micrófono. Estos audios se pueden escuchar accediendo al QR contenido en el Anexo XIV: QR para audios del UAV grabados desde Logitech C615.

Una vez obtenidos los audios (ver Figura 4-9), se ha aplicado el umbral de detección original, obteniendo los siguientes resultados. Esta comprobación se ha desarrollado nuevamente mediante el *script* “Prueba_detección.py”, el cual ha sido explicado anteriormente en el apartado 4.1.2.

-  Dron 1,5 m micro 2
-  Dron 3 m micro 2
-  Dron 5 m micro 2
-  Dron 10 m micro 2
-  Dron 15 m micro 2
-  Recorrido entero micro 2

Figura 4-9 Audios almacenados en ordenador y grabados por el Logitech C615

Los resultados se pueden ver ilustrados en las distintas subfiguras de la Figura 4-10:

```
=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 25.91 dB
Umbral inferior: 35.10 dB
Umbral superior: 40.38 dB
-----
XXX No se ha detectado un dron en el audio.
=====
```

a) Dron a 1,5 metros

```
=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 24.49 dB
Umbral inferior: 35.10 dB
Umbral superior: 40.38 dB
-----
XXX No se ha detectado un dron en el audio.
=====
```

b) Dron a 3 metros

```
=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 22.65 dB
Umbral inferior: 35.10 dB
Umbral superior: 40.38 dB
-----
XXX No se ha detectado un dron en el audio.
=====
```

c) Dron a 5 metros

```
=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 19.64 dB
Umbral inferior: 35.10 dB
Umbral superior: 40.38 dB
-----
XXX No se ha detectado un dron en el audio.
=====
```

d) Dron a 10 metros

```
=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 18.33 dB
Umbral inferior: 35.10 dB
Umbral superior: 40.38 dB
-----
XXX No se ha detectado un dron en el audio.
=====
```

e) Dron a 15 metros

```
=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 18.43 dB
Umbral inferior: 35.10 dB
Umbral superior: 40.38 dB
-----
XXX No se ha detectado un dron en el audio.
=====
```

f) Dron realizando recorrido por la sala

Figura 4-10 Resultados de detección con nuevo micrófono

Los resultados obtenidos confirman lo esperado: el nuevo micrófono no detecta el dron con el umbral original. En todas las distancias analizadas, el nivel medio de dB en la banda de 3000-6000 Hz es muy inferior al umbral de detección previamente establecido, lo que indica que este dispositivo capta la señal del dron con una menor intensidad en comparación con el micrófono utilizado en la Prueba 1.

En la prueba inicial, los valores de dB se encontraban dentro del rango de detección en la mayoría de las distancias, lo que permitía identificar la señal del dron con claridad. Sin embargo, con este nuevo

micrófono, incluso en la distancia más corta de 1,5 metros, la energía medida en la banda de interés apenas alcanza 25,91 dB, quedando lejos del umbral inferior de 35,10 dB. A medida que el dron se aleja, esta diferencia se acentúa aún más, llegando a valores de 18,33 dB a 15 metros, lo que confirma que el micrófono integrado en la *webcam* tiene una menor sensibilidad en las frecuencias relevantes para la detección del dron.

Este comportamiento era previsible, ya que se trata de un micrófono integrado en una webcam, diseñado para captación de voz y con un sistema de cancelación de ruido que podría estar afectando la detección de sonidos constantes como los generados por el dron. Aun así, estos resultados no implican que el micrófono sea inutilizable en el sistema, sino que será necesario redefinir el umbral y posiblemente ajustar el filtro paso-banda para adaptar la detección a sus características específicas. Este proceso se abordará en el siguiente apartado.

4.4.3 Ajuste de umbral y filtro paso-banda

Dado que el nuevo micrófono no ha sido capaz de detectar el dron con los parámetros originales, es necesario redefinir el umbral de detección y, si es preciso, ajustar el filtro paso-banda en función de su respuesta en frecuencia. Para ello, se ha realizado un análisis espectral de los audios captados con este micrófono, con el objetivo de identificar las frecuencias predominantes del dron y compararlas con las obtenidas en el desarrollo.

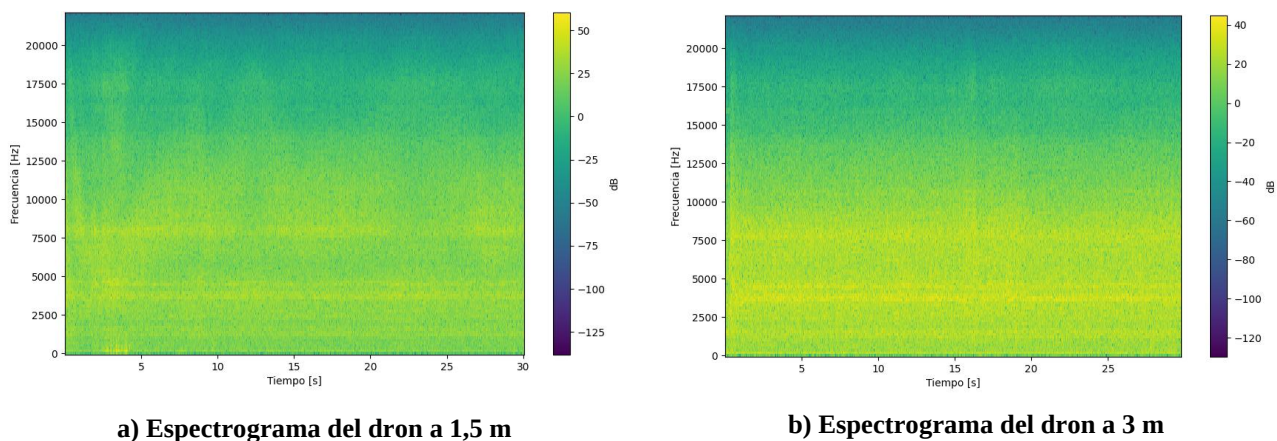
El procedimiento seguido es el mismo que en la calibración inicial: primero, se ha generado el espectrograma de los audios capturados con este micrófono, observando en qué regiones se concentra la mayor parte de la energía. A partir de esta información, se ha definido una nueva banda de frecuencias para aplicar el filtro paso-banda y, posteriormente, se ha calculado la media de los valores de dB en la nueva banda de interés. Con estos valores, se ha establecido un nuevo umbral adaptado a las características del micrófono.

Este ajuste permitirá que el sistema pueda operar correctamente con este dispositivo sin afectar la detección en el micrófono original.

4.4.4 Generación de espectrogramas e identificación de frecuencias dominantes

Se vuelve a ejecutar el *script* “Espectro.py” para realizar un primer análisis de las frecuencias predominantes en los audios capturados con el nuevo micrófono. Posteriormente, se emplean los *scripts* “Análisis_frecuencia.py” y “Análisis_boxplot.py” para identificar con mayor precisión la banda de frecuencias en la que se concentra la mayor parte de la energía, permitiendo así definir el rango óptimo para la aplicación del filtro paso-banda.

En la Figura 4-11 se pueden ver ilustrados los espectrogramas para cada distancia:



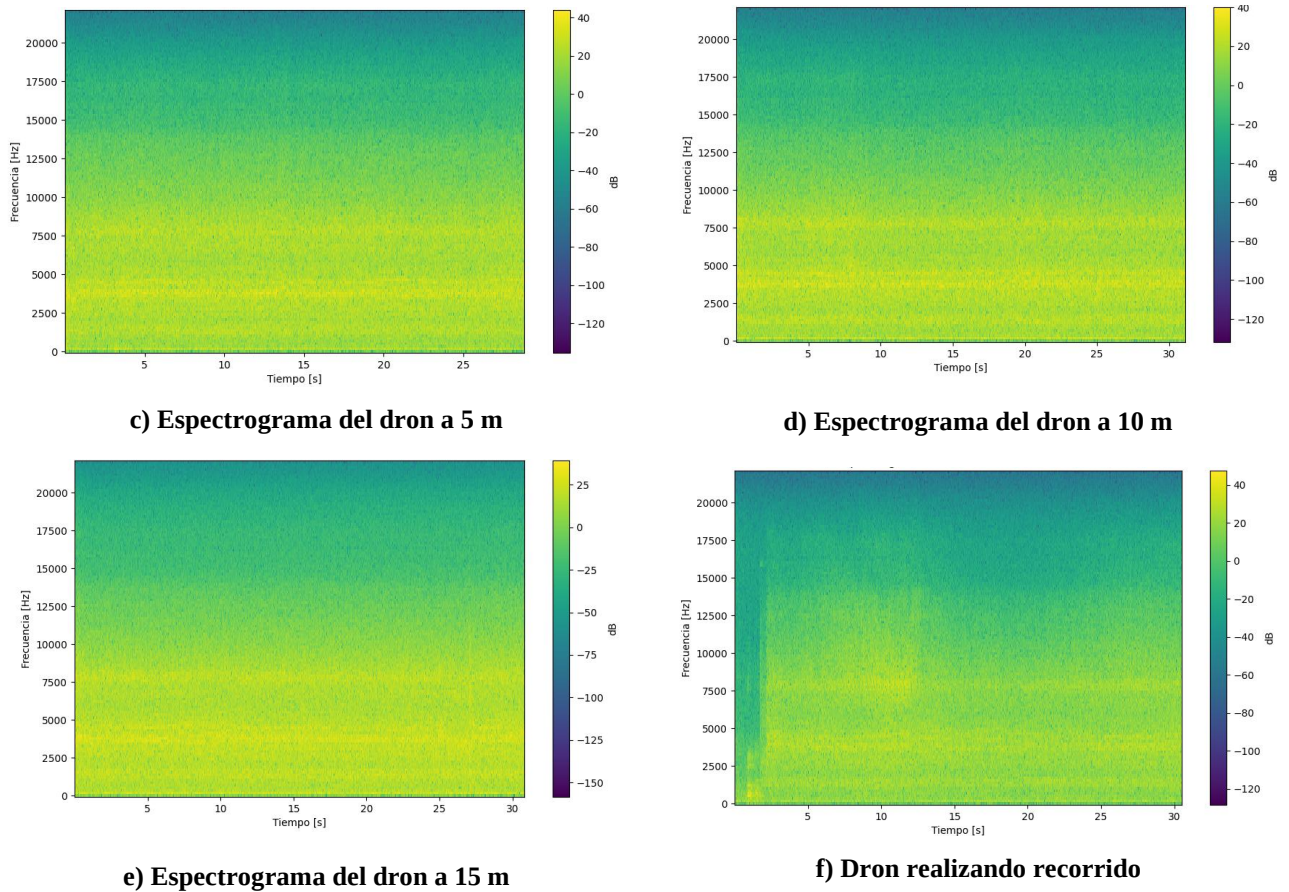
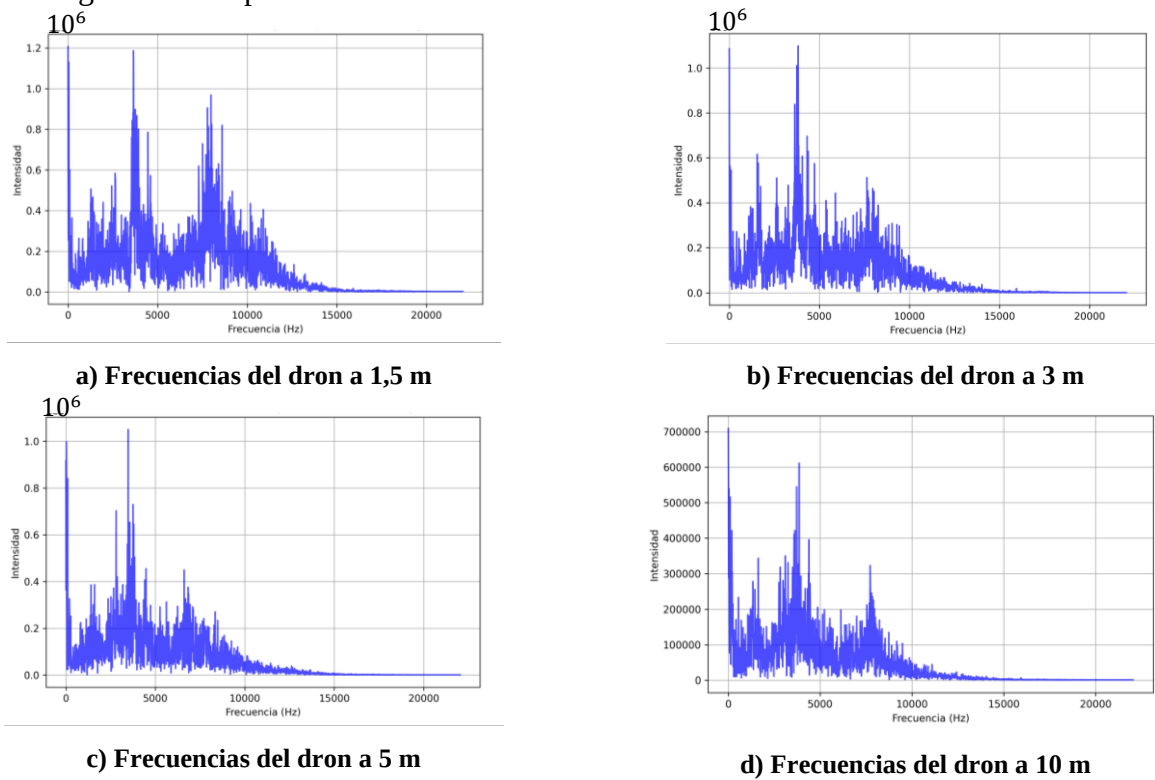
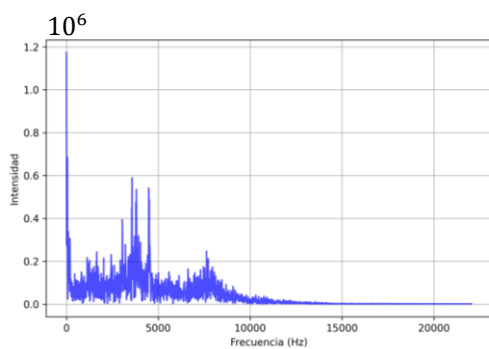


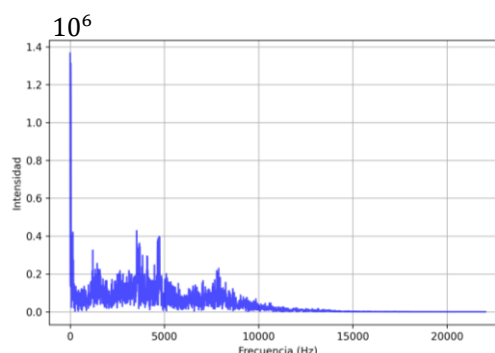
Figura 4-11 Espectrogramas del dron con micrófono Logitech C615

En la Figura 4-12 se pueden observar los análisis detallados de frecuencias a cada distancia:





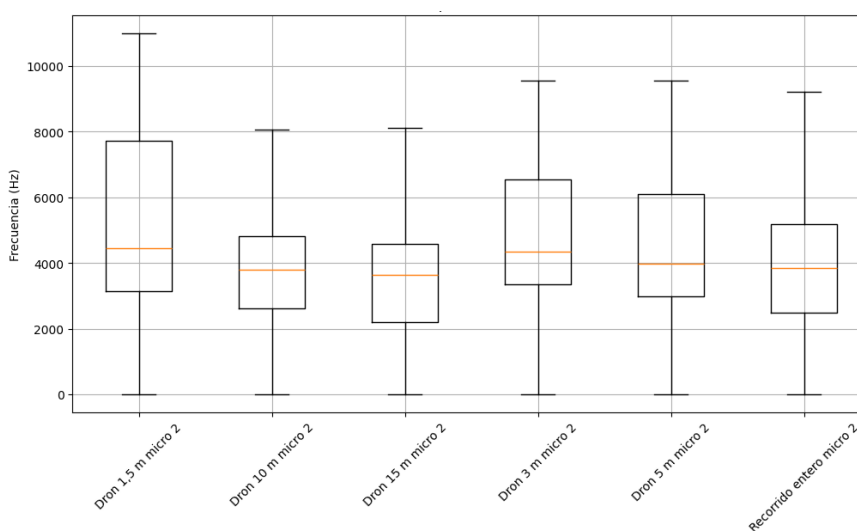
e) Frecuencias del dron a 15 m



f) Frecuencias del dron realizando recorrido

Figura 4-12 Análisis frecuencial del dron con micrófono Logitech C615

Finalmente, para hacer un análisis global, se genera el *boxplot* anteriormente comentado. Este se puede observar en la Figura 4-13:

**Figura 4-13 Distribución de frecuencias dominantes de los distintos audios**

El análisis de los resultados obtenidos con el nuevo micrófono, mostrados en la Figura 4-11, la Figura 4-12 y la Figura 4-13, confirman que la distribución de frecuencias predominantes sigue siendo similar a la obtenida en el análisis del anterior micrófono. Esto refuerza la idea de que la firma acústica del dron se mantiene estable en la banda de 3000-6000 Hz, independientemente del micrófono utilizado, lo que sugiere que la frecuencia del ruido de los motores y hélices es una característica distintiva que se conserva a pesar de las diferencias en los dispositivos de grabación.

Sin embargo, lo que sí cambia de manera notable es la intensidad de la señal captada. En el micrófono original, la energía en la banda de interés alcanzaba valores cercanos al umbral de detección previamente establecido, permitiendo la identificación del dron en varias distancias. En contraste, con este nuevo micrófono, la intensidad medida es considerablemente menor en todas las distancias, lo que explica por qué el dron no ha sido detectado en ninguno de los casos. Esto puede deberse a la menor sensibilidad del dispositivo, a su posible filtrado interno o a una menor calidad en la captación de sonidos en el rango de frecuencias clave para la detección del dron.

Dado que las frecuencias predominantes se han mantenido, lo más adecuado sería mantener el filtro pasa-banda en el rango de 3000-6000 Hz, pero ajustar el umbral de detección a valores más bajos que se adapten a la menor intensidad de señal captada por este micrófono. Si bien esto permitirá que el sistema pueda operar con este dispositivo, será fundamental comprobar en pruebas posteriores si este nuevo umbral sigue siendo lo suficientemente robusto para evitar falsos positivos y garantizar una detección fiable del dron en distintas condiciones.

Una vez tomada la decisión de mantener el filtro anterior, procedemos al cálculo del nuevo umbral en esa franja. Para ello, aplicamos de nuevo el código “Detección_umbral”, obteniendo el resultado observado en la Figura 4-14:

```
=====
UMBRAL DE DETECCIÓN (dB)
=====
Umbral inferior: 17.51 dB
Umbral superior: 26.51 dB
=====
```

Figura 4-14 Umbral de detección para micrófono Logitech C615

Este umbral se guardará en un archivo TXT y se cargará posteriormente para el análisis en tiempo real de los audios recibidos por el micrófono Logitech 215.

4.4.5 Comprobación de funcionamiento con nuevo umbral

Con el nuevo umbral definido, se repite la prueba anterior para verificar si la detección del dron es ahora efectiva. Los resultados obtenidos se presentan en la siguiente figura:

```
=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 25.91 dB
Umbral inferior: 17.51 dB
Umbral superior: 26.51 dB
=====
🔊 🔊 🔊 Posible DRON detectado en el audio.
=====
```

a) Dron a 1,5 metros

```
=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 24.49 dB
Umbral inferior: 17.51 dB
Umbral superior: 26.51 dB
=====
🔊 🔊 🔊 Posible DRON detectado en el audio.
=====
```

b) Dron a 3 metros

```
=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 22.65 dB
Umbral inferior: 17.51 dB
Umbral superior: 26.51 dB
=====
🔊 🔊 🔊 Posible DRON detectado en el audio.
=====
```

c) Dron a 5 metros

```
=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 19.64 dB
Umbral inferior: 17.51 dB
Umbral superior: 26.51 dB
=====
🔊 🔊 🔊 Posible DRON detectado en el audio.
=====
```

d) Dron a 10 metros

```
=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 18.33 dB
Umbral inferior: 17.51 dB
Umbral superior: 26.51 dB
=====
🔊 🔊 🔊 Posible DRON detectado en el audio.
=====
```

e) Dron a 15 metros

```

=====
RESULTADO DE DETECCIÓN
=====
Nivel de dB medio en la banda 3000-6000 Hz: 18.43 dB
Umbral inferior: 17.51 dB
Umbral superior: 26.51 dB
=====
🔥🔥🔥 Posible DRON detectado en el audio.
=====

```

f) Dron realizando recorrido por la sala

Figura 4-15 Resultados micrófono Logitech C615 con nuevo umbral de detección

Los resultados obtenidos con el nuevo umbral confirman que la detección del dron ahora es efectiva en todas las distancias analizadas. La adaptación de los parámetros ha permitido que este micrófono, a pesar de su menor sensibilidad inicial, sea capaz de identificar correctamente la señal del dron dentro de la banda de 3000-6000 Hz. De hecho, un aspecto interesante es que con el otro micrófono la detección fallaba a 15 metros, en este caso el sistema sigue reconociendo la presencia del dron a esa distancia.

Con esta validación completada, el siguiente paso es integrar este ajuste en el sistema de detección en tiempo real. A partir de ahora, cada micrófono deberá analizar su señal utilizando el umbral que ha sido definido específicamente para sus características. En el siguiente y último apartado de esta prueba, detallaremos cómo se implementará esta integración antes de avanzar a la prueba final de validación.

4.4.6 Integración con sistema en tiempo real

Con el nuevo umbral definido y validado, el siguiente paso es integrar esta adaptación en el sistema de detección en tiempo real. Hasta ahora, el *script* encargado del procesamiento utilizaba un único umbral global para analizar la señal captada por el micrófono, pero ahora será necesario modificar el código para que el sistema aplique el umbral correspondiente según el micrófono que esté en uso.

El ajuste consistirá en que, antes de evaluar la detección, el *script* identifique qué micrófono está capturando el audio y, en función de ello, aplique el umbral adecuado. Esto garantizará que cada dispositivo opere con los parámetros que mejor se ajustan a su respuesta en sensibilidad. Este cambio es fundamental para la prueba final de validación, ya que en ella se trabajará con múltiples micrófonos ubicados en distintas salas, cada uno con su propia configuración.

Antes, la carga del umbral de detección se realizaba directamente asignando el archivo correspondiente a una variable y leyendo sus valores de manera manual. Ahora, en lugar de repetir este proceso cada vez que se necesite un nuevo umbral, se ha definido una función específica que se encarga de leer y procesar los archivos de umbral de forma estructurada.

```

def cargar_umbral(archivo):
    with open(archivo, "r") as f:
        lines = f.readlines()
        umbral_inf=float(lines[0].split(":")[1].strip().replace("
dB",""))
        umbral_sup=float(lines[1].split(":")[1].strip().replace("
dB",""))
    return umbral_inf, umbral_sup

```

Con esta función, se simplifica la carga de los umbrales, permitiendo definir múltiples archivos de configuración sin necesidad de repetir código. Así, se establecen dos archivos de umbral: uno para la Raspberry Pi 1 y otro para la Raspberry Pi 2:

```

umbral_file1 = "umbral_deteccion.txt"

```

```

umbral_file2 = "umbral_deteccion2.txt"
umbral_inferior1, umbral_superior1 = cargar_umbral(umbral_file1)
umbral_inferior2, umbral_superior2 = cargar_umbral(umbral_file2)

```

Finalmente, para asegurarse de que cada audio se evalúe con el umbral adecuado, se implementa una condición que verifica a qué dispositivo pertenece el archivo. Si el archivo proviene de Raspberry Pi 2, se usa su respectivo umbral; en caso contrario, se utiliza el umbral de Raspberry Pi 1.

```

if "RaspberryPi2" in file_path:
    umbral_inf, umbral_sup = umbral_inferior2, umbral_superior2
else:
    umbral_inf, umbral_sup = umbral_inferior1, umbral_superior1

```

Esto es especialmente importante porque la Raspberry Pi 2 está conectada al micrófono *Logitech C615*, lo que puede influir en los niveles de detección y requerir ajustes específicos en los umbrales.

Con esta actualización, hemos logrado que cada Raspberry Pi utilice su propio umbral de detección, asegurando que los archivos de audio sean evaluados con los valores más adecuados según el dispositivo que los haya capturado. Gracias a la implementación de una función para la carga de umbrales y la lógica de selección automática, el sistema ahora puede manejar diferentes configuraciones sin necesidad de modificaciones manuales.

Con estos ajustes, el sistema de detección en tiempo real está completamente preparado para la prueba de validación, donde se evaluará su desempeño bajo condiciones realistas de transmisión y recepción. Ahora podremos comprobar si la adaptación de los umbrales mejora la precisión en la detección de drones en función de los distintos micrófonos y dispositivos empleados.

4.5 Escenario de validación: Red de dos sensores con detección del dron en tiempo real

Tras un extenso proceso de desarrollo y optimización, el sistema de detección en tiempo real está listo para someterse a su prueba definitiva. En esta prueba, se evaluará si la red de sensores es capaz de identificar un dron de manera estable y precisa, garantizando su correcto funcionamiento en condiciones realistas. No se trata solo de confirmar que el sistema detecta correctamente la presencia del UAV, sino de comprobar su fiabilidad en todo el proceso, desde la comunicación entre dispositivos hasta la correcta aplicación de los umbrales de detección.

Aunque en esta prueba se emplea un ordenador y dos Raspberry Pi, la estructura de la red de sensores está concebida para ser escalable, permitiendo su adaptación a configuraciones más amplias en entornos de mayor complejidad. La clave del éxito radica en la coherencia de la detección, la correcta sincronización de los sensores y la capacidad de respuesta del sistema ante la presencia del dron.

4.5.1 Escenario propuesto

Se ha definido un escenario en el que un UAV se infiltra en el Cuartel Marqués de la Victoria de la Escuela Naval Militar a través de su puerta principal, con el objetivo de llevar a cabo una misión de reconocimiento y obtención de inteligencia. Una vez dentro, el dron asciende a la primera planta para comenzar su reconocimiento, pasando primero por una sala que denominaremos Sala 1, y posteriormente avanzando hacia la Sala 2, desde donde continuará su trayectoria hasta el pasillo principal.

Para la detección, se han distribuido los sensores estratégicamente en distintos puntos del recorrido. La Raspberry Pi 1 (Pi1) se ha ubicado en la Sala 1, donde el dron realizará su primera incursión. A continuación, en la Sala 2, se ha instalado la Raspberry Pi 2 (Pi2), encargada de registrar la presencia

del UAV cuando este avance en su trayectoria. Mientras tanto, el operador del sistema de detección se encontrará en una sala independiente, donde supervisará el funcionamiento de ambas Raspberry Pi y analizará en tiempo real los datos captados por los sensores.

Se analizará en qué puntos del recorrido es detectado por cada sensor, verificando si la detección ocurre de manera individual en cada Raspberry Pi o si ambas detectan simultáneamente la presencia del UAV. Asimismo, se registrará con precisión el momento exacto en el que cada sensor deja de percibir la señal del dron, lo que permitirá evaluar la cobertura y sensibilidad del sistema en función de la ubicación del UAV. En la Figura 4-16 se puede observar la distribución:

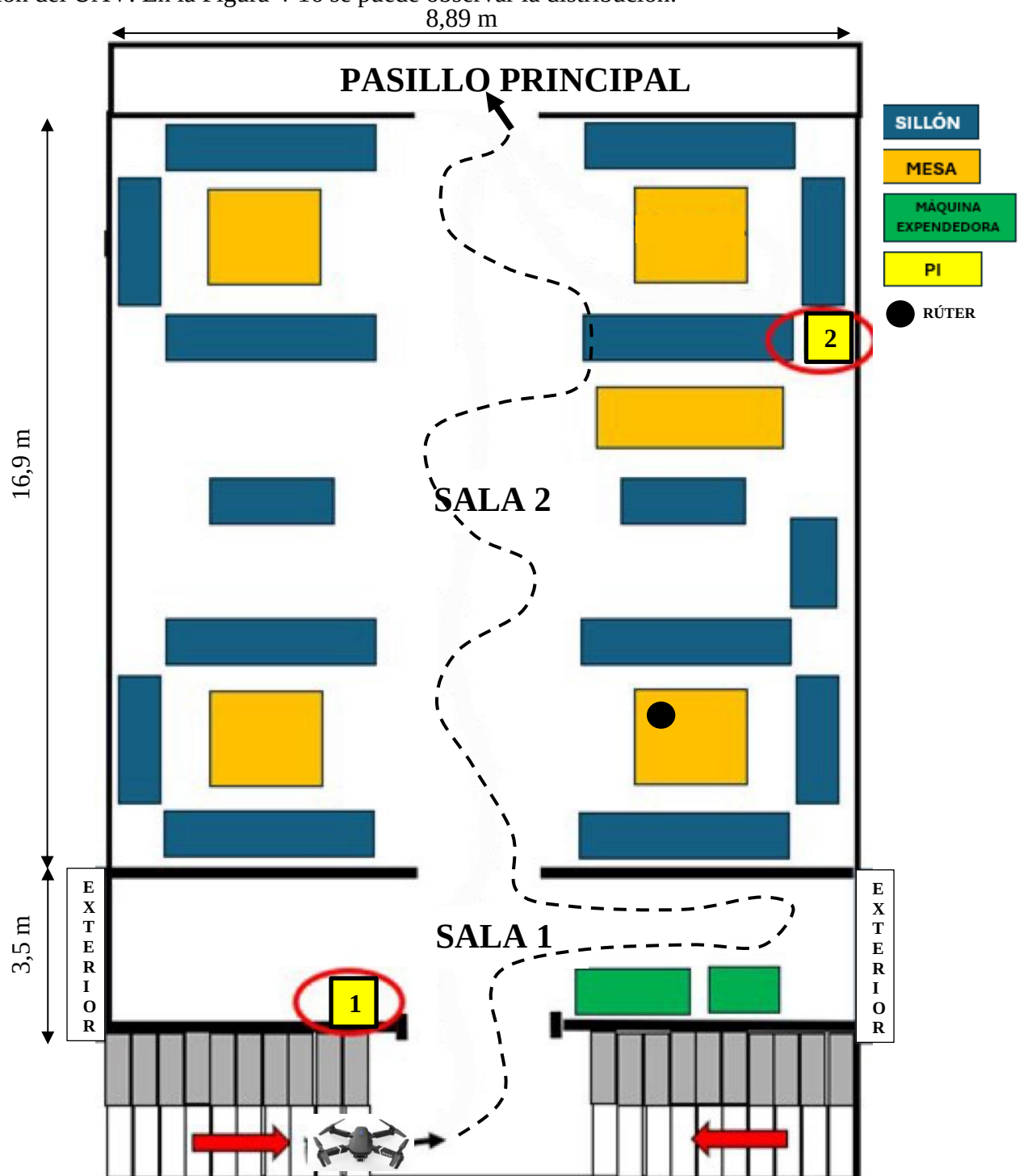
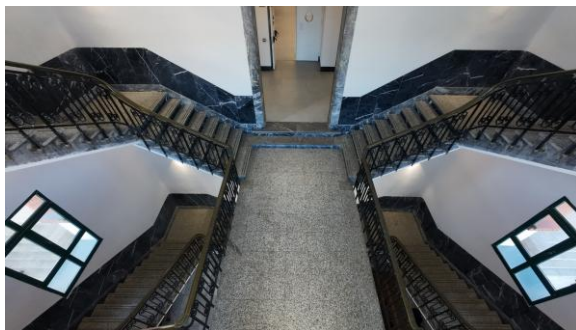


Figura 4-16 Esquema de maniobra del UAV y distribución de sensores

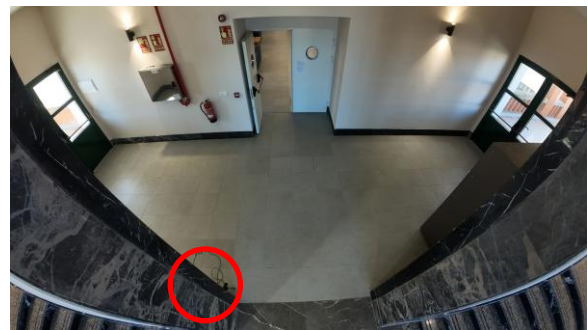
La Figura 4-16 representa la primera planta del edificio. En la parte inferior de la imagen se encuentran las escaleras de acceso, que marcan el punto desde el cual el dron ha ascendido tras su infiltración inicial. Una vez arriba, el UAV atraviesa la entrada y sigue una trayectoria predefinida que lo conduce a través de la Sala 1 y posteriormente a Sala 2, hasta alcanzar el pasillo principal.

La distribución del espacio muestra una serie de elementos de mobiliario, incluyendo sillones, mesas y máquinas expendedoras que podrían afectar la propagación del sonido y, en consecuencia, influir en la precisión de la detección. Dentro de este entorno, se han posicionado estratégicamente las Raspberry Pi, señalizadas en la imagen. Pi1 se encuentra en la parte izquierda de Sala 1, próxima a la entrada desde las escaleras, garantizando la cobertura de la zona inicial del recorrido. Pi2, en cambio, está ubicada en la parte derecha de Sala 2, lo que le permitirá detectar el paso del UAV cuando este avance hacia el pasillo.

La posición de los sensores permitirá analizar en qué momento y en qué zona el dron es detectado, si ambas Raspberry Pi lo identifican simultáneamente en algún punto y en qué instante desaparece su señal. En la Figura 4-17 se puede observar imágenes reales del escenario.



a) Entrada por las escaleras



b) Sala 1 o pre-hall donde se ve Pi1



c) Sala 2 o Hall



d) Pi2 posicionado en sala 2 antes de la llegada al pasillo principal

Figura 4-17 Imágenes del escenario

4.5.2 Secuencia de simulación

Se ha elaborado un esquema en el que se seguirán las distintas fases que conforman la secuencia completa del UAV, desde su entrada en el cuartel hasta la infiltración en el pasillo principal. Esta prueba se ha realizado el día 18 de marzo a las 18:00, se irán detallando los pasos adjuntando la hora para mejorar la comprensión. Este esquema se puede observar en la Figura 4-18:

18:25

MAC Address	In ACL	Last IP	Uptime		Signal Strength
10:68:38:05:1B:22	no	192.168.88.247	00:02:36	-47	
B8:27:EB:89:93:8E	no	192.168.88.248	00:13:05	-61	
B8:27:EB:E8:8F:48	no	192.168.88.246	00:11:28	-58	

Establecimiento de la red de sensores

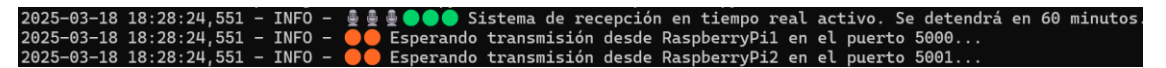
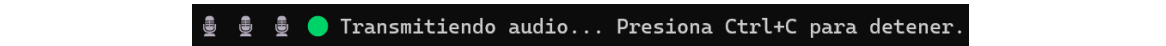
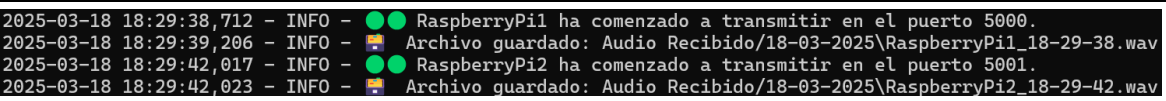
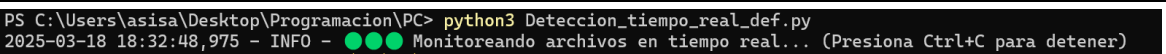
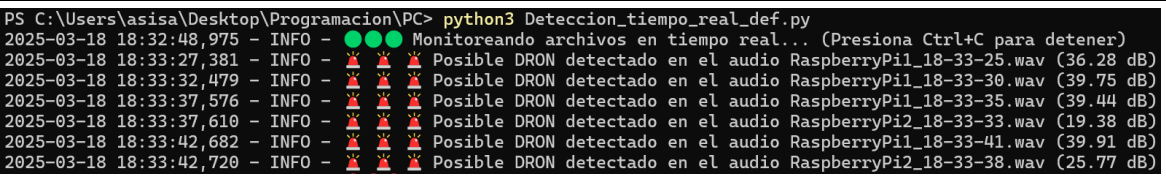
	 Contamos con las cuatro ventanas de terminal para ejecutar los programas: ○ Dos terminales de Windows para ejecutar programas recepción y detección de audio ○ Dos terminales de Raspberry Pi mediante ssh para ejecutar transmisión de audio
18:28	 Ejecutamos recepción de audio en tiempo real en uno de los terminales, el sistema espera.
18:29	 Ejecutamos simultáneamente transmisión de audio desde el terminal de cada Raspberry Pi
18:29	 El sistema de recepción detecta ese inicio y comienza a recibir y guardar audio. Ambas Raspberry almacenan el archivo “inicio” de 0 segundos.
18:32	 Ejecutamos detección de audio. A partir de este momento se comienzan a analizar todos los audios recibidos y aplicarles el umbral y filtro establecidos
18:33	 UAV hace su entrada en el cuartel y sube las escaleras
18:33 - 18:34	 Detecciones obtenidas durante su tránsito por las salas
18:34	Detención de programas, sistema de detección deja de analizar audios y avisa al usuario, finalmente se fuerza detención de sistema de detección En Raspberry Pi:  En ordenador: 

Figura 4-18 Secuencia prueba de validación

4.5.3 Resultados

Los resultados individuales de cada segmento de audio de 5 segundos se almacenan dinámicamente en la carpeta “Resultados detección”. Cada registro incluye el nombre del archivo, que refleja la hora de

captura, la energía promedio del sonido medida en decibelios y la clasificación del evento en función de si la intensidad registrada se encuentra dentro del umbral de detección definido para el micrófono correspondiente. En la Figura 4-19 podemos ver una comparación de un resultado con dron y otro sin dron.

Archivo: RaspberryPi1_18-33-25.wav	Archivo: RaspberryPi1_18-33-20.wav
Energía promedio: 36.28 dB	Energía promedio: 27.87 dB
Resultado: 🚀 Dron detectado	Resultado: ❌ No es un dron

Figura 4-19 Comparación de archivo con y sin dron

En este caso, el archivo “RaspberryPi1_18-33-25.wav” presenta una energía promedio de 36,28 dB, lo que está dentro de los límites del umbral establecido y da como resultado la detección de un dron. Por su parte, el archivo “RaspberryPi1_18-33-20.wav” registra una energía promedio de 27,87 dB, valor que queda por debajo del límite inferior del umbral, determinando que no se trata de un dron.

En la Figura 4-20 se presentan los seis archivos que describen el audio en el que se percibía un dron, indicando qué Raspberry lo detecta y su intensidad.

Archivo: RaspberryPi1_18-33-25.wav	Archivo: RaspberryPi1_18-33-30.wav
Energía promedio: 36.28 dB	Energía promedio: 39.75 dB
Resultado: 🚀 Dron detectado	Resultado: 🚀 Dron detectado
Archivo: RaspberryPi2_18-33-33.wav	Archivo: RaspberryPi1_18-33-35.wav
Energía promedio: 19.38 dB	Energía promedio: 39.44 dB
Resultado: 🚀 Dron detectado	Resultado: 🚀 Dron detectado
Archivo: RaspberryPi2_18-33-38.wav	Archivo: RaspberryPi1_18-33-41.wav
Energía promedio: 25.77 dB	Energía promedio: 39.91 dB
Resultado: 🚀 Dron detectado	Resultado: 🚀 Dron detectado

Figura 4-20 Detecciones

Los resultados de detección a nivel global, es decir, todos los eventos registrados durante un día, se almacenan en un archivo TXT diario generado automáticamente. De este modo, se crea un historial detallado que permite recopilar y analizar toda la actividad de detección ocurrida en una jornada, facilitando la trazabilidad de los datos y la evaluación del rendimiento del sistema. Este archivo se puede observar en la Figura 4-21.

deteccion_2025-03-18 →

```

2025-03-18 18:32:48,975 - INFO - 🟢🟢🟢 Monitoreando archivos en tiempo real... (Presiona Ctrl+C para detener)
2025-03-18 18:33:27,381 - INFO - 🚀🚀🚀 Posible DRON detectado en el audio RaspberryPi1_18-33-25.wav (36.28 dB)
2025-03-18 18:33:32,479 - INFO - 🚀🚀🚀 Posible DRON detectado en el audio RaspberryPi1_18-33-30.wav (39.75 dB)
2025-03-18 18:33:37,576 - INFO - 🚀🚀🚀 Posible DRON detectado en el audio RaspberryPi1_18-33-35.wav (39.44 dB)
2025-03-18 18:33:37,610 - INFO - 🚀🚀🚀 Posible DRON detectado en el audio RaspberryPi2_18-33-33.wav (19.38 dB)
2025-03-18 18:33:42,682 - INFO - 🚀🚀🚀 Posible DRON detectado en el audio RaspberryPi1_18-33-41.wav (39.91 dB)
2025-03-18 18:33:42,720 - INFO - 🚀🚀🚀 Posible DRON detectado en el audio RaspberryPi2_18-33-38.wav (25.77 dB)
2025-03-18 18:34:07,331 - INFO - 🛑🛑🛑 Detección detenida por el usuario.
  
```

Figura 4-21 Resultado diario de detección

En la Figura 4-21 se observa la prueba anterior y su registro diario, en donde se documenta la detección de drones a lo largo del tiempo, incluyendo toda la información que ha mostrado al operario ese día. Este sistema de almacenamiento garantiza un control exhaustivo del comportamiento del detector en el tiempo y permite revisar eventos pasados con precisión.

Es interesante analizar la información que nos proporciona la hora en la que se emiten los mensajes de *log* con el nivel INFO. Si comparamos estos registros con la hora a la que se ha recibido cada audio (indicada en el nombre del archivo), podemos extraer las siguientes conclusiones:

- Un audio recibido a las 18:33:30 (Archivo con nombre “RaspberryPi1_18-33-30”) corresponde al sonido captado en los últimos 5 segundos, es decir, desde 18:33:25 hasta 18:33:30.
- El resultado de la detección se produce, según el mensaje INFO, a las 18:33:27, esto indica que el análisis del audio y la notificación al operario tardan aproximadamente 2-3 segundos.

Este análisis es crucial, ya que, si un dron realiza una pasada rápida por la zona, aunque el sistema lo detecte, es posible que en el momento de la alerta ya se encuentre lejos de su posición original, pues desde que el dron está presente (18:33:25) hasta que se avisa al operario (18:33:32) pasan 7 segundos en los que el dron haber cambiado de sala ya. No obstante, la mera detección del dron ya es un avance, y consideramos 7 segundos un tiempo relativamente bueno para su rastreo.

En la Tabla 4-3 se presenta un resumen de los resultados obtenidos en la prueba, donde se detalla el número total de detecciones registradas y la participación de cada Raspberry Pi en el proceso de detección, incluidas las energías máximas y mínimas.

Raspberry Pi	Detecciones registradas	Energía máxima	Energía mínima
Pi1	4	39,91 dB	36,28 dB
Pi2	2	25,77 dB	19,38 dB

Tabla 4-3 Resumen de resultados de la prueba

Como se esperaba, existe una la diferencia en los niveles de energía detectados por cada Raspberry Pi, un aspecto ya analizado previamente en el estudio de sensibilidad de los micrófonos. Como se ha comprobado, esta variación se debe a que la Raspberry Pi 1 utiliza el *TONOR TC30*, mientras que la Raspberry Pi 2 emplea el *Logitech C615*, lo que influye en la captación de la señal y en la respuesta del sistema de detección.

4.5.4 Discusión de resultados

La prueba realizada no solo ha permitido evaluar la capacidad del sistema de detección de drones, sino también la estabilidad y eficiencia de la red de sensores establecida. El sistema ha mostrado una conectividad estable entre dispositivos, con una transmisión en tiempo real fluida y sin interrupciones.

En términos de infraestructura de la red de sensores, la transmisión de datos ha sido constante a pesar de las barreras físicas entre los dispositivos. La conexión inalámbrica ha permitido que los audios captados por cada Raspberry Pi sean enviados y analizados sin pérdidas de información, garantizando una cobertura estable del sistema. Un punto clave ha sido la capacidad del sistema para procesar y transmitir un alto volumen de datos en tiempo real, permitiendo la recepción de múltiples audios en intervalos de cinco segundos sin colapsar la red. A pesar de ello, se ha observado una diferencia significativa en los tiempos de procesamiento entre las dos unidades. Los audios recibidos por la Raspberry Pi 1 han sido analizados y procesados con mayor rapidez que los de la Raspberry Pi 2. Al comparar los registros de *log* con la hora de recepción de los archivos, se ha determinado que el ordenador tarda aproximadamente 1-2 segundos en analizar los audios de la Raspberry Pi 1 y reportar la detección al operador, mientras que los de la Raspberry Pi 2 presentan una latencia mayor, llegando a tardar 4-5 segundos. Esta diferencia se puede observar en la Figura 4-22:

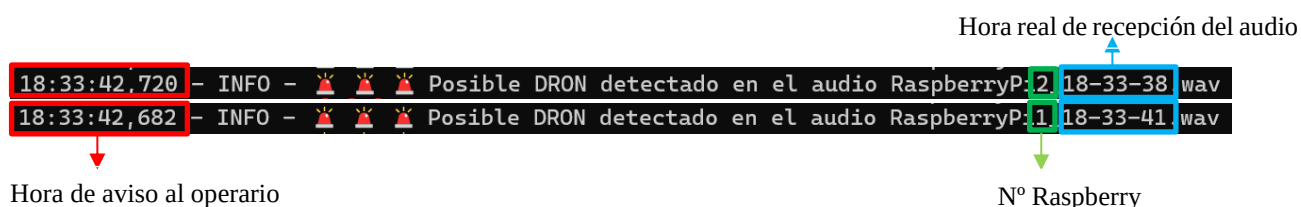


Figura 4-22 Comparación de latencias

La causa de esta diferencia parece estar relacionada con la distancia de cada Raspberry Pi al rúter principal. La Raspberry Pi 2, ubicada en la Sala 2, está más alejada del punto de acceso, lo que podría estar influyendo en el tiempo que tarda en transmitir cada archivo. Esta discrepancia en los tiempos de

análisis no afecta la detección en sí, pero sí representa un factor a considerar en términos de eficiencia del sistema, especialmente en escenarios donde la inmediatez de la alerta es crucial.

En cuanto al sistema de detección, para analizar los resultados se reutilizará el esquema que muestra el recorrido del dron (Figura 4-16). De esta manera, se podrá visualizar con claridad en qué momentos es detectado por Pi1, Pi2 o por ambas simultáneamente, así como identificar los instantes en los que se alcanzan los máximos y mínimos de energía. Esta información se presenta en la Figura 4-23:

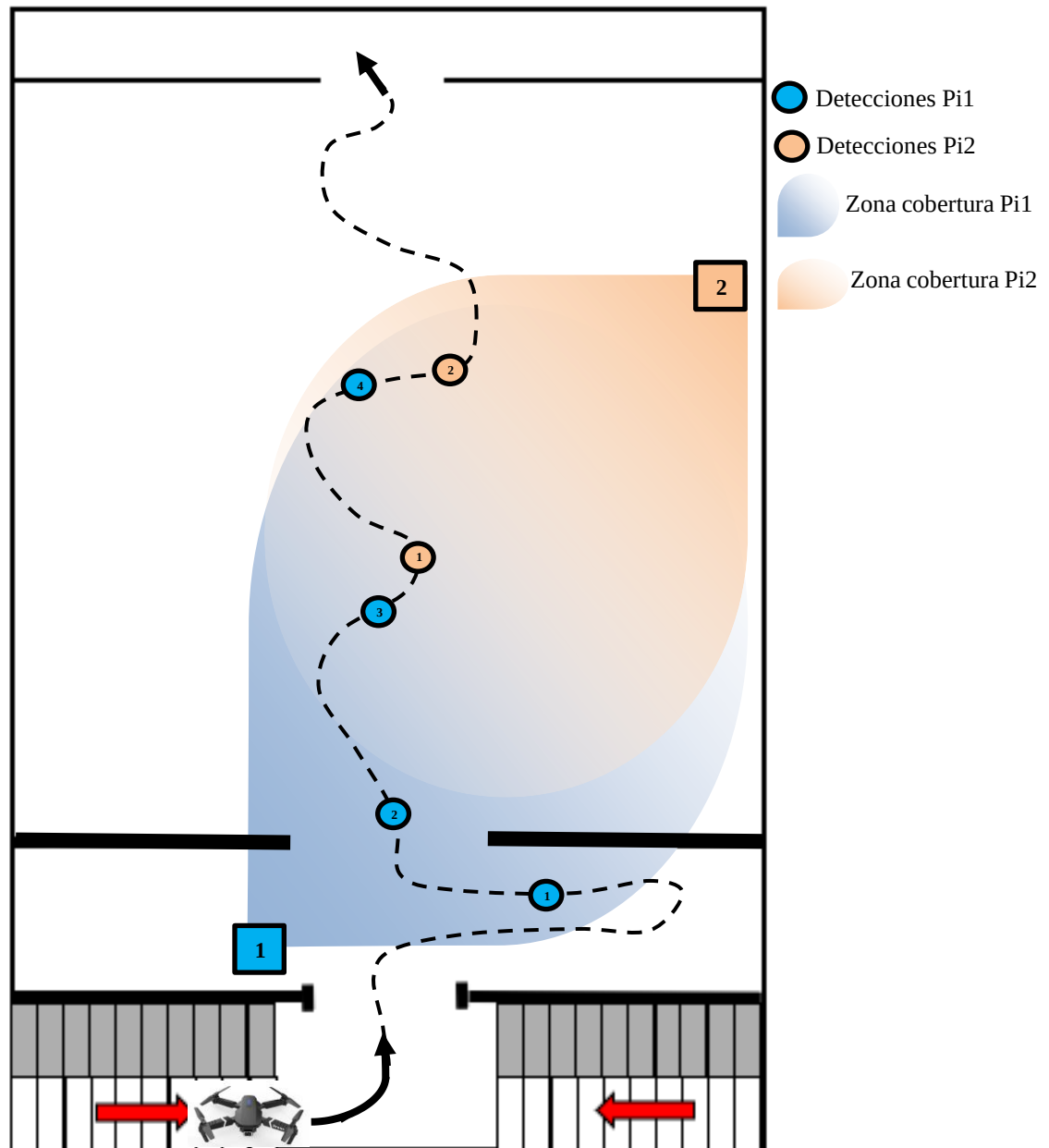


Figura 4-23 Resumen de resultados en esquema de maniobra del UAV

El análisis del recorrido del UAV ha demostrado que la detección está fuertemente condicionada por la zona de cobertura de cada Raspberry Pi. En total, el sistema ha identificado la presencia del dron en seis momentos distintos, lo que significa que, durante 30 segundos de su recorrido, el dron ha sido correctamente localizado en ambas salas a lo largo de su recorrido. Es importante recordar que la detección de cada audio corresponde a los cinco segundos previos, por lo que, aunque la Raspberry Pi 2 parece registrar la presencia del UAV cuando ya se encuentra bien adentrado en la Sala 2, en realidad lo ha estado captando desde su entrada. Lo mismo ocurre con la Raspberry Pi 1 en la Sala 1, donde su

primera detección corresponde al instante en el que el UAV está apenas ingresando en su rango de cobertura. Para un trayecto que ha durado aproximadamente un minuto, esta tasa de detección es significativa y muestra que, con una mayor densidad de sensores, la cobertura podría mejorar considerablemente.

Es importante destacar que, aunque la cobertura de cada sensor pueda parecer limitada en términos absolutos, el escenario en el que se ha realizado la prueba es una sala de $16,9 \times 8,9$ metros, un espacio amplio donde la dispersión del sonido es considerable. En entornos más reducidos, como pasillos estrechos o habitaciones de menor tamaño, es altamente probable que la detección sea mucho más efectiva. Además, en esta prueba solo se han utilizado dos sensores, por lo que una expansión del sistema con más unidades permitiría reducir las zonas ciegas y mejorar la precisión en la localización del UAV.

Por otro lado, los patrones de detección confirman la influencia del rango efectivo de cada sensor y la orientación del dron en su propagación acústica. La Raspberry Pi 1 ha detectado el UAV hasta aproximadamente 15 metros de distancia, un resultado que ya se había observado en la primera prueba y que confirma la estabilidad del sistema. A partir de este punto, la señal acústica se atenúa progresivamente hasta quedar fuera del umbral de detección, evidenciando la limitación natural del rango de cobertura del micrófono en este entorno específico. Esta atenuación se puede observar en la Figura 4-24:

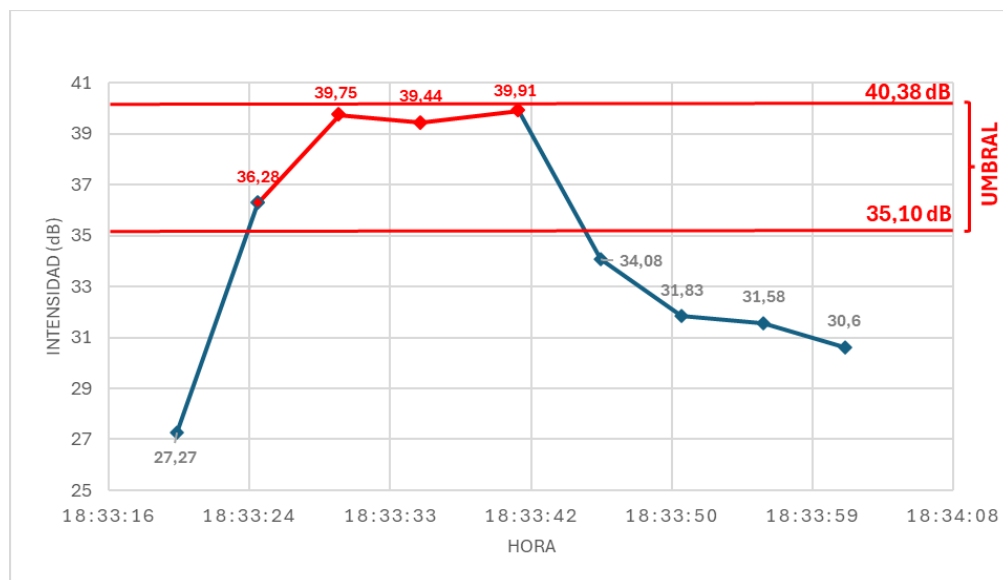


Figura 4-24 Resultados Raspberry Pi 1

Se aprecia claramente cómo la intensidad del sonido captado aumenta progresivamente mientras el UAV permanece dentro de la Sala 1. Posteriormente, a medida que el UAV se aleja hacia la Sala 2, la energía registrada disminuye gradualmente, en coherencia con la atenuación esperada por distancia. Este patrón confirma la correspondencia entre la propagación acústica y la capacidad de detección del sistema.

Además, se observa un caso interesante en el audio registrado entre las 18:33:41 y 18:33:46. Aunque la intensidad previa era alta (39,91 dB), la media de este periodo disminuye lo suficiente como para quedar fuera del umbral de detección. Esto ejemplifica claramente cómo el procesamiento basado en intervalos de cinco segundos influye en la percepción del sistema, registrando el declive de la señal con precisión y reflejando la transición entre una detección efectiva y la salida del rango de cobertura.

En el caso de la Raspberry Pi 2, la cobertura debería haber sido suficiente para identificar el dron cuando se aproximaba al pasillo principal, pero en ese punto concreto no se ha producido ninguna detección. La alta intensidad registrada en algunos puntos del recorrido se debe a la proximidad del dron al micrófono, lo que provoca un aumento significativo en los niveles de sonido detectados. Esta situación

ya había sido observada en la primera prueba y, como se mencionó anteriormente, no representa un problema, ya que el UAV ha sido detectado con antelación en distancias mayores antes de alcanzar estos valores elevados. Este comportamiento confirma que el sistema es capaz de registrar la presencia del dron antes de que llegue a una distancia crítica. Estos resultados se pueden observar en la Figura 4-25:

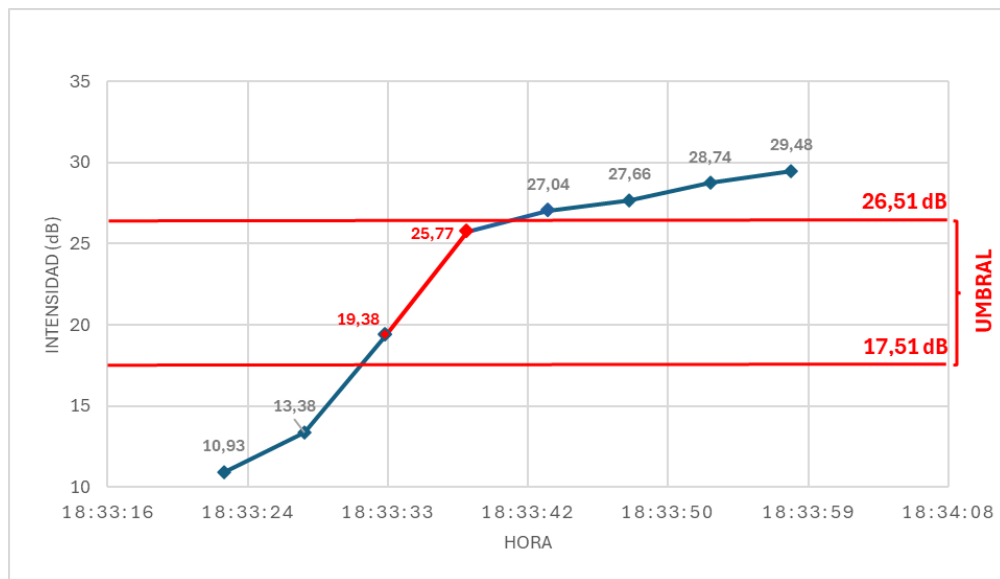


Figura 4-25 Resultados Raspberry Pi 2

Un aspecto particularmente interesante es el audio registrado entre las 18:33:38 y 18:33:43, cuya energía promedio coincide muy cerca con el límite superior del umbral de detección. Este resultado sugiere que un ajuste del factor k (por ejemplo, $k = 1,6$) podría evitar este tipo de situaciones en las que un valor muy cerca de los límites no es reconocido como una detección válida. Sin embargo, dado que el sistema ha sido configurado de esta manera, el comportamiento observado es coherente con los criterios establecidos previamente.

Lo sucedido plantea la duda de si el límite superior de detección es realmente necesario. Como se observa en los resultados de la Raspberry Pi 2, durante un intervalo de 20 segundos, el dron no es detectado a pesar de encontrarse muy próximo al micrófono. Anteriormente, se destacó que, antes de alcanzar estas distancias tan cortas, ya había sido identificado en puntos previos, lo que refuerza la idea de que lo fundamental es confirmar su presencia en la sala. Sin embargo, sigue siendo válido cuestionarse qué impacto tendría eliminar el umbral superior y si ello mejoraría la continuidad de la detección.

4.5.5 Eliminación de umbral superior

Al eliminar el umbral superior, obtenemos los mismos resultados en la Raspberry Pi 1, pues ninguno de los valores superaba el límite. No obstante, en la Raspberry Pi 2 pasaría a obtener más detecciones, como se puede observar en la Figura 4-26:

```

2025-03-18 18:32:48,975 - INFO - ●●●●● Monitoreando archivos en tiempo real... (Presiona Ctrl+C para detener)
2025-03-18 18:33:27,381 - INFO - 🚨🚨🚨 Posible DRON detectado en el audio RaspberryPi1_18-33-25.wav (36.28 dB)
2025-03-18 18:33:32,479 - INFO - 🚨🚨🚨 Posible DRON detectado en el audio RaspberryPi1_18-33-30.wav (39.75 dB)
2025-03-18 18:33:37,576 - INFO - 🚨🚨🚨 Posible DRON detectado en el audio RaspberryPi1_18-33-35.wav (39.44 dB)
2025-03-18 18:33:42,610 - INFO - 🚨🚨🚨 Posible DRON detectado en el audio RaspberryPi2_18-33-33.wav (19.38 dB)
2025-03-18 18:33:42,682 - INFO - 🚨🚨🚨 Posible DRON detectado en el audio RaspberryPi1_18-33-41.wav (39.91 dB)
2025-03-18 18:33:42,720 - INFO - 🚨🚨🚨 Posible DRON detectado en el audio RaspberryPi2_18-33-38.wav (25.77 dB)
2025-03-18 18:33:47,856 - INFO - 🚨🚨🚨 Posible DRON detectado en el audio RaspberryPi1_18-33-43.wav (27.04 dB)
2025-03-18 18:33:52,976 - INFO - 🚨🚨🚨 Posible DRON detectado en el audio RaspberryPi1_18-33-48.wav (27.66 dB)
2025-03-18 18:33:58,223 - INFO - 🚨🚨🚨 Posible DRON detectado en el audio RaspberryPi1_18-33-53.wav (28.74 dB)
2025-03-18 18:33:03,456 - INFO - 🚨🚨🚨 Posible DRON detectado en el audio RaspberryPi1_18-33-58.wav (29.48 dB)
2025-03-18 18:34:07,331 - INFO - ●●●●● Detección detenida por el usuario.

```

Figura 4-26 Resultados totales sin umbral superior

Por lo tanto, al analizar los resultados de la Raspberry Pi 2, obtenemos lo siguiente:

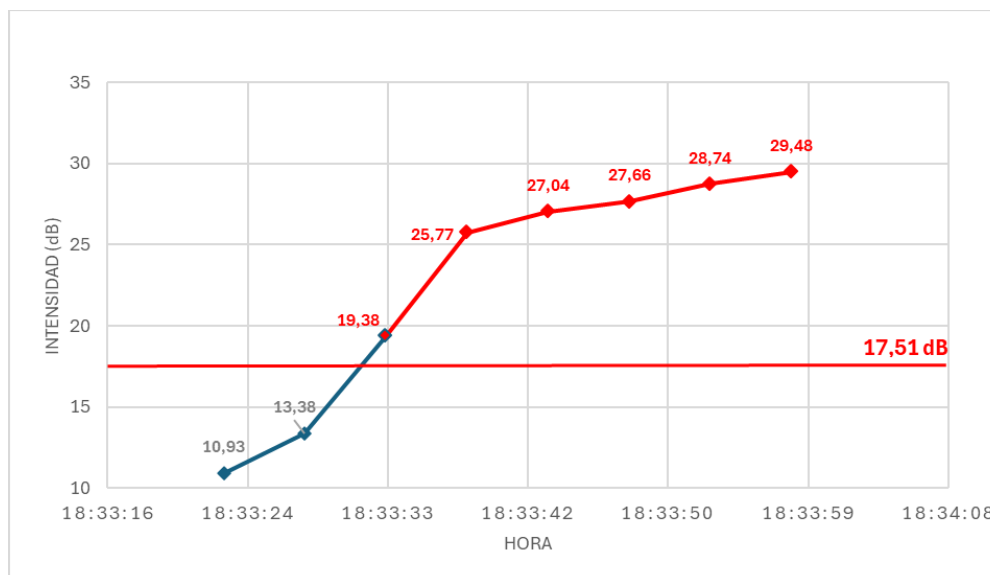


Figura 4-27 Resultados Raspberry Pi 2 sin umbral superior

El análisis de los resultados obtenidos sin el umbral superior puede parecer prometedor a primera vista, ya que la Raspberry Pi 2 registra un mayor número de detecciones. Sin embargo, es importante contextualizar estos datos dentro del escenario en el que se ha desarrollado la prueba. Hasta ahora, el entorno ha sido ideal: sin interferencias significativas, sin ruidos de fondo constantes y sin intentos de interferencia deliberada, como el *jamming*. En una situación real, el sistema de detección debe ser capaz de discriminar entre la señal generada por un dron y cualquier otra fuente de ruido que pueda encontrarse en la misma banda de frecuencias.

Precisamente, la razón por la que se estableció un umbral superior en un inicio fue para evitar que cualquier señal intensa en la banda de interés fuese interpretada erróneamente como la presencia de un dron. La eliminación de este umbral podría generar un sistema excesivamente sensible, susceptible a constantes falsas alarmas ante cualquier ruido ambiental suficientemente fuerte. Si bien es preferible un cierto margen de falsos positivos para garantizar que no se pase por alto la detección de un UAV, un exceso de activaciones erróneas supondría un problema grave, ya que haría que el sistema perdiera fiabilidad operativa.

Para evaluar de manera objetiva los efectos de esta modificación, se llevará a cabo una última prueba en la que se analizará cómo un ruido continuo, ya sea generado de forma artificial por un *jammer* o proveniente de fuentes presentes en el edificio, puede desencadenar detecciones falsas de manera recurrente. Esta prueba será clave para determinar si la eliminación del umbral superior es viable en un entorno con interferencias reales o si, por el contrario, confirma la necesidad de mantenerlo para preservar la precisión del sistema.

4.6 Prueba 5: Impacto del ruido sin umbral superior

El objetivo de esta prueba es demostrar las consecuencias de eliminar el umbral superior en el sistema de detección de drones. Se ha generado un ruido constante e intenso, que podría ser perfectamente provocado por un *jammer*, simulando una fuente de interferencia continua en la banda de detección. Hasta ahora, se ha argumentado teóricamente la importancia de este umbral para evitar falsas detecciones ante ruido ambiental o interferencias. Sin embargo, esta prueba busca evidenciar de manera práctica cómo un ruido constante dentro de la banda de detección puede desencadenar falsas alarmas si no se establece un límite superior en la clasificación. Para llevar a cabo esta prueba, se ha generado un *script* denominado “Creación_ruido.py” (Anexo XI: Código de creación de ruido)

4.6.1 Flujo de ejecución

Este *script* genera un ruido blanco de 50 segundos, filtrado en la banda de 3000-6000 Hz, y lo divide en fragmentos de 5 segundos para posteriormente ser analizado por el sistema de detección. Estos audios se pueden escuchar en el Anexo XV: QR para audios de ruido generado.

En primer lugar, se crea un ruido aleatorio mediante el comando:

```
noise = np.random.normal(0, 1, fs * duration)
```

En esta línea se genera un conjunto de datos aleatorios con distribución normal (media = 0, desviación estándar = 1), que representa ruido blanco gaussiano. Este ruido se extiende a lo largo de la cantidad de muestras necesarias según la duración (50 segundos) y la frecuencia de muestreo fs (44,1 kHz). Esto genera una señal aleatoria donde todos los valores tienen la misma probabilidad de ocurrencia en todas las frecuencias, lo que significa que inicialmente el ruido no tiene ninguna preferencia por la banda de 3000-6000 Hz.

Posteriormente, se define y se aplica el filtro paso-banda al audio generado, limitándose el audio al rango de 3000-6000 Hz, que es la banda de frecuencias característica.

```
filtered_noise = bandpass_filter(noise, 3000, 6000, fs)
```

El código divide el ruido filtrado en 10 fragmentos de 5 segundos cada uno, representando 50 segundos de ruido constante (duración equivalente a la del UAV en el escenario de validación).

```
for i in range(10):
    start_sample = i * 5 * fs
    end_sample = (i + 1) * 5 * fs
    segment = filtered_noise[start_sample:end_sample]
```

Ahora aplicamos un factor de intensidad aleatorio a cada archivo.

```
random_factor = np.random.uniform(1, 32767)
segment=(segment/np.max(np.abs(segment))*random_factor).astype(np.int16)
```

Se genera un número aleatorio entre 1 y 32,767 (límite máximo de amplitud con formato de 16 bits) que se multiplica por el segmento de audio normalizado para asignarle una amplitud variable. Este valor mínimo de 1 evita que se generen segmentos de ruido con una intensidad fija alta, permitiendo simular un entorno con variabilidad realista. La idea detrás de este procedimiento es reflejar cómo diferentes fuentes de interferencia pueden tener distintos niveles de intensidad, especialmente si son provocadas por un *jammer*, haciendo que la simulación sea más representativa de un escenario realista.

Finalmente, los audios se almacenan en una carpeta y se ponen a prueba con el sistema de detección, permitiendo comprobar si ese ruido creado intencionadamente genera falsos positivos.

4.6.2 Resultados

El análisis de los 10 audios ha mostrado que, sin el umbral superior, 6 de los 10 fragmentos han sido clasificados erróneamente como detecciones de dron, es decir, 30 de los 50 segundos han proporcionado falsos positivos. Estos resultados se pueden observar en la Figura 4-28:

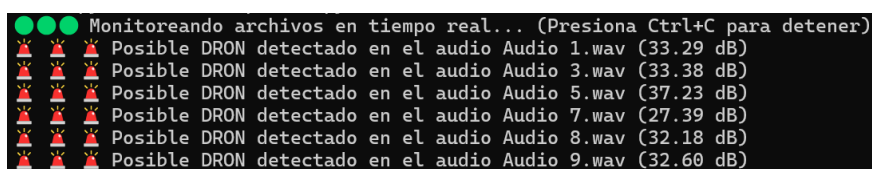


Figura 4-28 Resultados de detección de dron con ruido creado

Para observar con más detalle, se proporciona la gráfica mostrada en la Figura 4-29:

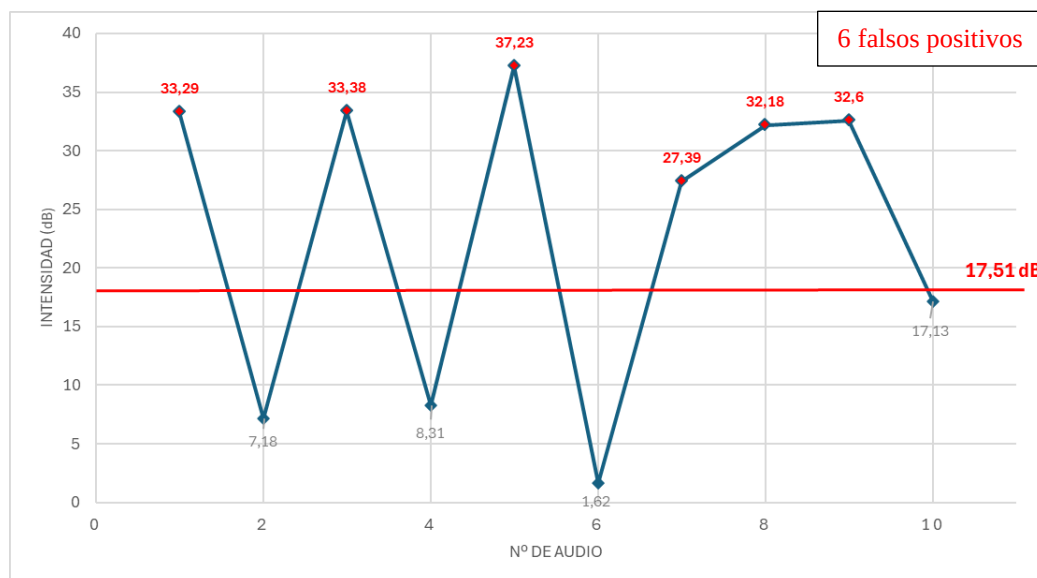


Figura 4-29 Gráfica con resultados de audio creado y un solo umbral

Esto confirma que, en un escenario sin un límite máximo, una gran parte del ruido fuerte en la banda de detección es interpretada como la presencia de un UAV, generando falsas alarmas de manera recurrente. Aunque no todos los audios han sido identificados como dron, la tasa de falsos positivos es lo suficientemente elevada como para comprometer la fiabilidad del sistema. Si bien es preferible reducir los falsos negativos, un sistema que detecta erróneamente en más de la mitad de los casos se vuelve poco fiable operativamente, ya que pierde su capacidad para discriminar eventos relevantes. En caso de haber utilizado el doble umbral, los resultados hubiesen sido los mostrados en la Figura 4-30. Como se puede observar, no habría falsos positivos.

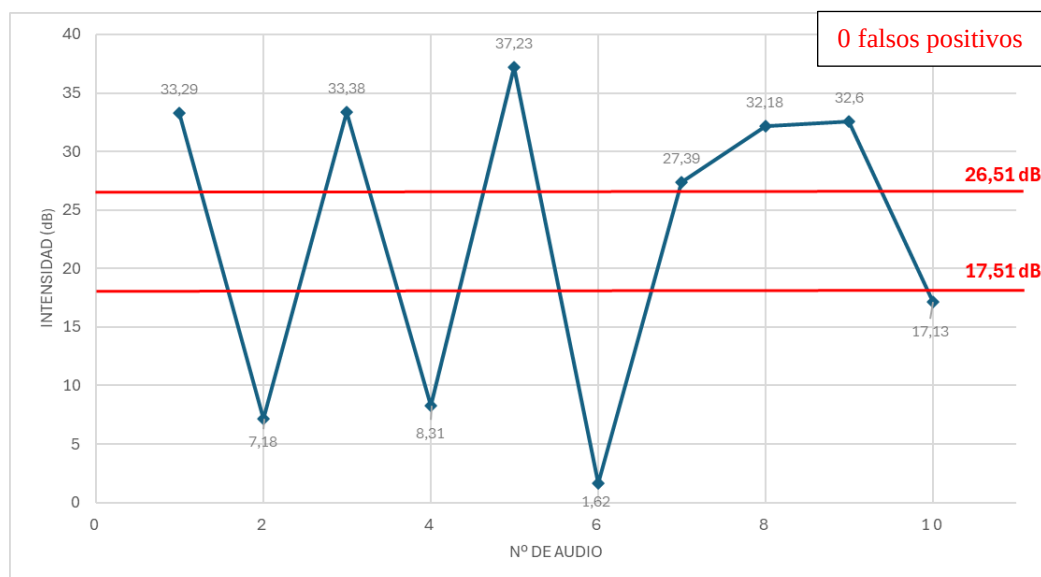


Figura 4-30 Gráfica con resultados de audio creado y ambos umbrales

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Conclusiones

El desarrollo de este trabajo ha permitido diseñar e implementar una red de sensores para la detección de drones en interiores, evaluando su viabilidad y rendimiento en distintos escenarios. A lo largo del proceso, se ha establecido un sistema basado en comunicación WiFi, utilizando microcontroladores y sensores acústicos que han demostrado su capacidad para identificar la presencia de drones mediante la detección de su firma sonora característica.

En términos de conectividad, la red inalámbrica ha operado de manera estable, permitiendo la transmisión continua de datos desde múltiples sensores sin interrupciones significativas. La elección de WiFi, a pesar de su consumo energético más elevado en comparación con otras opciones, ha resultado ser la más adecuada debido a su mayor ancho de banda y capacidad de transmisión de datos, facilitando el envío de archivos de audio cada cinco segundos sin degradación en el rendimiento. Si bien la cobertura en este experimento se ha limitado a dos salas, la estabilidad observada sugiere que el sistema podría ampliarse a áreas más extensas con la infraestructura de red adecuada.

En lo referente a la detección de drones, el sistema ha seguido un enfoque basado en la correlación de patrones, estableciendo un filtro paso-banda y un umbral de detección definidos a partir del análisis acústico del dron empleado en las pruebas. Es importante destacar que únicamente se ha detectado un dron, aquel que ha sido objeto de estudio durante el desarrollo, si bien se podría aumentar la base de datos en un futuro. La detección ha sido efectiva en condiciones controladas y validada en un entorno de transmisión en tiempo real. Se ha confirmado que los sensores acústicos empleados han sido capaces de identificar la firma sonora del UAV y compararla con los datos obtenidos en tiempo real, lo que ha permitido su detección.

Para garantizar la precisión del sistema, se han realizado grabaciones de sonidos reales del dron en un ambiente simulado, evaluando su capacidad de diferenciación frente a otros sonidos. Los resultados han demostrado que el sistema es capaz de operar de manera efectiva en entornos con condiciones acústicas complejas, cumpliendo así con el objetivo planteado en el trabajo. No obstante, se ha determinado que la detección está sujeta a la correcta calibración de umbrales y a la calidad del micrófono utilizado, aspectos que pueden ajustarse según el contexto operativo. Si bien la base de datos de detección podría ampliarse para incluir otros drones, este proceso requeriría una nueva fase de obtención de inteligencia, siguiendo el mismo procedimiento descrito en el trabajo. La identificación de nuevos UAVs implicaría un análisis detallado de sus características acústicas, su espectro de frecuencias y la generación de nuevos umbrales de detección, garantizando la precisión del sistema para cada modelo incorporado.

Otro aspecto relevante ha sido la evaluación del sistema con distintos sensores acústicos. Se ha observado que cada micrófono presenta variaciones en su respuesta en frecuencia y sensibilidad, pero el método de detección desarrollado ha demostrado ser adaptable a diferentes dispositivos mediante la recalibración de umbrales y parámetros de filtrado. Asimismo, el posicionamiento de los micrófonos ha demostrado ser un factor crítico en la detección del UAV, ya que una disposición inadecuada puede afectar la captura de la señal y la precisión en la identificación del dron. La correcta distribución de los sensores acústicos en el entorno de detección es fundamental para maximizar la cobertura y minimizar posibles puntos ciegos. Este resultado es clave, ya que sugiere que el sistema puede integrarse con una variedad de sensores acústicos sin necesidad de modificaciones estructurales significativas.

Finalmente, en la prueba de validación con la red de sensores en tiempo real, se ha demostrado que el sistema es funcional y capaz de identificar la presencia del dron con una fiabilidad considerable. Aunque se han identificado algunas limitaciones, como la pérdida de detección en ciertas situaciones debido a la configuración del umbral superior, los ajustes realizados han permitido optimizar el rendimiento del sistema. Se ha determinado que mantener un umbral superior es esencial para evitar la detección errónea de cualquier ruido como un dron, lo que podría generar numerosos falsos positivos y comprometer la fiabilidad del sistema. En pruebas adicionales, se ha verificado que eliminar el umbral superior provocaría una detección constante de ruido ambiente, incluyendo interferencias como señales de *jamming*, como si fuesen un UAV, lo que invalidaría la utilidad del sistema en entornos operativos reales.

Además, la elección de priorizar minimizar los falsos positivos, incluso a costa de la generación ocasional de falsos negativos, se fundamenta en el hecho de que la detección del dron no se pasa por alto completamente. Se ha comprobado que, aunque pueda haber momentos en los que no se detecte en ciertas ventanas temporales, el sistema ha identificado previamente su presencia en otros instantes, permitiendo confirmar con suficiente certeza la existencia de la amenaza. Este enfoque garantiza una detección más confiable y útil en escenarios tácticos donde una alerta constante y equivocada podría desencadenar respuestas innecesarias o ineficaces.

En definitiva, el objetivo de este trabajo era desarrollar una red de sensores básica para la detección de drones en interiores, y los resultados obtenidos confirman que se ha logrado con éxito.

5.2 Líneas futuras

Los resultados obtenidos han sentado una base sólida sobre la cual se pueden plantear diversas mejoras y extensiones del sistema. Entre las principales líneas futuras se destacan:

- Ampliación del sistema: Actualmente, la red de sensores ha cubierto únicamente dos salas, pero con una infraestructura de red adecuada y un mayor número de sensores, se podría expandir para monitorizar un edificio completo o incluso áreas más amplias en instalaciones estratégicas.
- Procesamiento descentralizado en las Raspberry Pi: En la implementación actual, el procesamiento de datos se realiza en el ordenador central, pero las Raspberry Pi cuentan con la capacidad para ejecutar el análisis de detección directamente en cada nodo sensor. Esto permitiría que el ordenador solo recibiera alertas de detección en lugar de procesar todos los datos de audio, reduciendo el tráfico en la red y mejorando la eficiencia del sistema. Así, se podría visualizar la ubicación de los sensores en un mapa con indicadores que señalen la detección o no de un UAV.
- Integración de sensores adicionales: La incorporación de sensores de imagen o video permitiría combinar la detección acústica con otros métodos, mejorando la precisión del sistema y reduciendo la posibilidad de errores en la identificación de drones. Esto podría ser especialmente útil en entornos urbanos complejos donde la detección acústica por sí sola podría verse limitada por la presencia de múltiples fuentes de ruido.

- Exploración de nuevas técnicas de detección: Si bien este trabajo se ha basado en la intensidad del sonido y la correlación de patrones, futuras investigaciones podrían evaluar la aplicación de técnicas de *Machine Learning* o análisis cepstrales para extraer características más avanzadas de la señal acústica del dron. Esto permitiría mejorar la capacidad del sistema para discriminar entre diferentes modelos de UAVs o adaptarse mejor a variaciones en el entorno de operación.
- Ampliación de base de datos. Hasta el momento, el estudio se ha centrado en la detección de un único dron, pero el mismo procedimiento podría aplicarse a otros modelos en función de las amenazas previstas. Incluso sería posible establecer una correlación entre distintos tipos de UAVs, utilizando las técnicas mencionadas anteriormente para identificar patrones acústicos característicos. Esto permitiría construir una base de datos extensa que no solo detecte la presencia de un dron, sino que también pueda diferenciar qué modelo específico está siendo identificado.
- Validación en escenarios más realistas: La implementación de pruebas en entornos más dinámicos, con múltiples drones o con interferencias intencionadas en la señal acústica, ayudaría a evaluar el límite del sistema y a desarrollar estrategias para mejorar su robustez frente a intentos de evasión o bloqueos electrónicos.
- Estudio de aplicación en múltiples escenarios. Se podría estudiar la viabilidad de emplear el sistema no solo para aplicaciones militares, sino también para la protección de aeropuertos o eventos de relevante importancia.

Estas mejoras contribuirían a incrementar la eficiencia, escalabilidad y aplicabilidad del sistema en contextos operativos reales, permitiendo su integración en infraestructuras de seguridad y defensa para la detección y monitoreo de drones en entornos urbanos y estratégicos.

6 BIBLIOGRAFÍA

- [1] «Aircraft detection before radar, 1917-1940 - Rare Historical Photos», <https://rarehistoricalphotos.com/>. Accedido: 18 de enero de 2025. [En línea]. Disponible en: <https://rarehistoricalphotos.com/aircraft-detection-radar-1917-1940/>
- [2] G. Douhet, *The Command of the Air*. Air University Press, 2019. Accedido: 20 de enero de 2025. [En línea]. Disponible en: https://www.airuniversity.af.edu/Portals/10/AUPress/Books/B_0160_DOUHET_THE_COMMAND_OF_THE_AIR.pdf
- [3] G. Mary, «El Impacto de los drones en la guerra moderna», Pucará Defensa. Accedido: 26 de noviembre de 2024. [En línea]. Disponible en: <https://www.pucara.org/post/el-impacto-de-los-drones-en-la-guerra-moderna>
- [4] A. Piñeiro Sánchez, «ARMADA 2050». 22 de enero de 2025. Accedido: 23 de enero de 2025. [En línea]. Disponible en: <https://armada.defensa.gob.es/html/home/ARMADA2050.pdf>
- [5] J. Trevithick, «Ukraine Using Thousands Of Networked Microphones To Track Russian Drones», The War Zone. Accedido: 2 de enero de 2025. [En línea]. Disponible en: <https://www.twz.com/land/thousands-of-networked-microphones-are-tracking-drones-in-ukraine>
- [6] H. A. Rogoway Tyler, «Ukraine's Acoustic Drone Detection Network Eyed By U.S. As Low-Cost Air Defense Option», The War Zone. Accedido: 2 de enero de 2025. [En línea]. Disponible en: <https://www.twz.com/air/ukraines-acoustic-drone-detection-network-eyed-by-u-s-as-low-cost-air-defense-option>
- [7] C. von Clausewitz, *De La Guerra*. La Esfera De Los Libros, 2014.
- [8] A. Madhu, «Wireless Sensor Network Sercurity in Military Application using Unmanned Vehicle», ago. 2014, p. 9. Accedido: 21 de enero de 2025. [En línea]. Disponible en: https://www.researchgate.net/publication/304887916_Wireless_Sensor_Network_Security_in_Military_Application_using_Unmanned_Vehicle
- [9] L. de J. M. Palacio y A. E. U. Taborda, «Armamento aéreo: una mirada hacia el futuro», *Cienc. Poder Aéreo*, vol. 16, n.º 2, 2021, Accedido: 19 de enero de 2025. [En línea]. Disponible en: <https://www.redalyc.org/journal/6735/673571919008/html/>

- [10] Armada, «Una breve Historia de la Aviación Naval Española 1917-2017 - Aeronaves - Armada - Ministerio de Defensa - Gobierno de España». Accedido: 11 de febrero de 2025. [En línea]. Disponible en: <https://armada.defensa.gob.es/ArmadaPortal/page/Portal/ArmadaEspañola/buquesaeronaves/prefLang-es/001historiaaviacion>
- [11] C. M. E. Krause, «El Poderío Aéreo en la Guerra Moderna», Accedido: 20 de enero de 2025. [En línea]. Disponible en: https://www.airuniversity.af.edu/Portals/10/ASPJ_Spanish/Journals/Volume-27_Issue-4/2015_4_05_krause_s.pdf
- [12] «Aviones de combate de la II Guerra Mundial (VIII): Boeing B-17 Flying Fortress | El Cedazo». Accedido: 21 de enero de 2025. [En línea]. Disponible en: <https://eltamiz.com/elcedazo/2012/01/31/aviones-de-combate-de-la-ii-guerra-mundial-viii-boeing-b-17-flying-fortress/>
- [13] J. García Camargo, «Poder Aéreo», mar. 2003, doi: 10.25062/0120-0631.1435.
- [14] J. A. Cebrián de Barrio, «Sistema global contra drones», Universidad de Vigo, Vigo, 2023.
- [15] J. M. García Montaña, «Plan Director de RPAS». Ministerio de Defensa, febrero de 2015. Accedido: 22 de enero de 2025. [En línea]. Disponible en: <https://www.tecnologiaeinnovacion.defensa.gob.es/Lists/Publicaciones/Attachments/199/2014%2010%2030%20PD%20RPAS%20Version%20Publica%20Maquetacion.pdf>
- [16] A. B. Micó Catalán, «Estados Unidos revela los secretísimos proyectos de miles de millones de dólares con Ucrania», ElHuffPost. Accedido: 22 de enero de 2025. [En línea]. Disponible en: <https://www.huffingtonpost.es/global/estados-unidos-revela-secretisimos-proyectos-miles-millones-dolares-ucrania.html>
- [17] Vandenbosch, «Ukrainian Baba Yaga Unmanned Aerial Vehicle Has Become True Nightmare for Russian Forces». Accedido: 23 de enero de 2025. [En línea]. Disponible en: <https://armyrecognition.com/focus-analysis-conflicts/army/conflicts-in-the-world/ukraine-russia-conflict/ukrainian-baba-yaga-unmanned-aerial-vehicle-has-become-true-nightmare-for-russian-forces>
- [18] J. Kappenman, «Army Unmanned Aircraft Systems: Decisive in Battle», Accedido: 23 de enero de 2025. [En línea]. Disponible en: <https://apps.dtic.mil/sti/tr/pdf/ADA516600.pdf>
- [19] G. J. Astaburuaga, «Drones y vehículos no tripulados en guerra | Revista de Marina». Accedido: 22 de enero de 2025. [En línea]. Disponible en: <https://revistamarina.cl/es/articulo/drones-y-vehiculos-no-tripulados-en-guerra>
- [20] L. Criollo, C. Mena-Arciniega, y S. Xing, «Classification, military applications, and opportunities of unmanned aerial vehicles», *Aviation*, vol. 28, n.º 2, Art. n.º 2, ago. 2024, doi: 10.3846/aviation.2024.21672.
- [21] «En Ucrania se ha creado una nueva categoría de municiones para los UAV», *Militarnyi*. Accedido: 23 de enero de 2025. [En línea]. Disponible en: <https://mil.in.ua/es/news/en-ucrania-se-ha-creado-una-nueva-categoria-de-municiones-para-los-uav/>
- [22] A. Balas, «UAVs in the Middle East: Coming of Age | Royal United Services Institute». Accedido: 27 de enero de 2025. [En línea]. Disponible en: <https://rusi.org/explore-our-research/publications/commentary/uavs-middle-east-coming-age>

- [23] A. Milic, A. Randelović, y M. Radovanović, «Use of drones in operations in the urban environment», Accedido: 22 de enero de 2025. [En línea]. Disponible en: https://www.researchgate.net/publication/336589680_Use_of_drones_in_operations_in_the_urban_environment
- [24] Y. Masanori, N. Atsuhiko, M. Takuya, y N. Masafumi, «Detection system and detection method», sep. 2019, Accedido: 16 de marzo de 2025. [En línea]. Disponible en: <https://scispace.com/papers/detection-system-and-detection-method-2rq05xmx86>
- [25] I. Harvey, «Sound Mirrors – Pre-radar Early Warning System | War History Online», warhistoryonline. Accedido: 27 de enero de 2025. [En línea]. Disponible en: <https://www.warhistoryonline.com/war-articles/sound-mirrors-pre-radar-early-warning-system.html>
- [26] R. R. Systems, «Why Traditional Radar Isn't Effective at Tracking Drones». Accedido: 12 de febrero de 2025. [En línea]. Disponible en: <https://www.robinradar.com/why-traditional-radar-isnt-effective-at-tracking-drones>
- [27] M. A. Khan, H. Menouar, A. Eldeeb, A. Abu-Dayya, y F. D. Salim, «On the Detection of Unauthorized Drones—Techniques and Future Perspectives: A Review», *IEEE Sens. J.*, vol. 22, n.º 12, pp. 11439-11455, jun. 2022, doi: 10.1109/JSEN.2022.3171293.
- [28] A. Bernardini, F. Mangiatordi, E. Pallotti, y L. Capodiferro, «Drone detection by acoustic signature identification», *Electron. Imaging*, vol. 29, n.º 10, pp. 60-64, ene. 2017, doi: 10.2352/ISSN.2470-1173.2017.10.IMAWM-168.
- [29] F. Svanström, F. Alonso-Fernandez, y C. Englund, «Drone Detection and Tracking in Real-Time by Fusion of Different Sensing Modalities», *Drones*, vol. 6, n.º 11, p. 317, oct. 2022, doi: 10.3390/drones6110317.
- [30] M. Ezuma, F. Erden, C. Kumar Anjinappa, O. Ozdemir, y I. Guvenc, «Detection and Classification of UAVs Using RF Fingerprints in the Presence of Wi-Fi and Bluetooth Interference», *IEEE Open J. Commun. Soc.*, vol. 1, pp. 60-76, 2020, doi: 10.1109/OJCOMS.2019.2955889.
- [31] A. Lindner, T. Müller, L. W. Sommer, T. Emter, A. Zube, y A. Kröker, «MODEAS – a Counter-UAV System with Optical and Radar Sensors», 16 de enero de 2024, *Engineering*. doi: 10.20944/preprints202401.1249.v1.
- [32] D. Tejera Berengué, «Análisis de señales acústicas para la detección de UAVs», Universidad de Alcalá, 2021. Accedido: 28 de enero de 2025. [En línea]. Disponible en: <https://ebuah.uah.es/dspace/handle/10017/49529>
- [33] V. U. Castrillo, A. Manco, D. Pascarella, y G. Gigante, «A Review of Counter-UAS Technologies for Cooperative Defensive Teams of Drones», *Drones*, vol. 6, n.º 3, p. 65, mar. 2022, doi: 10.3390/drones6030065.
- [34] J. Najafi, S. Mirzakuchaki, y S. Shamaghdari, «Autonomous Drone Detection and Classification Using Computer Vision and Prony Algorithm-Based Frequency Feature Extraction», *J. Intell. Robot. Syst.*, vol. 111, n.º 1, p. 8, ene. 2025, doi: 10.1007/s10846-024-02216-x.
- [35] C. Stegmayer, «¿Qué es un micrófono MEMS? (Sistemas micro electromecánicos)». Accedido: 1 de febrero de 2025. [En línea]. Disponible en: <https://microfono.rocks/que-es-un-microfono-mems-sistemas-micro-electromecanicos/>

- [36] «What is the difference between STFT and FFT?», Quora. Accedido: 2 de febrero de 2025. [En línea]. Disponible en: <https://www.quora.com/What-is-the-difference-between-STFT-and-FFT>
- [37] S. Park *et al.*, «Combination of radar and audio sensors for identification of rotor-type Unmanned Aerial Vehicles (UAVs)», en *2015 IEEE SENSORS*, Busan: IEEE, nov. 2015, pp. 1-4. doi: 10.1109/ICSENS.2015.7370533.
- [38] G. A. Martínez Mascorro y G. Aguilar Torres, «Reconocimiento de voz basado en MFCC, SBC y Espectrogramas», *Ingenius*, n.º 10, dic. 2013, doi: 10.17163/ings.n10.2013.02.
- [39] Z. Shi, X. Chang, C. Yang, Z. Wu, y J. Wu, «An Acoustic-Based Surveillance System for Amateur Drones Detection and Localization», *IEEE Trans. Veh. Technol.*, vol. 69, n.º 3, pp. 2731-2739, mar. 2020, doi: 10.1109/TVT.2020.2964110.
- [40] L. Shi, I. Ahmad, Y. He, y K. Chang, «Hidden Markov model based drone sound recognition using MFCC technique in practical noisy environments», *J. Commun. Netw.*, vol. 20, n.º 5, pp. 509-518, oct. 2018, doi: 10.1109/JCN.2018.000075.
- [41] X. Zhuang, J. Huang, G. Potamianos, y M. Hasegawa-Johnson, «Acoustic fall detection using Gaussian mixture models and GMM supervectors», en *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, Taipei, Taiwan: IEEE, abr. 2009, pp. 69-72. doi: 10.1109/ICASSP.2009.4959522.
- [42] C. A. Ahmed, F. Batool, W. Haider, M. Asad, y S. H. Raza Hamdani, «Acoustic Based Drone Detection Via Machine Learning», en *2022 International Conference on IT and Industrial Technologies (ICIT)*, oct. 2022, pp. 01-06. doi: 10.1109/ICIT56493.2022.9989229.
- [43] «¿Qué es el deep learning? | IBM». Accedido: 30 de enero de 2025. [En línea]. Disponible en: <https://www.ibm.com/es-es/topics/deep-learning>
- [44] S. Al-Emadi, A. Al-Ali, A. Mohammad, y A. Al-Ali, «Audio Based Drone Detection and Identification using Deep Learning», en *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, Tangier, Morocco: IEEE, jun. 2019, pp. 459-464. doi: 10.1109/IWCMC.2019.8766732.
- [45] A. Ali, Y. K. Jadoon, S. A. Changazi, y M. Qasim, «Military Operations: Wireless Sensor Networks based Applications to Reinforce Future Battlefield Command System», en *2020 IEEE 23rd International Multitopic Conference (INMIC)*, nov. 2020, pp. 1-6. doi: 10.1109/INMIC50486.2020.9318168.
- [46] C. Leeser, «Sensor Network Assembly», US 2022/0417327 A1 Accedido: 1 de febrero de 2025. [En línea]. Disponible en: <https://ppubs.uspto.gov/pubwebapp/external.html>
- [47] J. Weaver, «Wired vs. wireless networks: Challenges and advantages of cellular routers for 5G networking», Ericsson. Accedido: 13 de febrero de 2025. [En línea]. Disponible en: <https://cradlepoint.com/resources/blog/wired-vs-wireless-networks-challenges-and-advantages-of-cellular-routers-for-5g-networking/>
- [48] N. Lipka, «What is drone jamming and how can it be countered?», Doodle Labs. Accedido: 13 de febrero de 2025. [En línea]. Disponible en: <https://doodlelabs.com/what-is-drone-jamming/>
- [49] S. J. Danbatta y A. Varol, «Comparison of Zigbee, Z-Wave, Wi-Fi, and Bluetooth Wireless Technologies Used in Home Automation», en *2019 7th International Symposium on Digital*

- Forensics and Security (ISDFS)*, Barcelos, Portugal: IEEE, jun. 2019, pp. 1-5. doi: 10.1109/ISDFS.2019.8757472.
- [50] R. Mubashar, M. A. B. Siddique, A. U. Rehman, A. Asad, y A. Rasool, «Comparative performance analysis of short-range wireless protocols for wireless personal area network», *Iran J. Comput. Sci.*, vol. 4, n.º 3, pp. 201-210, sep. 2021, doi: 10.1007/s42044-021-00087-1.
- [51] «Long-Range Backscatter». Accedido: 3 de febrero de 2025. [En línea]. Disponible en: <https://longrange.cs.washington.edu/>
- [52] «Router Mikrotik RB960PGS 5x RJ-45 10/100/1000 Mb/s 1x SFP Port», © ESUS IT 2009 - 2025. Accedido: 28 de febrero de 2025. [En línea]. Disponible en: <https://www.esus-it.es/product-spa-42356-Router-Mikrotik-RB960PGS-5x-RJ-45-10-100-1000-Mb-s-1x-SFP-Port.html>
- [53] «hEX PoE | MikroTik». Accedido: 1 de marzo de 2025. [En línea]. Disponible en: <https://mikrotik.com/product/RB960PGS>
- [54] «Manual de instrucciones del micrófono de condensador USB TONOR TC30», Manuals+. Accedido: 5 de marzo de 2025. [En línea]. Disponible en: <https://manuals.plus/es/tonor/tc30-usb-condenser-microphone-manual>
- [55] «Buy a Raspberry Pi 3 Model B», Raspberry Pi. Accedido: 28 de febrero de 2025. [En línea]. Disponible en: <https://www.raspberrypi.com/products/raspberry-pi-3-model-b/>
- [56] «Dron profesional E88 4K gran angular, cámara HD 1080P, WiFi, FPV, Soporte de altura, plegable, RC, cuadricóptero, juguetes para niños - AliExpress 44», aliexpress. Accedido: 28 de febrero de 2025. [En línea]. Disponible en: [//es.aliexpress.com/item/1005006749040149.html](https://es.aliexpress.com/item/1005006749040149.html)
- [57] «Download Python», Python.org. Accedido: 11 de marzo de 2025. [En línea]. Disponible en: <https://www.python.org/downloads/>
- [58] «Download Visual Studio Code - Mac, Linux, Windows». Accedido: 11 de marzo de 2025. [En línea]. Disponible en: <https://code.visualstudio.com/Download>
- [59] «VEEKTOMX Mini batería externa de 10000 mAh (10000 mAh gris): Amazon.es: Electrónica». Accedido: 14 de marzo de 2025. [En línea]. Disponible en: <https://www.amazon.es/VEEKTOMX-Mini-bater%C3%ADa-externa-10000/dp/B09CZ6TY7F>
- [60] «Logitech C615 Webcam Full HD 1080p | PcComponentes.com», PcComponentes. Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://www.pccomponentes.com/logitech-c615-webcam-full-hd-1080p>

ANEXO I: IMPLICACIONES SOCIALES, Y/O ECONÓMICAS, Y/O AMBIENTALES

El avance tecnológico en los vehículos aéreos no tripulados ha transformado radicalmente diversos ámbitos de la sociedad, desde la logística hasta la vigilancia. Sin embargo, su creciente accesibilidad también ha abierto la puerta a usos ilícitos que comprometen la seguridad en espacios urbanos y en zonas estratégicas. En este contexto, la detección eficiente de drones no es solo una necesidad técnica, sino una cuestión de interés social de primer orden.

Uno de los ámbitos donde esta tecnología cobra especial relevancia es la seguridad y defensa. En entornos urbanos, donde los drones pueden operar con discreción y maniobrabilidad, su detección se vuelve esencial para prevenir incursiones no autorizadas en instalaciones sensibles, como infraestructuras gubernamentales, centrales energéticas o aeropuertos. En el ámbito militar, la amenaza que representan estos dispositivos es aún más crítica, ya que pueden ser empleados para espionaje, interferencias electrónicas o incluso ataques directos en escenarios de conflicto. La capacidad de identificarlos con rapidez es clave para evitar riesgos mayores y responder de manera eficaz.

Más allá de la defensa convencional, el uso de drones con fines terroristas o delictivos ha aumentado la urgencia de implementar sistemas de detección en entornos civiles. Casos documentados han demostrado que estos dispositivos pueden ser utilizados para el transporte de explosivos, la interrupción de eventos multitudinarios o el contrabando en fronteras y centros penitenciarios. Frente a estas amenazas, contar con una tecnología accesible y fiable para su detección se convierte en un elemento disuasorio que contribuye a la seguridad colectiva.

Desde una perspectiva económica, una de las grandes ventajas de los sistemas acústicos de detección radica en su bajo coste en comparación con otras soluciones tecnológicas, como los radares o los sistemas ópticos avanzados. Mientras que estos últimos requieren infraestructuras costosas y dependen de condiciones ambientales favorables, la detección acústica permite una implementación más sencilla y adaptable a distintos entornos. Esta accesibilidad favorece su despliegue en infraestructuras críticas y espacios públicos sin generar una inversión prohibitiva, lo que amplía su aplicabilidad en el ámbito civil y de seguridad pública.

En definitiva, la detección de drones no solo representa un desafío tecnológico, sino una necesidad social ante el creciente uso de estos dispositivos con fines ilícitos. La capacidad de identificar y neutralizar amenazas en tiempo real es un factor clave para garantizar la seguridad en entornos urbanos, prevenir actos hostiles y reforzar la protección de infraestructuras críticas. A medida que la presencia de drones siga expandiéndose, contar con sistemas de detección efectivos no será una opción, sino un pilar fundamental en las estrategias de seguridad del futuro.

ANEXO II: REFLEXIONES ÉTICAS Y SOCIALES

El desarrollo de tecnologías de detección de drones plantea interrogantes que van más allá de su viabilidad técnica o su impacto en la seguridad. Su implementación en entornos urbanos y en infraestructuras críticas implica dilemas éticos y sociales que deben abordarse con responsabilidad. La clave reside en encontrar un equilibrio entre protección y privacidad, entre la necesidad de vigilar y el derecho a no ser vigilado.

Uno de los principales puntos de debate es el uso de tecnologías de detección en espacios públicos. En el contexto de la seguridad, identificar drones potencialmente peligrosos es una prioridad, pero esto no debe convertirse en una justificación para la vigilancia indiscriminada. La posibilidad de registrar y analizar sonidos en determinadas áreas genera preguntas sobre la privacidad de las personas y el uso que podría darse a los datos capturados. Si bien el objetivo principal de estos sistemas es identificar drones, su implementación sin regulaciones adecuadas podría derivar en la monitorización no autorizada de otros sonidos ambientales o incluso conversaciones, lo que supondría una invasión de la privacidad. Este riesgo se amplía cuando consideramos la posibilidad de que la detección de drones se utilice con fines distintos a los previstos. Sin una regulación clara, lo que comienza como una medida de seguridad podría transformarse en una herramienta de control excesivo. Por ejemplo, un sistema diseñado para detectar incursiones no autorizadas en infraestructuras críticas podría acabar utilizándose para monitorear indiscriminadamente ciertas áreas urbanas o restringir el uso legítimo de drones en actividades científicas, comerciales o recreativas. Incluso en escenarios más preocupantes, esta tecnología podría ser empleada para identificar y rastrear el movimiento de personas en manifestaciones o eventos públicos, afectando derechos fundamentales como la libertad de reunión y la privacidad.

Por ello, es crucial que la detección de drones esté respaldada por marcos normativos claros, que definan con precisión sus alcances y limitaciones. La transparencia en su implementación es clave para evitar el riesgo de que estas tecnologías terminen sirviendo intereses ajenos a la seguridad colectiva. Un despliegue adecuado debe garantizar que su uso se mantenga en el propósito para el que fue diseñado y que no genere una sensación de vigilancia constante en la sociedad.

Desde una perspectiva más amplia, la proliferación de sistemas de detección también invita a reflexionar sobre el impacto de la tecnología en la percepción del espacio público. La presencia de mecanismos de vigilancia —incluso aquellos destinados a la seguridad— puede modificar la manera en que las personas se relacionan con su entorno. Es importante evitar que el desarrollo de estos sistemas genere un clima de sospecha generalizada, en el que cualquier uso de drones sea visto como una amenaza, restringiendo su potencial en ámbitos como la investigación científica, la filmación o la logística.

Finalmente, en términos éticos, la detección de drones se sitúa en un punto intermedio entre prevención y respuesta. Su desarrollo debe orientarse a minimizar amenazas sin comprometer libertades individuales ni fomentar un uso desproporcionado de la tecnología. La clave no está solo en el avance técnico, sino en la manera en que se decide integrarlo en la sociedad. Como cualquier otra herramienta de seguridad, su valor dependerá de cómo se utilice y de las garantías que se establezcan para evitar abusos.

En última instancia, la pregunta que queda abierta es cómo garantizar que estos sistemas cumplan su función sin generar efectos no deseados en el tejido social. La respuesta no radica solo en la tecnología, sino en la manera en que decidimos integrarla en nuestro día a día.

ANEXO III: CÓDIGO DE ENVÍO DE AUDIO DESDE RASPBERRY PI

```
import pyaudio #Maneja reproducción de audio
import socket #Permite recibir audio enviado por la Raspberry mediante conexión
UDP

# Configuración del audio
FORMAT = pyaudio.paInt16 # Formato de audio en 16 bits
CHANNELS = 1 # Se usa un solo canal (mono)
RATE = 44100 # Frecuencia de muestreo en 44.1 kHz (estándar en audio)
CHUNK = 1024 # Tamaño del buffer, cantidad de muestras por lectura

# Configuración del socket UDP
SERVER_IP = "192.168.88.247" # Dirección IP del ordenador receptor en la red
PORT = 5000 # Puerto de comunicación para la transmisión de audio

# Inicializamos PyAudio
# Creamos una instancia de PyAudio para manejar la captura de sonido
audio = pyaudio.PyAudio()
stream = audio.open(format=FORMAT, channels=CHANNELS, rate=RATE,
                    input=True, frames_per_buffer=CHUNK) # Se abre el flujo de
audio con los parámetros definidos

# Creamos el socket UDP
# Definimos un socket para enviar los datos de audio usando el protocolo UDP
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

print("\n🔊🔊🔊 Transmitiendo audio... Presiona Ctrl+C para detener.")

# Bucle de transmisión continua
try:
    while True:
        data = stream.read(CHUNK) # Lee datos del micrófono en bloques de tamaño
        sock.sendto(data, (SERVER_IP, PORT)) # Envía datos al ordenador receptor
        por la red
except KeyboardInterrupt:
    print("\n🔊🔊🔊 Deteniendo transmisión...")

    # Cerramos los recursos en caso de interrupción
    stream.stop_stream() # Detiene el flujo de audio
    stream.close() # Cierra el flujo de audio
    audio.terminate() # Libera los recursos de PyAudio
    sock.close() # Cierra el socket de transmisión
```

ANEXO IV: CÓDIGO DE RECEPCIÓN DE AUDIO EN TIEMPO REAL

```
# =====
# Este script recibe y guarda audio en tiempo real desde múltiples Raspberry Pi.
#   - Se configuran múltiples hilos para recibir audio de diferentes puertos.
#   - Se almacena cada audio recibido en archivos .wav organizados por fecha y
#     hora.
#   - Cada sesión dura 30 minutos, tras lo cual el operario decide si continuar o
#     salir.
# =====

import pyaudio # Manejo del audio y captura de sonido
import socket # Comunicación en red mediante UDP
import threading # Uso de hilos para procesar múltiples flujos de audio en
paralelo
import wave # Guardado de archivos de audio en formato WAV
import time # Gestión de tiempos y ventanas de almacenamiento
import os # Manejo del sistema de archivos y creación de carpetas
import signal # Captura de señales del sistema como Ctrl+C
import sys # Gestión de la salida del programa
import logging # Manejo de logs en lugar de prints para mejorar estética

# Configuración del logging
logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)s -
%(message)s')

# Configuración del audio
FORMAT = pyaudio.paInt16 # Formato de audio en 16 bits
CHANNELS = 1 # Usa un solo canal (mono)
RATE = 44100 # Frecuencia de muestreo en 44.1 kHz
CHUNK = 1024 # Tamaño de cada fragmento de audio
WINDOW_TIME = 5 # Guarda un archivo cada 5 segundos
SESSION_DURATION = 1800 # Duración de la sesión en segundos (30 minutos)

# Configuración de los puertos y dispositivos
ports = {5000: "RaspberryPi1", 5001: "RaspberryPi2"} # Diccionario de Raspberry
Pi y sus puertos

# Carpeta base para guardar los archivos de audio, debe ser creada manualmente por
el operario
output_folder = "Audio Recibido/12-03-2025" # Ruta donde se almacenarán los
archivos de audio

# Variables de control
active_streams = {port: False for port in ports} # Indica si hay transmisión en
cada puerto
sockets = {} # Diccionario global para almacenar los sockets abiertos
running = True # Control para detener el script
threads = [] # Lista para manejar los hilos activos

# =====
# Función para recibir audio desde las Raspberry Pi en los distintos puertos.
# =====
def receive_audio(port, device_name):
    global running
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) # Crea un socket UDP
    sockets[port] = sock # Guarda el socket en el diccionario global
    sock.bind(("", port)) # Asigna el puerto a la escucha
    sock.settimeout(1) # Define un tiempo de espera para evitar bloqueos
```

```

logging.info(f"⌚ Esperando transmisión desde {device_name} en el puerto
{port}...")
started_transmitting = False # Indica si el dispositivo ha comenzado a
transmitir

frames = [] # Almacena fragmentos de audio
start_time = time.time() # Registra el tiempo de inicio

try:
    while running and time.time() - start_time < SESSION_DURATION:
        try:
            data, addr = sock.recvfrom(CHUNK * 2) # Recibe los datos del
socket
            active_streams[port] = True # Marca la transmisión como activa
            if not started_transmitting:
                logging.info(f"⌚ {device_name} ha comenzado a transmitir en
el puerto {port}.")
                started_transmitting = True
                frames.append(data) # Añade los datos recibidos a la lista de
frames

            # Guarda el archivo cada WINDOW_TIME segundos
            if time.time() - start_time >= WINDOW_TIME:
                timestamp = time.strftime("%H-%M-%S") # Genera el timestamp
del archivo
                filename = os.path.join(output_folder,
f"{device_name}_{timestamp}.wav")

                with wave.open(filename, 'wb') as wf:
                    wf.setnchannels(CHANNELS)
                    wf.setsampwidth(pyaudio.PyAudio().get_sample_size(FORMAT))
                    wf.setframerate(RATE)
                    wf.writeframes(b''.join(frames)) # Escribe los datos en
el archivo

                logging.info(f"💾 Archivo guardado: {filename}")

                frames = [] # Reinicia la lista de frames
                start_time = time.time() # Reinicia el contador de tiempo
        except socket.timeout:
            if active_streams[port]:
                logging.warning(f"⚠ No se detecta transmisión de
{device_name} en {port}.")
                started_transmitting = False
                active_streams[port] = False
    except Exception as e:
        logging.error(f"✖ Error en la recepción de {device_name}: {e}")
    finally:
        try:
            sock.close()
            logging.info(f"✅ Socket en {device_name} cerrado.")
        except Exception as e:
            logging.warning(f"⚠ Error al cerrar socket en {device_name}: {e}")

# =====
# Inicio de los hilos de recepción de audio.
# =====
for port, name in ports.items():
    thread = threading.Thread(target=receive_audio, args=(port, name),
daemon=True)
    threads.append(thread)
    thread.start()

```

```
logging.info("⌚ Sistema de recepción en tiempo real activo. Se  
detendrá en 30 minutos.")  
time.sleep(SESSION_DURATION) # Espera el tiempo de la sesión  
  
# =====  
# Finalización de la sesión y opción de reinicio.  
# =====  
  
logging.info("⌚ Sesión finalizada. ¿Desea continuar? (Sí = Reiniciar, No =  
Salir)")  
user_input = input("Presiona 's' para continuar o cualquier otra tecla para salir:  
\n").strip().lower()  
if user_input == 's':  
    logging.info("🔄 Reiniciando la sesión...")  
    os.execv(sys.executable, ['python'] + sys.argv) # Reinicia el script  
else:  
    logging.info("✅ Finalizando el programa.")  
    sys.exit(0) # Sale del programa
```

ANEXO V: CÓDIGO DE GENERACIÓN DE ESPECTROGRAMAS

```

import numpy as np #Para manejar arrays numéricos (convierte audio en datos
numéricos)
import matplotlib.pyplot as plt #Para realizar la gráfica del espectrograma
import wave #Para leer archivos .wav
import os #Para manejar directorios y rutas
from scipy.signal import spectrogram #Función de SciPy que calcula el
espectrograma del audio

# Ruta del archivo de audio en la carpeta "Audios" dentro de "PC"
audio_folder = os.path.join(os.getcwd(), "AudiosDron1") # Ruta absoluta de la
carpeta
filename = "Dron 1,5 m.wav" # Nombre del archivo de audio
file_path = os.path.join(audio_folder, filename) # Ruta completa del archivo

# Cargamos el archivo de audio
with wave.open(file_path, "rb") as wav_file: #nombre del archivo y lectura
binaria. With asegura que el archivo se cierre después de usarse
    fs = wav_file.getframerate() # Obtiene la frecuencia de muestreo (número de
muestras por segundo)
    frames = wav_file.readframes(-1) # Lee todos los frames del archivo de audio
    audio_data = np.frombuffer(frames, dtype=np.int16) # Convierte datos binarios
en un array de enteros de 16 bits

# Generamos el espectrograma
# El espectrograma muestra cómo varían las frecuencias del audio a lo largo del
tiempo
f, t, Sxx = spectrogram(audio_data, fs) #Calcula el espectrograma del audio
#audio_data: Datos del audio en forma de array
#fs: frecuencia de muestreo del audio (Número de muestras a tomar por segundo)
#Lo devuelve como: f frecuencias en Hz, t Tiempo en segundos y Sxx intensidad de
la señal en cada frecuencia y campo

# Generamos la gráfica del espectrograma
plt.figure(figsize=(10, 6)) # Crea una nueva figura con tamaño 10x6 pulgadas
plt.pcolormesh(t, f, 10 * np.log10(Sxx), shading='auto') # Representa el
espectrograma con colores y convierte la potencia en decibelios (dB). Shading
suaviza la interpolación de los valores
plt.ylabel('Frecuencia [Hz]') # Etiqueta del eje Y (frecuencia en Hz)
plt.xlabel('Tiempo [s]') # Etiqueta del eje X (tiempo en segundos)
plt.title(f'Espectrograma {os.path.splitext(filename)[0]}') # Título del gráfico
con el nombre del archivo sin la extensión
plt.colorbar(label='dB') # Agrega barra de colores indicando la escala en
decibeles (dB)

# Guardamos el espectrograma en una carpeta aparte
output_folder = os.path.join(os.getcwd(), "Espectrogramas varios") # Ruta
absoluta de la carpeta "Espectrogramas"
output_path = os.path.join(output_folder, f"{os.path.splitext(filename)[0]}.png")
# Ruta del archivo de salida
plt.savefig(output_path) # Guarda la imagen del espectrograma

plt.show() # Muestra la gráfica

print(f" Espectrograma filtrado guardado en: {output_path}")

```

ANEXO VI: CÓDIGO DE ANÁLISIS FRECUENCIAL

Este script analiza la distribución de frecuencias dominantes en un instante específico de un audio de dron.
Primero, leemos el archivo de audio y extraemos un fragmento corto (100 ms) para analizarlo.
Luego, aplicamos la Transformada de Fourier (FFT) para obtener la distribución de frecuencias e intensidades.
Finalmente, generamos una gráfica con la frecuencia en el eje X y la intensidad en el eje Y y guardamos la imagen en la carpeta "Análisis frecuencia".

```
import numpy as np
import scipy.io.wavfile as wav #Para leer archivos .wav teniendo en cuenta un
análisis complejo de datos
import matplotlib.pyplot as plt #Para hacer gráfica
import os

# Definimos la carpeta donde están los audios y la carpeta de salida de imágenes
audio_folder = os.path.join(os.getcwd(), "AudiosDron1") # Carpeta de entrada con
los audios a analizar
output_folder = os.path.join(os.getcwd(), "Frecuencia-Intensidad") # Carpeta
donde guardaremos los resultados
os.makedirs(output_folder, exist_ok=True) # Creamos la carpeta si no existe

# Especificamos el archivo de audio a analizar
filename = "Dron 1,5 m.wav" # Podemos cambiarlo por cualquier otro archivo en la
carpeta "Audios"
file_path = os.path.join(audio_folder, filename)

# Leemos el archivo de audio
fs, audio = wav.read(file_path)

# Si el audio es estéreo, lo convertimos a mono para facilitar el análisis
if len(audio.shape) > 1:
    audio = np.mean(audio, axis=1)

# Seleccionamos un fragmento del audio para analizar (100 ms en el centro del
archivo)
duracion_audio = len(audio) / fs # Calculamos la duración total en segundos
tiempo_inicial = duracion_audio / 2 # Escogemos el punto medio del audio como
referencia
muestras_inicial = int(tiempo_inicial * fs) # Convertimos el tiempo a muestras
muestras_final = muestras_inicial + int(0.1 * fs) # Tomamos 100 ms de audio
segmento = audio[muestras_inicial:muestras_final] # Extraemos el segmento
seleccionado

# Aplicamos la FFT para obtener el espectro de frecuencias de este fragmento
frecuencias = np.fft.fftfreq(len(segmento), d=1/fs) # Calculamos la escala de
frecuencias
espectro = np.abs(np.fft.fft(segmento)) # Calculamos la magnitud del espectro

# Tomamos solo la mitad positiva del espectro, ya que es simétrico
frecuencias = frecuencias[:len(frecuencias)//2]
espectro = espectro[:len(espectro)//2]

# Generamos la gráfica de la distribución de frecuencias
plt.figure(figsize=(8, 5))
plt.plot(frecuencias, espectro, color='b', alpha=0.7)
plt.xlabel("Frecuencia (Hz)")
plt.ylabel("Intensidad")
plt.title(f"Distribución de frecuencias ({filename})")
plt.grid()
```

```
# Guardamos la imagen en la carpeta "Análisis frecuencia" con el mismo nombre del
archivo de audio
output_path = os.path.join(output_folder, filename.replace(".wav", ".png"))
plt.savefig(output_path, dpi=300)

plt.show()

print(f"Análisis de frecuencia guardado en: {output_path}")
```

ANEXO VII: CÓDIGO DE GENERACIÓN DE BOXPLOT

```
# Este script genera un box-plot de las frecuencias predominantes en los audios
analizados.
# Nos permitirá identificar el rango de frecuencias más recurrente y definir un
filtro paso-banda adecuado.

import numpy as np
import scipy.io.wavfile as wav
import scipy.signal as signal
import matplotlib.pyplot as plt
import os

# Definimos la carpeta donde se encuentran los audios
audio_folder = os.path.join(os.getcwd(), "AudiosDron1")

# Indicamos la carpeta donde se guardará el box-plot
output_folder = os.path.join(os.getcwd(), "Frecuencia-Intensidad")

# Lista para almacenar los valores de frecuencia predominante por archivo
frecuencias_por_audio = []
nombres_archivos = []

# Procesamos cada archivo de audio en la carpeta
for filename in os.listdir(audio_folder):
    if filename.endswith(".wav"):
        file_path = os.path.join(audio_folder, filename)

        # Leemos el archivo de audio
        fs, audio = wav.read(file_path)

        # Convertimos a mono si es estéreo
        if len(audio.shape) > 1:
            audio = np.mean(audio, axis=1)

        # Aplicamos la FFT para obtener el espectro de frecuencia
        espectro = np.fft.fft(audio)
        espectro_mag = np.abs(espectro[:len(espectro) // 2]) # Nos quedamos solo
con la mitad positiva
        frecuencias = np.fft.fftfreq(len(audio), d=1/fs)[:len(espectro) // 2] #
Calculamos las frecuencias correspondientes

        # Seleccionamos las frecuencias con mayor intensidad
        umbral_intensidad = np.percentile(espectro_mag, 90) # Tomamos las
frecuencias más representativas (top 10%)
        frecuencias_dominantes = frecuencias[espectro_mag >= umbral_intensidad]

        # Guardamos las frecuencias dominantes y el nombre del archivo
        frecuencias_por_audio.append(frecuencias_dominantes)
        nombres_archivos.append(filename.replace(".wav", ""))

# Creamos el box-plot
plt.figure(figsize=(12, 6))
plt.boxplot(frecuencias_por_audio, labels=nombres_archivos, showfliers=False)
plt.xlabel("Pruebas de audio")
plt.ylabel("Frecuencia (Hz)")
plt.title("Distribución de frecuencias predominantes en los audios de dron")
plt.xticks(rotation=45)
plt.grid()

# Guardamos la imagen
output_path = os.path.join(output_folder, "Boxplot de frecuencias.png")
```

```
plt.savefig(output_path, bbox_inches="tight")

# Mostramos la imagen
plt.show()

print(f"Box-plot de frecuencias generado y guardado en: {output_path}")
```

ANEXO VIII: CÓDIGO DE FILTRO PASO-BANDA Y DEFINICIÓN DE UMBRAL

```
# En este script vamos a calcular el umbral de detección de un dron basado en su
# energía en decibelios (dB).
# Primero, aplicamos un filtro paso-banda (3000-6000 Hz) para centrarnos en las
# frecuencias características del dron.
# Luego, obtenemos los valores en dB de los espectrogramas filtrados y calculamos
# una media para determinar un umbral de detección.
# Este umbral servirá para comparar con nuevos audios y decidir si contienen el
# sonido del dron o no.
import numpy as np
import matplotlib.pyplot as plt
import os
import wave
from scipy.signal import butter, filtfilt, spectrogram
#butter: Para definir el filtro paso banda
#Filtfilt: Para pasar el filtro dos veces (sin distorsión de fase)

# =====
# FUNCIÓN PARA FILTRO PASO-BANDA
# =====
def butter_bandpass(lowcut, highcut, fs, order=5):
    #Crea un filtro paso-banda Butterworth.
    #Parámetros:
    #- lowcut: Frecuencia baja de corte en Hz.
    #- highcut: Frecuencia alta de corte en Hz.
    #- fs: Frecuencia de muestreo en Hz.
    #- order: Orden del filtro (define la atenuación fuera del rango, mayor rango
    mayor restricción de filtro).
    nyq = 0.5 * fs # Frecuencia de Nyquist (mitad de la frecuencia de muestreo)
    low = lowcut / nyq # Normalización de la frecuencia baja
    high = highcut / nyq # Normalización de la frecuencia alta
    b, a = butter(order, [low, high], btype='band') # Diseño del filtro
    return b, a

def bandpass_filter(data, lowcut, highcut, fs, order=5): #Aplica el filtro paso-
banda a la señal de audio.
    b, a = butter_bandpass(lowcut, highcut, fs, order=order)
    return filtfilt(b, a, data) # Aplicación del filtro sin distorsión de fase

# =====
# CONFIGURACIÓN GENERAL
# =====
# Definimos las carpetas de entrada y salida
audio_folder = "AudiosDron1" # Carpeta con audios del dron
output_folder = "Pruebas" # Carpeta donde se guardarán los espectrogramas
filtrados

# Frecuencias de corte del filtro paso-banda (definidas con el análisis previo)
low_freq = 3000 # Frecuencia mínima en Hz
high_freq = 5000 # Frecuencia máxima en Hz

# Lista para almacenar valores de energía (en dB) en la banda de interés
energy_values = []

audio_files = [f for f in os.listdir(audio_folder) if f.endswith('.wav')]

# =====
# PROCESAMOS CADA AUDIO
# =====
```

```

for audio_filename in audio_files:
    audio_path = os.path.join(audio_folder, audio_filename)

    # Cargamos el archivo de audio
    with wave.open(audio_path, "rb") as wav_file:
        fs = wav_file.getframerate() # Frecuencia de muestreo
        frames = wav_file.readframes(-1) # Leemos los frames del audio
        audio_data = np.frombuffer(frames, dtype=np.int16) # Convertimos en array
numpy

    # Aplicamos el filtro paso-banda
    filtered_audio = bandpass_filter(audio_data, low_freq, high_freq, fs)

    # Generamos el espectrograma del audio filtrado
    f, t, Sxx = spectrogram(filtered_audio, fs)
    Sxx_dB = 10 * np.log10(Sxx + 1e-10) # Convertimos a dB, evitando log(0)

    # Guardamos el espectrograma filtrado
    plt.figure(figsize=(10, 6))
    plt.pcolormesh(t, f, Sxx_dB, shading='auto')
    plt.ylabel('Frecuencia [Hz]')
    plt.xlabel('Tiempo [s]')
    plt.title(f'Espectrograma Filtrado (3000-6000 Hz) - {audio_filename}')
    plt.colorbar(label='dB')

    output_path = os.path.join(output_folder,
f"{os.path.splitext(audio_filename)[0]}_filtrado.png")
    plt.savefig(output_path)
    plt.close()

    # Extraemos los valores en la banda de interés (3000-6000 Hz)
    freq_low_idx = np.searchsorted(f, low_freq)
    freq_high_idx = np.searchsorted(f, high_freq)
    energy_band = Sxx_dB[freq_low_idx:freq_high_idx, :]
    avg_energy = np.mean(energy_band) # Promedio de dB en la banda de frecuencias
seleccionada
    energy_values.append(avg_energy)

# =====
# CÁLCULO DEL UMBRAL DE DETECCIÓN
# =====
mean_energy = np.mean(energy_values) # Media de dBs
std_energy = np.std(energy_values) # Desviación estándar

k = 1.5 # Factor de ajuste del umbral (se puede modificar para más o menos
sensibilidad)
threshold_low = mean_energy - k * std_energy # Umbral inferior
threshold_high = mean_energy + k * std_energy # Umbral superior

print("\n=====")
print(" UMBRAL DE DETECCIÓN (dB)")
print("=====")
print(f"Umbral inferior: {threshold_low:.2f} dB")
print(f"Umbral superior: {threshold_high:.2f} dB")
print("=====")

# Guardamos el umbral en un archivo para futuras detecciones
umbral_file = os.path.join(output_folder, "umbral_deteccion2.txt")
with open(umbral_file, "w") as f:
    f.write(f"Umbral inferior: {threshold_low:.2f} dB\n")
    f.write(f"Umbral superior: {threshold_high:.2f} dB\n")

print(f"☑ Umbral guardado en: {umbral_file}")

```

ANEXO IX: CÓDIGO DE PRUEBAS DEL DRON

```
# Script de detección de drones basado en el umbral de dB
# Cargamos el umbral desde un archivo .txt y verificamos si un audio contiene un
dron

import numpy as np
import matplotlib.pyplot as plt
import wave
import os
from scipy.signal import butter, filtfilt, spectrogram

# Definimos el filtro paso-banda
def butter_bandpass(lowcut, highcut, fs, order=5):
    nyq = 0.5 * fs # Frecuencia de Nyquist
    low = lowcut / nyq
    high = highcut / nyq
    b, a = butter(order, [low, high], btype='band')
    return b, a

# Aplicamos el filtro paso-banda
def bandpass_filter(data, lowcut, highcut, fs, order=5):
    b, a = butter_bandpass(lowcut, highcut, fs, order=order)
    return filtfilt(b, a, data) # Aplica el filtro sin desfase

# Cargamos el umbral de detección desde el archivo .txt que hemos creado antes
umbral_file = "umbral_deteccion.txt"
if os.path.exists(umbral_file):
    with open(umbral_file, "r") as file:
        umbral_values = file.readlines()
        umbral_low = float(umbral_values[0].split(":")[1].strip().replace(" dB",
"")) # Umbral inferior
        umbral_high = float(umbral_values[1].split(":")[1].strip().replace(" dB",
"")) # Umbral superior
else:
    raise FileNotFoundError("El archivo de umbral 'umbral_deteccion.txt' no fue
encontrado.")

# Carpeta donde está el audio de prueba
audio_folder = "AudiosDron1"
audio_filename = "Dron recorrido entero.wav" # Archivo de prueba
audio_path = os.path.join(audio_folder, audio_filename)

# Carga el archivo de audio
with wave.open(audio_path, "rb") as wav_file:
    fs = wav_file.getframerate() # Frecuencia de muestreo
    frames = wav_file.readframes(-1)
    audio_data = np.frombuffer(frames, dtype=np.int16)

# Definimos las frecuencias de corte para el filtro
low_freq = 3000
high_freq = 6000

# Aplicamos el filtro paso-banda al audio
filtered_audio = bandpass_filter(audio_data, low_freq, high_freq, fs)

# Generamos el espectrograma del audio filtrado
f, t, Sxx = spectrogram(filtered_audio, fs)

# Convertimos la energía del espectrograma a dB
Sxx_dB = 10 * np.log10(Sxx + 1e-10) # Se añade un pequeño valor para evitar
log(0)
```

```

# Calculamos la media de los dB en el rango de interés
freq_low_idx = np.searchsorted(f, low_freq)
freq_high_idx = np.searchsorted(f, high_freq)
dB_mean = np.mean(Sxx_dB[freq_low_idx:freq_high_idx, :])

# Verificamos si el audio está dentro del umbral de detección
deteccion = umbral_low <= dB_mean <= umbral_high

# Mostramos el resultado
print("\n=====")
print("          RESULTADO DE DETECCIÓN          ")
print("=====")
print(f"Nivel de dB medio en la banda {low_freq}-{high_freq} Hz: {dB_mean:.2f} dB")
print(f"Umbral inferior: {umbral_low:.2f} dB")
print(f"Umbral superior: {umbral_high:.2f} dB")
print("-----")
if deteccion:
    print("##8
t8
t8Posible DRON detectado en el audio.")
else:
    print("XXX No se ha detectado un dron en el audio.")
print("=====\\n")

# Guardamos y mostramos el espectrograma filtrado
output_folder = "Detecciones Pruebas"
if not os.path.exists(output_folder):
    os.makedirs(output_folder)

output_path = os.path.join(output_folder,
f"{os.path.splitext(audio_filename)[0]}_detected.png")

plt.figure(figsize=(10, 6))
plt.pcolormesh(t, f, Sxx_dB, shading='auto', cmap='viridis')
plt.ylabel('Frecuencia [Hz]')
plt.xlabel('Tiempo [s]')
plt.title(f"Espectrograma del audio analizado: {audio_filename}")
plt.colorbar(label='dB')
plt.savefig(output_path)
plt.show()

print(f"Espectrograma guardado en: {output_path}")

```

ANEXO X: CÓDIGO DE DETECCIÓN EN TIEMPO REAL

```
# =====
# Este script analiza el audio en tiempo real almacenado en la carpeta en la que
# se han guardado al mismo tiempo
#   - Se carga el umbral de detección hallado anteriormente
#   - Se define y se aplica el filtro al audio recibido
#   - Se genera un espectrograma del que se extrae energía y se calcula la media
#     en la banda de interés
#   - Se compara ese valor con el de umbral establecido y se comprueba si el
#     audio corresponde a un dron o no
#   - Se almacenan los resultados de la detección en otra carpeta
# =====

import numpy as np
import matplotlib.pyplot as plt
import os
import wave
import time
import logging
from datetime import datetime
from scipy.signal import butter, filtfilt, spectrogram

# =====
# CONFIGURACIÓN DEL LOGGING
# =====
log_folder = "Archivos diarios de detección" # Carpeta donde se guardan los logs
diarios
log_filename = os.path.join(log_folder, f"deteccion_{datetime.now().strftime('%Y-%m-%d')}.log")

logging.basicConfig(
    level=logging.INFO,
    format='%(asctime)s - %(levelname)s - %(message)s',
    handlers=[
        logging.FileHandler(log_filename, encoding="utf-8"), # Log diario con
codificación UTF-8
        logging.StreamHandler()
    ]
)

# =====
# FUNCIÓN PARA FILTRO PASA-BANDA (explicado anteriormente)
# =====
def butter_bandpass(lowcut, highcut, fs, order=5):
    nyq = 0.5 * fs
    low = lowcut / nyq
    high = highcut / nyq
    b, a = butter(order, [low, high], btype='band')
    return b, a

def bandpass_filter(data, lowcut, highcut, fs, order=5):
    b, a = butter_bandpass(lowcut, highcut, fs, order=order)
    return filtfilt(b, a, data)

# =====
# CARGA DEL UMBRAL DE DETECCIÓN DESDE ARCHIVO
# =====
umbral_file = "umbral_deteccion.txt"
with open(umbral_file, "r") as f:
    lines = f.readlines()
    umbral_inferior = float(lines[0].split(":")[1].strip().replace("dB", ""))
```

```

umbral_superior = float(lines[1].split(":")[1].strip().replace("dB", ""))

# =====
# CONFIGURACIÓN DEL PROCESO DE DETECCIÓN EN TIEMPO REAL
# =====
input_folder = "Audio Recibido/12-03-2025" # Carpeta donde se reciben los audios
output_folder = "Resultados Detección" # Carpeta donde se guardan los resultados

low_freq = 3000 # Frecuencia mínima del filtro pasa-banda
high_freq = 6000 # Frecuencia máxima del filtro pasa-banda

#Procesamiento de cada audio, aplicación del filtro y comparación con el umbral
def process_audio_file(file_path):
    with wave.open(file_path, "rb") as wav_file:
        fs = wav_file.getframerate()
        frames = wav_file.readframes(-1)
        audio_data = np.frombuffer(frames, dtype=np.int16)

    # Aplicamos el filtro pasa-banda
    filtered_audio = bandpass_filter(audio_data, low_freq, high_freq, fs)

    # Generamos el espectrograma del audio filtrado
    f, t, Sxx = spectrogram(filtered_audio, fs)
    Sxx_dB = 10 * np.log10(Sxx + 1e-10) # Convertimos a dB (evitando log(0))

    # Extraemos los valores de la banda de interés
    freq_low_idx = np.searchsorted(f, low_freq)
    freq_high_idx = np.searchsorted(f, high_freq)
    energy_band = Sxx_dB[freq_low_idx:freq_high_idx, :]
    avg_energy = np.mean(energy_band)

    # Comparación con el umbral
    if umbral_inferior <= avg_energy <= umbral_superior:
        logging.info(f"#{8}
#{8}
#{8}Posible DRON detectado en el audio {os.path.basename(file_path)}
({avg_energy:.2f} dB)")

    # Guardamos el resultado con el nombre del archivo, que ya incluye la hora
    result_path = os.path.join(output_folder, os.path.basename(file_path) +
".txt")
    with open(result_path, "w", encoding="utf-8") as res_file:
        res_file.write(f"Archivo: {os.path.basename(file_path)}\n")
        res_file.write(f"Energía promedio: {avg_energy:.2f} dB\n")
        res_file.write(f"Resultado: {'#{8}Dron detectado' if umbral_inferior <=
avg_energy <= umbral_superior else 'X No es un dron'}\n")

logging.info("°°°° Monitoreando archivos en tiempo real... (Presiona Ctrl+C para
detener)")

try:
    while True:
        # Buscamos nuevos archivos en la carpeta de entrada
        audio_files = [f for f in os.listdir(input_folder) if f.endswith('.wav')]

        if not audio_files:
            logging.warning("⚠ No se encontraron archivos para analizar. Verifica
si el sistema de recepción está funcionando.")
        else:
            for audio_file in audio_files:
                file_path = os.path.join(input_folder, audio_file)

```

```
        process_audio_file(file_path)
    time.sleep(5) # ⌚ Ajustamos el tiempo de espera a 5 segundos
except KeyboardInterrupt:
    logging.info("🛑🛑🛑Detección detenida por el usuario.")
```

ANEXO XI: CÓDIGO DE CREACIÓN DE RUIDO

```
# =====
# PRUEBA DE ELIMINACIÓN DEL UMBRAL SUPERIOR
# =====
# En este script generamos ruido blanco y lo filtramos en la banda de 3000-6000
# Hz,
# la misma que utiliza el sistema de detección de drones. El objetivo es demostrar
# cómo la eliminación del umbral superior podría generar falsas detecciones,
# activando alertas constantemente en presencia de ruido ambiental.
# =====
import numpy as np
import soundfile as sf
import os
from scipy.signal import butter, filtfilt

# =====
# PARÁMETROS DEL RUIDO
# =====
fs = 44100 # Frecuencia de muestreo estándar en Hz
duration = 50 # Duración total en segundos (10 audios de 5 segundos cada uno)

# Generamos ruido blanco aleatorio
noise = np.random.normal(0, 1, fs * duration)

# =====
# DEFINICIÓN DEL FILTRO PASA-BANDA
# =====
def butter_bandpass(lowcut, highcut, fs, order=5):
    nyq = 0.5 * fs
    low = lowcut / nyq
    high = highcut / nyq
    b, a = butter(order, [low, high], btype='band')
    return b, a

def bandpass_filter(data, lowcut, highcut, fs, order=5):
    b, a = butter_bandpass(lowcut, highcut, fs, order=order)
    return filtfilt(b, a, data)

# =====
# APLICACIÓN DEL FILTRO PASA-BANDA
# =====
filtered_noise = bandpass_filter(noise, 3000, 6000, fs)
# =====
# CREACIÓN DE 10 AUDIOS DE 5 SEGUNDOS
# =====
output_folder = "Audio recibido" # Carpeta donde se guardarán los audios
os.makedirs(output_folder, exist_ok=True) # Crear la carpeta si no existe

# Dividimos el ruido en 10 partes y guardamos cada una con un nombre del 1 al 10
for i in range(10):
    start_sample = i * 5 * fs # Inicio del fragmento
    end_sample = (i + 1) * 5 * fs # Fin del fragmento
    segment = filtered_noise[start_sample:end_sample]

    # Aplicamos un factor de intensidad aleatorio para cada archivo
    random_factor = np.random.uniform(1, 32767) # Intensidad aleatoria entre
    10000 y 32767
    segment = (segment / np.max(np.abs(segment)) * random_factor).astype(np.int16)

    # Guardamos el archivo de audio
    filename = os.path.join(output_folder, f"Audio {i+1}.wav")
    sf.write(filename, segment, fs)
```

ANEXO XII: QR PARA AUDIOS DEL UAV GRABADOS DESDE TONOR TC30



ANEXO XIII: QR PARA AUDIOS SONIDO EXTERNO



ANEXO XIV: QR PARA AUDIOS DEL UAV GRABADOS DESDE LOGITECH C615



ANEXO XV: QR PARA AUDIOS DE RUIDO GENERADO

