



Universidad de Jaén

Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

MONITORIZACIÓN DE PLAGAS DE INSECTOS MEDIANTE REDES DE SENSORES INALÁMBRICOS.

Alumno: Jesús Calzada Sarrión

**Tutor: Damián Martínez Muñoz, Manuel
Ángel Gadeo Martos.**

Depto.: Ingeniería de Telecomunicación

Junio, 2018



Universidad de Jaén

Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

MONITORIZACIÓN DE PLAGAS DE INSECTOS MEDIANTE REDES DE SENSORES INALÁMBRICOS.

Alumno: Jesús Calzada Sarrión

**Tutor: Damián Martínez Muñoz, Manuel
Ángel Gadeo Martos.**

Depto.: Ingeniería de Telecomunicación

Junio, 2018

Fdo Alumno:

Fdo: Tutores:

Índice

1. INTRODUCCIÓN	7
2. ESTADO DEL ARTE	9
2.1. Microcontroladores y placas programables	9
2.1.1. Historia	9
2.1.2. Definición, componentes y tipos	10
2.1.3. Arduino	11
2.1.3.1. Tipos de Arduino	11
2.2. El Lenguaje Arduino	17
2.2.1. El lenguaje C	17
2.2.2. El lenguaje C++	17
2.2.3. Lenguaje Arduino	18
2.3. Caracterización del sonido	19
2.3.1. Sonoridad	19
2.3.2. Tono	21
2.3.3. Timbre	22
2.4. Detección de plagas de insectos	23
3. DEFINICIÓN DE LA SOLUCIÓN	26
3.1. Transformada de Fourier	26
3.2. Implementación rápida de Fourier en software	27
4. HARDWARE NECESARIO	30
4.1. Arduino UNO	30
4.2. Arduino DUE	31
4.3. Micrófono	31
5. IMPLEMENTACIÓN DE LA SOLUCIÓN (I): ANALIZANDO INSECTOS EN MATLAB	33
5.1. Código en Matlab	33

5.1.1.	Primera parte: Analizando la señal de entrada _____	33
5.1.2.	Segunda parte: Realizando la transformada de Fourier _____	34
5.1.3.	Tercera parte: Histograma y número de tramas _____	35
5.1.4.	Cuarta parte: Obteniendo los valores del reconocimiento _____	36
5.1.5.	Quinta parte: Identificación del insecto _____	37
5.2.	Extrayendo el perfil de cada insecto _____	38
6.	<i>IMPLEMENTACIÓN DE LA SOLUCIÓN (II): RECONOCIENDO INSECTOS EN ARDUINO</i>	40
6.1.	Parte 1: Estructura principal _____	40
6.2.	Parte 2: Muestreando sonido _____	44
6.3.	Parte 3: Procesando la trama _____	45
6.4.	Parte 4: Reconociendo insecto _____	46
6.4.1.	Inicializando insectos _____	47
6.4.2.	Análisis de tramas _____	49
6.4.3.	Ventanas y similitud _____	51
7.	<i>RESULTADOS</i> _____	56
7.1.	Tests de Optimización de Parámetros para el Reconocimiento de Insectos	56
7.1.1.	Tamaño de la ventana _____	56
7.1.2.	Relación señal ruido (SNR) _____	57
7.1.3.	Solapamiento _____	57
7.1.4.	Número de tramas _____	57
7.1.5.	Fórmula de similitud: _____	57
7.2.	Tests realizados _____	60
7.2.1.	Test realizados en Matlab _____	60
7.2.1.1.	Primera Ronda de Tests de Optimización _____	60
7.2.1.2.	Segunda Ronda de Tests de Optimización _____	65
7.2.1.3.	Tercera fase de test: Resultados aplicados _____	71
7.2.2.	Conclusión de los resultados de los tests de Matlab _____	77
7.3.	Test de Evaluación sobre el Sistema Final _____	79
7.3.1.	Configuración de los test sobre el sistema final _____	79
7.3.2.	Conclusión de los resultados sobre los tests Evaluación sobre el Sistema Final _____	80

8.	CONCLUSIÓN	84
9.	REFERENCIAS	85
10.	PRESUPUESTO	87
10.1.	Recursos materiales	87
10.2.	Honorarios	87
10.3.	Presupuesto total	87
	Anexo 1: Detalle de la Primera Ronda de Tests de Optimización	88
	Anexo 2: Detalle de la Segunda Ronda de Test de Optimización	103
	Anexo 3: Test finales de Matlab.	110
	Anexo 4: Tests de Evaluación Sistema Final	121

1. INTRODUCCIÓN

Hoy en día es posible alcanzar una muy alta productividad en el sector agrícola mediante el uso de (1) fertilizantes, (2) el monocultivo de las variedades de alta producción o (3) el cultivo múltiple mediante la reducción o eliminación de los suelos descansados [1], entre otros. Sin embargo, esta alta productividad es un caldo de cultivo excelente para la proliferación de plagas, especialmente de plagas de insectos, que pueden llegar a ocasionar graves pérdidas de rendimiento y recursos [1].

Es por esta razón que la continua monitorización de los cultivos es necesaria para conseguir una detección temprana y evitar su expansión, ya que el número de insectos atacantes crece de una forma exponencial cuanto mayor es su población, registrando números de hasta decenas de miles de millones en algunos casos como el de la langosta [2].

Por lo tanto, el desarrollo de un sistema que automáticamente monitorice una cosecha y nos permita identificar una posible plaga en tiempo real y lo antes posible sería altamente beneficioso e incrementaría considerablemente nuestro tiempo de reacción para atajar el problema antes de que se torne demasiado grande.

En este proyecto trataremos de identificar posibles plagas de una forma temprana, a través de un sensor acústico conectado a un microcontrolador. Esto permitiría no solamente ahorrar tiempo y dinero en la monitorización de cosechas sino que además posibilitaría la detección de una posible plaga en un tiempo más reducido.

Para que una solución de este tipo fuera exitosa, sería crucial reconocer qué tipo de insecto emite el sonido reconocido por nuestro micrófono, para después proceder a evaluar su frecuencia y cantidad y si es necesario alertar sobre una posible plaga. Para este proyecto hemos desarrollado una aplicación software que se basa en las siguientes partes:

1. Conexión entre el micrófono y el microcontrolador
2. Muestreo continuo del sonido ambiente
3. Procesamiento del sonido ambiente para obtener la frecuencia del sonido
4. Reconocimiento del patrón muestreado y comparación con muestras de insectos para reconocer qué insecto está transitando cerca del sistema

Durante el resto de esta memoria revisaremos la base teórica de este proyecto, los diferentes componentes hardware y software del sistema desarrollado y detallaremos los tests y resultados obtenidos tras la aplicación de la solución. Para cubrir todos estos aspectos, el texto está dividido de la siguiente forma:

- El capítulo 2 detalla el estado del arte, primero centrándose en los tres aspectos básicos de nuestro proyecto: hardware (microcontroladores), software (lenguajes de programación utilizados) y teoría (caracterización del sonido); para luego continuar con un breve resumen de otras técnicas y soluciones para el reconocimiento de insectos a través de su sonido recientemente utilizadas.
- El capítulo 3 explica cómo la Transformada de Fourier puede utilizarse para procesar la señal de entrada analógica y obtener la frecuencia del sonido en hercios y veremos cómo funciona la librería que utilizaremos para realizar esta operación en nuestro programa.
- En el capítulo 4 detallamos los componentes hardware de la solución que hemos ideado para este proyecto y justificamos su elección frente a otras alternativas.
- El capítulo 5 describe el entorno de simulación y análisis del reconocimiento que hemos desarrollado para optimizar distintos parámetros del sistema antes de exportar su lógica a la placa del sistema final en Arduino, para ello se ha utilizado el software de Matlab.
- El capítulo 6 detalla la implementación de la solución software en Arduino, explicando cada parte de la misma y señalando los aspectos más importantes del código y el porqué de su diseño final y su implementación.
- El capítulo 7 lista los tests que se diseñaron tanto en el entorno de Matlab para optimizar parámetros como en el sistema final para medir la eficacia de la solución implementada en este proyecto además de sus resultados y limitaciones.
- Finalmente, el capítulo 8 concluye el proyecto resumiendo lo hecho hasta la fecha y las oportunidades de desarrollo futuro que este trabajo posibilita.

2. ESTADO DEL ARTE

El siguiente capítulo resume las bases, teoría y algo de historia sobre los puntos claves de este proyecto. La primera parte de este capítulo se centrará en el hardware, exponiendo un breve resumen sobre microcontroladores y placas programables, para después centrarnos en la descripción y justificación para utilizar placas de Arduino. Después, moviéndonos hacia la parte software, se describirán los lenguajes de programación más relevantes y se enumerarán las características principales del lenguaje Arduino. Por último se detalla la caracterización del sonido para finalizar con una breve revisión de la literatura sobre detección de insectos mediante su sonido.

2.1. Microcontroladores y placas programables

2.1.1. *Historia*

Los primeros microcontroladores empezaron a comercializarse en 1974 gracias a unos ingenieros de Texas Instruments [3]. Combinaba memoria ROM [4], memoria RAM [5], microprocesador [6] y un reloj en un chip y su destino iba a ser para los sistemas embebidos [7].

Hoy en día es uno de los logros más sobresalientes del siglo XX. Hace un cuarto de siglo tal afirmación habría parecido absurda. Pero cada año, el microcontrolador se acerca más al centro de nuestras vidas, forjándose un sitio en el núcleo de una máquina tras otra. Hornos microondas, cepillos de dientes eléctricos, lavavajillas, robot de limpieza, automóviles, son tan solo algunos ejemplos de los muchos objetos que lo tienen.

Las placas de desarrollo de hardware se basan en un microcontrolador que se nos permite programar y crear funcionalidades de gran diversidad de tipos como sensores de movimientos, alarmas, detectores de luz, reconocimiento de voz o como en nuestro caso un detector de plagas.

2.1.2. Definición, componentes y tipos

Un microcontrolador es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. En su interior incluye las tres principales unidades funcionales de una computadora:

- Unidad central de procesamiento o procesador [6]: Es la parte de un microcontrolador en la que se encuentran los elementos que sirven para procesar datos.
- Memoria. [4] [5]: Es el dispositivo que retiene, memoriza o almacena datos informáticos durante algún periodo de tiempo. Normalmente un microcontrolador posee tres dispositivos de memoria: ROM, RAM y Flash, las cuales son utilizados de la misma forma que en cualquier ordenador.
- Periféricos de entrada y salida. [8]: Son dispositivos que permiten la comunicación entre un sistema de procesamiento de información, tal como la computadora y el mundo exterior, posiblemente un humano u otro sistema de procesamiento de información. Ejemplos de periféricos de entrada pueden ser un micrófono, un teclado o un ratón; así como de salida pueden ser un altavoz, una pantalla o una impresora.

Existen dos tipos de placas programables: aquellas capaces de albergar un sistema operativo y las que no. Esto difiere en si poseen un microprocesador [6]. Aquellas placas que sí lo tienen se les conoce como computador de placa reducida, computador de placa única o computador de placa simple. Algunos modelos de este tipo de placas son Raspberry Pi [9] o ECB AT91 [10].

Las placas de desarrollo de hardware que no tienen un microprocesador, tienen un microcontrolador. Una analogía que nos puede ayudar a entender la diferencia entre ellos sería el que un microcontrolador sería un autómatas y un microprocesador sería un ordenador.

En un proyecto la elección de escoger uno u otro dependería de la aplicación que se le quiera dar. Si lo que se quiere es recolección de datos, envío de alarmas, accionar motores o como en nuestro caso supervisar el entorno la opción más acertada sería un microcontrolador. Por otro lado, el tratamiento de datos recogidos, la interfaz gráfica de usuario o el envío de correos electrónicos lo dejaríamos para un microprocesador, ya sea un ordenador, una Raspberry Pi o similar.

La placa de desarrollo con microcontrolador más común es Arduino y en ella existen diferentes modelos que pasamos a detallar en la siguiente sección.

2.1.3. *Arduino*

Las placas Arduino se iniciaron en el año 2005 como un proyecto enfocado para estudiantes en Ivrea (Italia). En aquellos años, los estudiantes usaban un microcontrolador BASIC Stamp [11]. En la actualidad las placas Arduino son muy famosas debido a su compatibilidad de unir varios módulos de hardware adicionales (que son denominados Shields) [12] y el lenguaje de programación en C [13] que puede integrarse a distintos objetos.

2.1.3.1. *Tipos de Arduino*

En la actualidad Arduino dispone de una amplia variedad de placas y shields para usar dependiendo de nuestras necesidades. A continuación nombraré las placas de las que Arduino dispone con una breve definición que cada una de ellas:

1. Arduino Uno:

Es la placa estándar y la más conocida y documentada. Salió a la luz en septiembre de 2010 sustituyendo su predecesor Duemilanove [14] con varias mejoras de hardware que consisten básicamente en el uso de un USB HID propio en lugar de utilizar un conversor FTDI para la conexión USB. Su microcontrolador es un Atmega328p [15] con 32 Kbytes de ROM para el programa. El esquema del Arduino Uno se detalla en la referencia [16]

Si se quieren ver más especificaciones se pueden ver en la Tabla 1.

2. Arduino Mega:

Es el más potente de las placas con microcontrolador de 8 bits y el que más pines entrada/salida tiene, apto para trabajos ya algo más complejos aunque tengamos que sacrificar un poco el espacio. El microcontrolador es el Atmega2560 [17]. El esquema del Arduino Mega se detalla en la referencia [18].

Si se quieren ver más especificaciones se pueden ver en la Tabla 1.

3. Arduino Ethernet:

Incorpora un puerto ethernet, está basado en el Arduino Uno y nos permite conectarnos a una red o a Internet mediante su puerto de red.

4. Arduino Due:

Arduino con la mayor capacidad de procesamiento, basado en un microcontrolador de 32 bit y arquitectura ARM: Atmel SAM3X8E ARM Cortex-M3 CPU [19]. Este Arduino está alimentado a 3.3V y dado que gran parte de los shields, sensores, actuadores para Arduino y compatible son a 5V lo limita, pero cada vez se ven más elementos donde se puede elegir el voltaje entre 3.3 y 5V. El esquema del Arduino Due se detalla en la referencia [20]

Si se quieren ver más especificaciones se pueden ver en la Tabla 1.

5. Arduino Leonardo:

Este Arduino tiene la particularidad de eliminar la necesidad de tener un segundo procesador debido a que está formado por un microcontrolador ATmega32u4 [21] que tiene integrada la comunicación USB. Esto hace que el Arduino Leonardo no se comporte de la misma forma al conectarlo al ordenador que los demás modelos de Arduino por lo que es una versión de Arduino no recomendada para principiantes.

Si se quieren ver más especificaciones se pueden ver en la Tabla 1.

6. Arduino Micro:

Basado también en el procesador ATmega32u4 pero más compacto.

Ejemplo de placa para uso de Arduino con bornas y de pequeño tamaño.

Si se quieren ver más especificaciones se pueden ver en la Tabla 1.

7. Arduino Mini:

Versión miniaturizada de la placa Arduino UNO basado en el ATmega328. Mide tan sólo 30x18mm y permite ahorrar espacio en los proyectos que lo requieran. Las funcionalidades son las mismas que Arduino UNO.

Si se quieren ver más especificaciones se pueden ver en la Tabla 1.

8. Arduino Lilypad:

Diseñado para dispositivos “wearables” y e-textiles [22]. Para coser con hilo conductor e instalarlo sobre prendas.

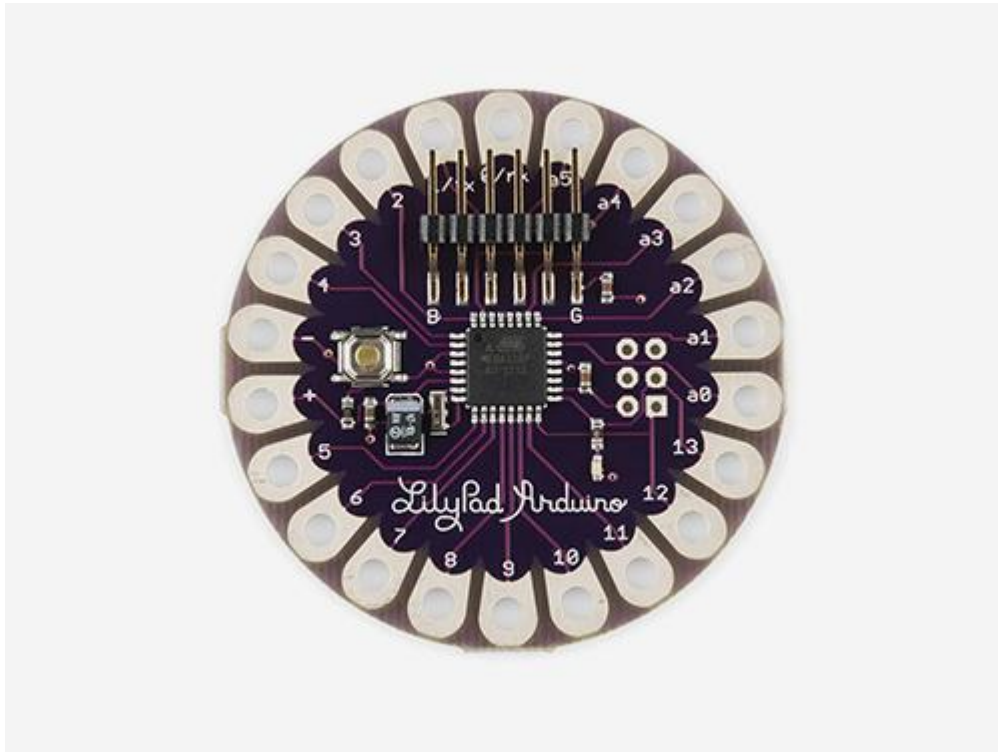


Imagen 2.1. Arduino LilyPad el cual se utiliza para hacer textiles electrónicos también conocidos como prendas inteligentes

9. Arduino Yun:

Es diferente al resto de placas Arduino porque además de llevar un microcontrolador, incorpora un microprocesador con un sistema operativo Linux embebido. La ventaja que tiene este modelo es que el microcontrolador y el microprocesador están conectados mediante un puerto serie y Arduino nos ofrece una serie de librerías para facilitar la interconexión entre ellos.

2.1.3.2. *Arduino Due*

Como ya hemos citado, la versión escogida para este proyecto ha sido Arduino Due, en este apartado se detallarán las especificaciones más importantes de dicha placa.

Arduino Due es la primera tarjeta de desarrollo construida con un microcontrolador de 32 bit CortexM3 ARM el cual puede ser programado mediante el IDE de Arduino.

El Arduino Due posee 54 pines digitales de entrada y salida, 12 entradas análogas, 2 salidas análogas, 4 UART (puertas seriales por hardware) [22], cristal oscilador [23] de 84MHz, Jack de poder (Imagen 2), conexión JTAG [24], botón reset y un botón borrar.



Imagen 2.2. Jack de poder.

A diferencia de otras tarjetas, Arduino Due trabaja con 3,3V, pudiendo tolerar un voltaje máximo en sus pines I/O de 3.3V. Alimentar los mismos con voltajes más altos, como 5V por ejemplo, podría dañar la tarjeta.

Estas son las características técnicas de la placa:

- Microcontrolador: AT91SAM3X8E.
- Voltaje de operación: 3.3V.
- Voltaje recomendado de entrada (pin Vin): 7-12V.
- Pines de entrada y salida digitales: 54 pines I/O, de los cuales 12 proveen salida PWM.
- Pines de entrada análogos: 12.
- Pines de salida análogos: 2.
- Corriente de salida total en los pines I/O: 130mA.
- Corriente DC máxima en el pin de 3.3V: 800mA.
- Corriente DC máxima en el pin de 5V: 800mA.
- Memoria Flash: 512 KB toda disponible para aplicaciones del usuario.

- SRAM: 96 KB (en dos bancos de: 64KB y 32KB).
- Velocidad de reloj: 84 MHz

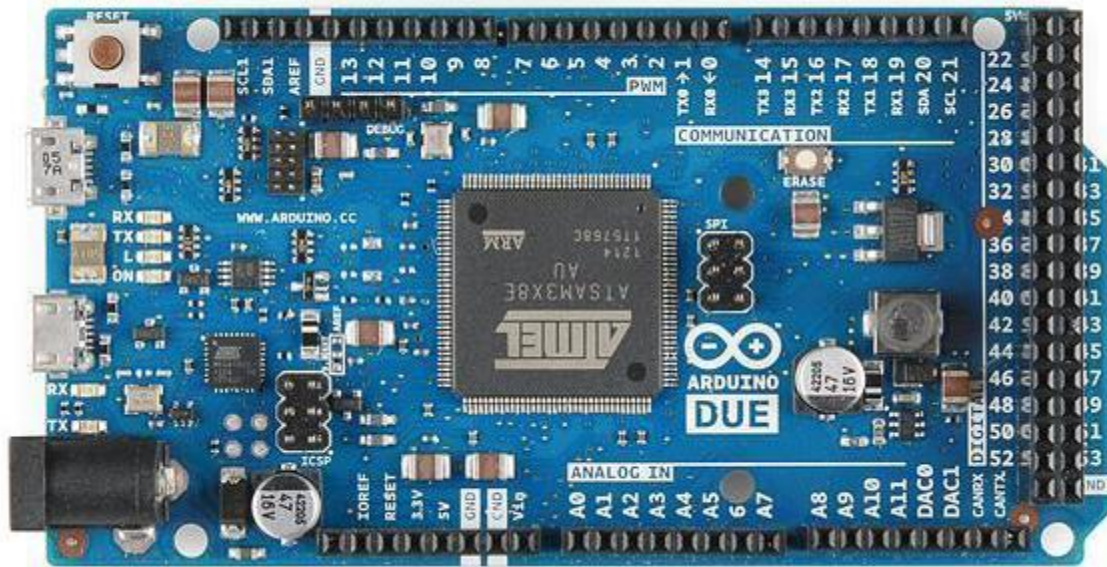


Imagen 2.3. Placa de desarrollo Arduino Due.











										
Fabricante	Arduino	Arduino	Arduino	Arduino	Arduino	Arduino	Arduino	Netduino	Texas Instruments	Fundación Raspberry Pi
Modelo	Pro Mini	Nano	Uno	Mega / Mega 2560	Leonardo	Micro	Due	Netduino 2	Stellaris Launchpad LM4F120	Raspberry Pi Mod.B
Microcontrolador	AVR Atmega 168 ó 328 8bits	AVR ATmega 168 ó 328 8bits	AVR ATmega 328 8bits	AVR ATmega2560 8bits	AVR ATmega 32u4 8bits	AVR ATmega 32u4 8bits	ARM SAM3X8E Cortex-M3 32bits	ARM STMMicro STM32F2 Cortex-M3 32bits	ARM LM4F120H5QR Cortex-M4 32bits	ARM Broadcom BCM2835
Frecuencia	16Mhz	16Mhz	16Mhz	16Mhz	16Mhz	16Mhz	84Mhz	120Mhz	80Mhz	700Mhz
Memoria RAM	2KiB	2KiB	2KiB	8KiB	2.5KiB	2.5KiB	96KiB (64+32KiB)	60KiB	32KiB	512MiB
Memoria EEPROM	1KiB	1KiB	1KiB	4KiB	1KiB	1KiB	0	0	-	-
Memoria FLASH	16 ó 32KiB	16 ó 32KiB	32KiB	128 ó 256KiB	32KiB	32KiB	512KiB	192KiB	256KiB	-
Pines digitales entradas/salidas	14/14	14/14	14/14	54/54	20/20	20/20	54/54	20/20	43/43	8/8
Tensión/corriente pines digitales	3.3v ó 5v 40mA	5v 40mA	5v 40mA	5v 40mA	5v 40mA	5v 40mA	3.3v 3~15mA (130mA entre todos)	3.3v~5v 25mA (125mA entre todos)	5v	-
Pines analógicos entradas/salidas	6/0	8/0	6/0	16/0	12/0	12/0	12/2	6/0	-	-
Tensión/resolución pines analógicos	3.3v ó 5v 10bits (1024 valores)	5v 10bits (1024 valores)	5v 10bits (1024 valores)	5v 10bits (1024 valores)	5v 10bits (1024 valores)	5v 10bits (1024 valores)	3.3v 12bits (4096 valores)	5v 12bits (4096 valores)	-	-
Pines con interrupción externa	2	2	2	6	2	2	-	-	-	-
Pines PWM	6	6	6	15	7	7	12	6	-	-
Conexiones Serial / UART	1	1	1	4	1	1	4	4	8	Si
Conexiones I2C / TWI	1	1	1	1	1	1	2	1	4	Si
Conexiones ISP / ICSP	1	1	1	1	1	1	1	1	-	Si
Conexión USB	No (necesita adaptador externo)	Si	Si, USB-B	Si, USB-B	Si, Nativa, MicroUSB	Si, Nativa, MicroUSB	Si, Nativa, MicroUSB	Si, Nativa, MicroUSB	Si, Nativa, MicroUSB	Si, MicroUSB
Conexión USB de depuración	No	No	No	No	No	No	Si, MicroUSB	Si, MicroUSB	Si, MicroUSB	-
Conexión Bluetooth	No	No	No	No	No	No	No	No	No	-
Conexión WiFi	No	No	No	No	No	No	No	No	No	-
Conexión Ethernet	No	No	No	No	No	No	No	No	No	Si
Conexión USB Host	No	No	No	No	No	No	Si	No	Si	Si
Almacenamiento por SD	No	No	No	No	No	No	No	No	No	Si
Corriente en el pin de 5v	-	500mA	500~800mA	500~800mA	500~800mA	500mA	800mA	-	-	-
Corriente en el pin de 3.3v	-	50mA	50mA	50mA	50mA	50mA	800mA	-	-	-
Voltaje de alimentación por el USB	3.3v ó 5v (sin usb)	5v	5v	5v	5v	5v	5v	5v	5v	5v
Voltaje de alimentación recomendado por el Jack	3.35 -12 V (modelo 3.3V) ó 5 - 12 V (modelo 5V)	7~12v	7~12v	7~12v	7~12v	7~12v	7~12v	7.5~9v	-	-
Voltaje de alimentación limite por el Jack	-	6~20v	6~20v	6~20v	6~20v	6~20v	6~20v	-	-	-
Precio oficial	15+gi	-	20€+gi	40€+gi	18€+gi	18€+gi	39€+gi	~35\$+gi	13\$+gi	~43\$+gi
Precio BBB	~4€	~9€	~10€	~12€	11€~	~16€	~38€	25~30€	~15€	~35€

Imagen 2.4. Tabla de especificaciones de algunas placas electrónicas.

El dispositivo en el que nosotros nos hemos centrado es la placa Arduino y más concretamente la versión DUE, ver Imagen 2.4.

2.2. El Lenguaje Arduino

Los microcontroladores Arduino se pueden programar utilizando el lenguaje del mismo nombre [25]. Aunque está considerado por sus fabricantes y muchos usuarios como un lenguaje de programación por sí mismo, está ampliamente basado en los lenguajes clásicos de programación llamados C y C++. Las siguientes secciones resumen brevemente estos lenguajes para luego resaltar las características más interesantes del lenguaje Arduino utilizadas en nuestro proyecto.

2.2.1. El lenguaje C

El lenguaje C se desarrolló inicialmente a principios de los años 70 en los laboratorios Bell [26]. Está considerado como uno de los más eficientes en cuanto al código que se produce con el y es el más utilizado para crear software de sistemas [26]. Normalmente se clasifica como un lenguaje de medio nivel, ya que aunque posee estructuras típicas de los lenguajes de alto nivel (como estructuras de datos o bucles), dispone de funciones y construcciones que permiten control a muy bajo nivel (tales como control sobre posiciones de memoria y uso de la misma). Fue estandarizado a principios de los 90 con el estándar ISO/IEC 9899:1990, cuya adopción es muy amplia ya que permite portar programas entre distintas plataformas y/o arquitecturas [26].

Durante el diseño del lenguaje C, uno de los principales objetivos fue que solamente fueran necesarias unas pocas instrucciones del lenguaje para traducirlo al lenguaje máquina, sin necesitar demasiado tiempo de compilación y ejecución. Es por ello que es posible escribir C a muy bajo nivel de abstracción e incluso se suele utilizar como lenguaje intermediario entre diferentes lenguajes [26]. Dadas estas características y que C posee un conjunto de estructuras de control relativamente sencillas hacen que C esté disponible y haya sido extendido en una gran variedad de plataformas y sea ampliamente utilizado en microcontroladores como Arduino UNO y DUE.

2.2.2. El lenguaje C++

C++ fue diseñado e implementado a mediados de los 80 como una extensión de C que permitiera la manipulación de objetos y permitir así la programación orientada a objetos que estaba entrando en alza en aquel tiempo [27]. Sin embargo, desde este estricto punto de vista, C++ se entiende como un lenguaje híbrido, ya que aunque permite la programación orientada a objetos, no la impone para todo el código.

Una de las características más sonadas de C++ es la posibilidad de re-definir operadores como la suma, el igual o la resta, posibilitando así que tipos definidos por el programador se comporten como tipos fundamentales.

Al ser una extensión del C original, posee similitudes en cuanto a las ventajas para incorporarlo en un microcontrolador, lo cual, junto con la capacidad de representar conceptos del mundo real mediante objetos, lo hacen un candidato ideal sobre el que basar el lenguaje Arduino, que se describe a continuación.

2.2.3. *Lenguaje Arduino*

Los microcontroladores de Arduino presentados en secciones anteriores se programan utilizando el lenguaje que lleva el mismo nombre. Este lenguaje Arduino está principalmente basado en C y puede ser fácilmente expandible usando librerías que pueden estar escritas en C y/o C++ [25]. De hecho, es posible trasladar programas escritos puramente en C a Arduino siguiendo ciertas reglas de configuración.

El lenguaje Arduino se puede entender como el conjunto de cuatro grandes bloques: variables, estructuras de control, estructura principal y funciones propias [28]:

- Variables: Son referencias a valores que pueden ser enteros, decimales, cadenas de caracteres, listas u otros tipos definidos por el usuario. Las variables normales pueden cambiar de valor en tiempo de ejecución y además existen las constantes cuyos valores no pueden cambiar durante la ejecución del programa. Estas características son compartidas por C y C++.
- Estructuras de Control: Al igual que cualquier programa en C/C++ u otros lenguajes, Arduino posee varias estructuras de control tales como bucles “for” y “while” y condicionales como “if ... else”.
- Estructura Principal: Todo programa Arduino debe implementar las funciones “setup” and “loop” en su función principal (“main”). La función setup inicializa el entorno de

ejecución y la función loop se ejecuta una y otra vez en bucle hasta que cierta condición de parada se cumpla. Para programas complejos y/o extensos que requieren de muchas líneas de código, el usuario puede especificar otras funciones propias que pueden ser llamadas desde cualquier parte de esta estructura principal. Nótese que C y C++ difieren de Arduino en este punto ya que no requieren de las funciones setup y loop dentro de su función principal.

- Funciones propias: Además de las características destacadas anteriormente, el lenguaje Arduino incorpora una serie de utilidades en forma de funciones para facilitar el trabajo del usuario y simplificar el código fuente producido. Algunas de estas funciones son matemáticas (para ejecutar cálculos de raíces cuadradas o trigonometría por ejemplo) o para leer y escribir en los puertos de entrada y salida del microcontrolador (por ejemplo para leer señales analógicas y digitales o para escribir resultados en el puerto de salida). Algunas de estas funciones serán utilizadas en este proyecto, y serán descritas en detalle en capítulos posteriores.

Algunas de las funciones propias del lenguaje Arduino utilizadas en este proyecto son en relación a la lectura de señales de sonido analógicas en el programa para detección de insectos. Para entender mejor el funcionamiento de las mismas, es necesario comprender la caracterización y propiedades del sonido en sí, las cuales son descritas en la siguiente sección.

2.3. Caracterización del sonido

El sonido audible consiste en ondas sonoras que se producen cuando las oscilaciones de la presión del aire son convertidas en ondas mecánicas en el oído humano y traducidas a impulsos eléctricos que finalmente percibe el cerebro. De igual forma la propagación del sonido en los fluidos consta de las variaciones de presión en este medio, sólo que la presión aquí es diferente por lo que la propagación del sonido también será diferente. Para entender mejor cómo varía el sonido en el oído humano hemos de entender los conceptos de sonoridad, tono y timbre, que son descritos en los siguientes apartados.

2.3.1. Sonoridad

La sonoridad o intensidad sonora, es un atributo de los sonidos en función del cual se pueden ordenar en una escala de más bajo a más alto en intensidad. La intensidad sonora se puede interpretar en decibelios (dB) como aparece en la *Figura 2.5*.



Figura 2.5. Ejemplo de ruidos y sonoridad en dB.

Para el oído humano esta sonoridad, o intensidad sonora, varía en función de la frecuencia en la que se emite el sonido, ya que el oído humano no tiene una audición en frecuencia lineal sino que es más sensible para determinadas frecuencias (bandas críticas) [29].

Fletcher y Munson establecieron las primeras curvas isofónicas en 1930 [30]. Las curvas isofónicas calculan la relación existente entre la frecuencia y la intensidad de dos sonidos para que estos sean percibidos como igual de fuertes por el oído, con lo que todos los puntos de una misma curva isofónica tiene la misma sonoridad (*Figura 2.6*).

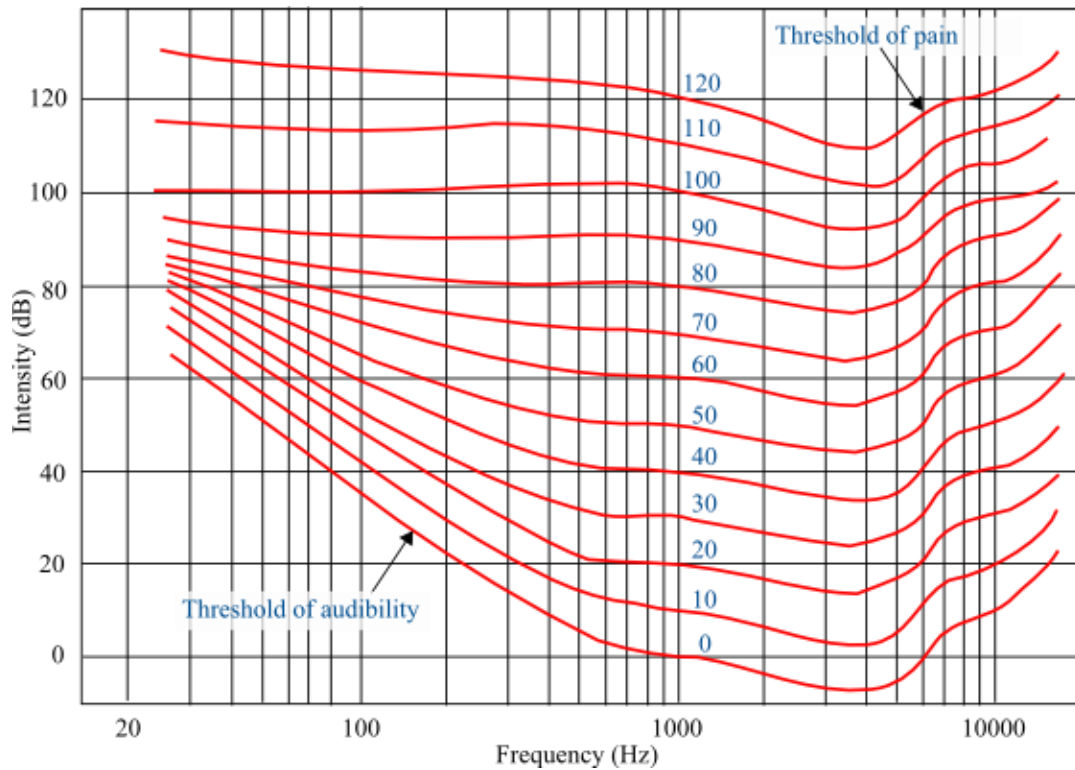


Figura 2.6. Curvas isofónicas de Fletcher & Munson

Se observa que entre 1 y 4kHz el oído humano es más sensible, las curvas isofónicas son más bajas, y que por debajo de los 25Hz y por encima de los 20kHz aproximadamente la sonoridad es inexistente para las personas. El sonido emitido por los insectos que nos atañen en este proyecto suele centrarse en el espectro audible para el oído humano, normalmente entre 4 y 8kHz.

2.3.2. Tono

El tono es la frecuencia de la onda sonora, es decir, es la propiedad del sonido por la cual el oído le asigna un lugar en una escala musical dada la vibración de dicho sonido, de modo que el oído asigna a cada tono un lugar pero no a sonidos de otras frecuencias. Por ejemplo:

Sabemos que al pulsar una cuerda de guitarra tensa se produce un sonido que nos produce una determinada sensación de tono, pero aumentamos la tensión, aumenta la frecuencia de la onda emitida, el sonido se hace menos grave o más agudo.

La clasificación del tono para el oído humano se podría denotar de la siguiente forma:

- Sonidos graves: 20 ~500 (Hz)
- Sonidos medios: 500 ~5000 (Hz)
- Sonidos agudos: 5000 ~20000 (Hz)

Como hemos notado en la sección anterior, los insectos en los que nos centraremos para la identificación de plagas están en el rango de 4 a 8KHz, por lo que podrían clasificarse como sonidos medios y agudos, aunque mayormente agudos.

2.3.3. *Timbre*

Es otra sensación auditiva aunque no se pueden ordenar los sonidos en función del timbre en una escala. El timbre es la propiedad por la cual dos sonidos se distinguen como diferentes aunque tengan la misma sonoridad y el mismo tono, es decir, nos permite distinguir la misma nota musical tocada con un piano y una flauta, por ejemplo.

El timbre se detecta en el cerebro al estudiar la composición de señales en diferentes bandas críticas, las bandas críticas son cada uno de los 24 receptores selectivos que tiene el oído en la membrana basilar [31].

Los sonidos simples o tonos puros son ondas sinusoidales de una frecuencia determinada. Sin embargo, en la naturaleza, no existe ese sonido puro, libre de armónicos [32].

El teorema de Fourier [33] demuestra que cualquier forma de onda periódica puede descomponerse en una serie de ondas (armónicos) con una frecuencia que es múltiplo de la onda original, llamada frecuencia fundamental [34]. Así, los armónicos son múltiplos de la frecuencia fundamental a la que acompañan como se aprecia en la *Figura 2.7*.

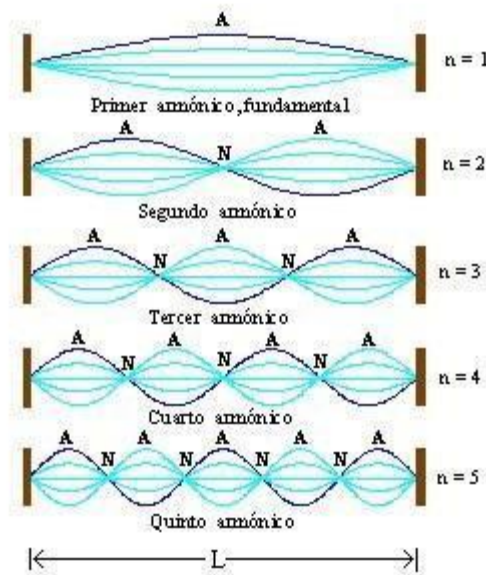


Figura 2.7. Armónicos de una onda.

El timbre viene determinado por la cantidad e intensidad de estos armónicos. A veces, como en el caso del oboe, estos armónicos pueden tener una amplitud igual o superior a la forma de onda fundamental.

El Teorema de Fourier [35] mencionado anteriormente al timbre, será clave en la implementación de nuestro proyecto, ya que nos permitirá descomponer las muestras de sonido obtenidas por el micrófono para saber cuál es la frecuencia exacta del sonido escuchado. En el próximo capítulo detallaremos más a fondo dicho teorema.

2.4. Detección de plagas de insectos

Una plaga es el conjunto de insectos de una misma especie que se alimentan de una planta o cultivo y afectan el desarrollo o crecimiento de la misma, provocando una pérdida en la producción. Desde tiempos remotos, el hombre se ha visto afectado por las distintas especies de insectos que han provocado daños inmensos a las cosechas.

Originariamente la detección de plagas se realizaba mediante la inspección, se observaban los huertos con tal de hallar algunos insectos en grandes cantidades para intentar acabar con ellos.

Al cabo de mucho tiempo, principios de los 80, la incapacidad de poder detectar una plaga a tiempo hizo que se tomaran medidas drásticas, como era el rociar los campos con

productos químicos tales como el DDT [36] el cual al cabo de los años se demostró que era nocivo para la salud [37].

En el siglo XXI se empezó a usar la tecnología para intentar detectar a tiempo las plagas. A continuación expongo algunos de estos métodos empleados:

- Termografía:

Los nidos de termitas en el interior de una vivienda tienen una mayor humedad que el resto de la casa lo que hace que exista una refrigeración anómala y por lo tanto diferencias de temperatura en la casa. Esto hace que con una cámara infrarrojos se pueda detectar plagas de termitas en las viviendas [38]. Vease Imagen 2.8.

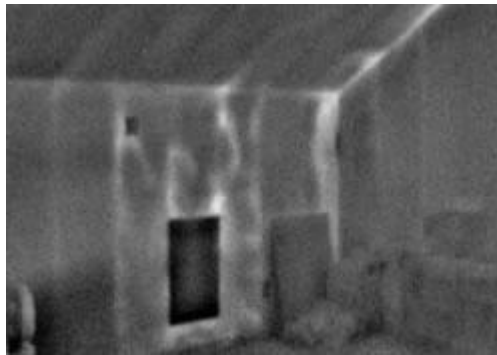


Imagen 2.8. Visión infrarrojos de un edificio con nidos de termitas.

- Por sus movimientos:

El sistema consiste en diversos sensores de micrófonos, diodos de emisión de luz [39] y película piezoeléctricas [40] que recogen muestras de señales infrarrojas, acústicas y vibraciones. El software analiza las señales para crear un perfil del insecto objetivo y distinguirlo de otros insectos. [41]

- Captura de imágenes:

Los sistemas avanzados de cámaras combinados con procedimientos eficientes para calibrar y corregir los datos de reflectancia foliar, permiten detectar las respuestas de las plantas a niveles muy bajos de infestación de plagas. De esta manera se puede detectar las infestaciones de plagas.

La tecnología de captura de imágenes avanzadas requieren grandes volúmenes de datos, los cuales, debido a la necesidad del análisis, requieren cierto tiempo de procesamiento en la computadora.

En cuanto a técnicas de para detectar insectos mediante aplicaciones software, Zhu Le-Qing demostró en 2011 [42] que este es un problema complejo de aproximar, ya que propuso una red neuronal probabilística como el método que que mayor porcentaje de acierto podía obtener. El método que se presenta en su investigación obtuvo un 96% de acierto entre 50 insectos analizados en su experimento. Nuestro proyecto intenta aproximar este mismo problema de una forma teóricamente más sencilla y computacionalmente menos compleja, utilizando la Transformada de Fourier para descomponer el sonido inicial del insecto en sus frecuencias originales para luego fijarnos en la frecuencia máxima.

En cuanto a aplicaciones móviles, Cicada Hunt [43] intenta aproximar este problema usando la misma técnica que la popular aplicación Shazam para reconocer canciones. Solo que en este caso la aplicación móvil depende de sus propios usuarios para incrementar su registro de insectos. Además, los desarrolladores de Cicada Hunt no pretenden que su programa se utilice para detectar plagas, sino simplemente como diversión para usuarios que hagan camping y quieran identificar el insecto que escuchan. De esta forma podemos concluir que su porcentaje de acierto podría no ser muy elevado.

En el siguiente capítulo expondremos con más detalle la base teórica de la solución que proponemos para la identificación de insectos, centrándonos principalmente en la Transformada de Fourier y su implementación software.

3. DEFINICIÓN DE LA SOLUCIÓN

En este capítulo expondremos las bases teóricas que utilizaremos en la solución propuesta para este proyecto. Primero, analizaremos la Transformada de Fourier y su utilidad en el ámbito de reconocimiento de frecuencias elementales. Después, veremos brevemente como la librería en C++ que utilizamos para su implementación software (“kiss-fft”) implementa dicho teorema matemático. Finalmente, describiremos la forma como utilizamos esta técnica para diferenciar unos insectos de otros.

3.1. Transformada de Fourier

La transformada de Fourier es una transformación matemática que se usa para traspasar señales desde el dominio del tiempo al dominio de la frecuencia, y viceversa [44]. Aunque su uso está ampliamente extendido en los campos de la física y la ingeniería, su aplicación más popular es para descomponer señales sonoras en sus amplitudes (o tonos) elementales. Por ejemplo, si en un instrumento musical (como el piano) tocamos las notas *sol* y *re* a la vez (cuyo sonido tiene frecuencias distintas), el sonido que producen es una función periódica que combine esas dos notas elementales. La Transformada de Fourier podría ayudarnos a reconocer las notas musicales elementales partiendo del sonido que forman siendo tocadas a la vez, tal como explica el autor de [45].

Abajo, en la imagen 3.1 (extraída de [45]) muestra un ejemplo de un sonido compuesto de 4 notas (o frecuencias) distintas. Dado que la función de entrada es periódica [44], gracias a la Transformada de Fourier es posible identificar esas 4 notas fundamentales que componen el sonido de entrada.

Es importante notar que la Transformada de Fourier es reversible, así que tomando las 4 notas de entrada en el ejemplo de la Imagen 3.1, es posible obtener la frecuencia compuesta por esas 4 notas utilizando la misma técnica.

Por último, cabe remarcar que, como explicamos en el Capítulo 2, estas notas o tonos musicales suelen ser medidos en hercios, por lo que la Transformada de Fourier de una señal compleja puede darnos una serie de amplitudes de onda básicas (medidas en hercios) que componen dicha señal compleja de entrada.

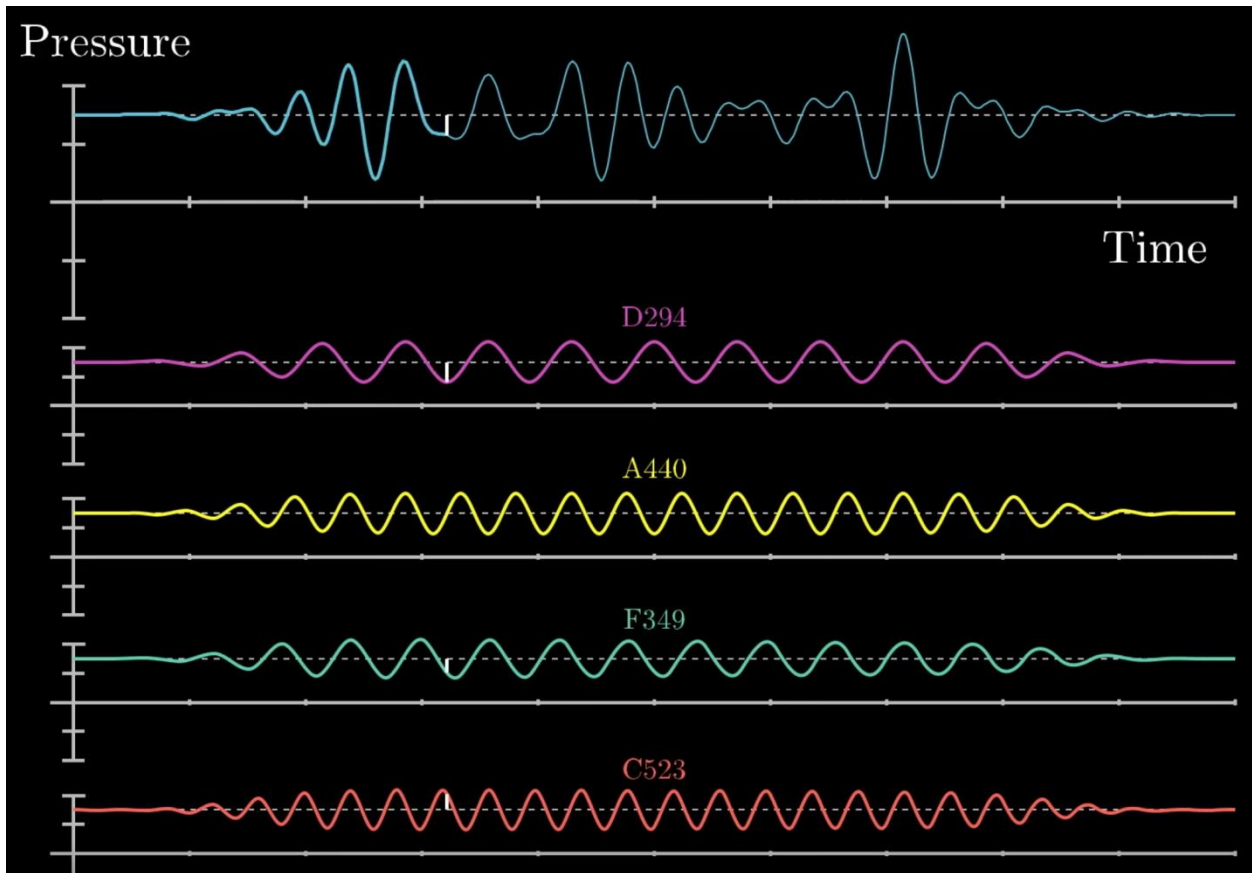


Imagen 3.1. Ejemplo de entrada/salida de la Transformada de Fourier

En la siguiente sección detallamos brevemente la utilización de la librería que implementa esta Transformada de Fourier en C++, la cual hemos usado en nuestro proyecto.

3.2. Implementación rápida de Fourier en software

En esta sección detallamos el uso de la librería que utilizamos para calcular la transformada de Fourier en lenguaje C de forma rápida. Dicha librería es llamada `kiss-fft` [44] por su autor, refiriéndose al principio software de mantener el código y la funcionalidad tan simple como sea posible (de sus siglas en inglés: *keep it simple, stupid!*).

Cabe notar que la librería finalmente seleccionada para este proyecto es la más eficiente que hemos podido encontrar. A modo de comparativa, la segunda librería más eficiente con la que hemos trabajado fue `ardFFT` [45], cuya función `myFFT2` la cual tardaba alrededor 1300 milisegundos para calcular la transformada de Fourier con trama de tamaño

2048 en Arduino DUE (que presentaremos en el siguiente capítulo), comparado con los alrededor de 130 milisegundos que *kiss-fft* tarda en realizar dicho cálculo.

Al utilizar *kiss-fft* incrementó considerablemente los resultados de los tests realizados en Arduino DUE, los cuales se aproximaron mucho más a los realizados primeramente en el entorno de MatLab gracias a que fuimos capaces de procesar más tramas en menor tiempo, incrementando así la precisión del programa de reconocimiento de insectos que describiremos en los capítulos 5 y 6.

El uso de esta librería es bastante sencillo, ya que bastan apenas 3 líneas de código para poder obtener la transformada de Fourier de nuestras tramas:

```
1. kiss_fft_cfg cfg = kiss_fft_alloc(int nfft, int inverse_fft, void *
   mem, size_t * lenmem);
2. kiss_fft_cpx in[N], out[N];
3. kiss_fft(cfg, in, out);
```

En la primera línea de código presentada, inicializamos la configuración de *kiss-fft* (que en realidad es un bloque de memoria continuo que será dedicado a guardar y procesar las tramas), pasando como parámetros:

- `nfft`: el tamaño de las tramas que vamos a tomar,
- `inverse_fft`: para especificar que no queremos realizar la transformada de Fourier inversa pasaremos `false` (o el valor 0 este parámetro sería `true` o 1 si ese fuera el caso)
- `mem` y `lenmem`: Son parámetros internos de la librería que son inicializados en esta función si pasamos sus valores por defecto (null o 0), por lo que los valores que pasemos no son de demasiada importancia. En todos los ejemplos que hemos encontrado estos parámetros relativos a las posiciones de memoria que se van a inicializar suelen inicializarse con sus valores por defecto (0). En la siguiente referencia [46] puede encontrarse el código fuente de esta función en el que se ve que estos dos parámetros son modificados dentro de la misma. Parece ser que estos valores no deberían ser por defecto solamente en el caso de que nosotros previamente hayamos dedicado otras posiciones de memoria para la configuración de *kiss-fft*, lo que no es nuestro caso.

En la segunda línea de código estamos declarando los arrays de entrada y salida. El primero de los arrays se corresponderá con la trama que tomamos directamente del

micrófono (ya calibrado), mientras que en la segunda almacenaremos la transformada de Fourier usando `kiss-fft`. Cabe notar que estos arrays contienen valores del tipo definido en la librería llamado `kiss_fft_cpx`, el cual define un número complejo, compuesto por una parte real y otra imaginaria. Para cada elemento de nuestro array, podemos asignar o acceder a sus partes real e imaginaria usando los atributos `r` e `i`, respectivamente, como podemos ver en el siguiente ejemplo, en el cual definimos el número complejo $4-3i$:

```
kiss_fft_cpx numero_complejo;
numero_complejo.r = 4;
numero_complejo.i = -3;
```

Finalmente, en la última línea de código utilizamos todo lo definido anteriormente: la configuración (bloque de memoria) `cfg`, y los arrays de entrada y salida (`in` y `out`) para llamar a la función `kiss_fft`. Esta función usa el array de entrada `in` (en el que se encontrará nuestra muestra de sonido a procesar), procesa la transformada de Fourier y la almacena en el array de salida `out`, utilizando el bloque de memoria definido en `cfg`.

Tras esta última línea de código podremos utilizar la transformada de Fourier almacenada en `out` para seguir realizando el reconocimiento de insectos en nuestro programa, como veremos en el capítulo 6.

En el siguiente capítulo nos centraremos en la parte hardware de la solución que hemos ideado para este proyecto, describiendo todos los componentes físicos necesarios para su desarrollo. Tras esto, pasaremos a describir los componentes software en los capítulos 5 y 6.

4.HARDWARE NECESARIO

Para el desarrollo de este proyecto es necesario trabajar con algunos elementos que desarrollamos a continuación.

4.1. Arduino UNO

En los comienzos de la implementación del hardware se intentó desarrollar el proyecto con una placa programable de tipo Arduino UNO. Los motivos que nos motivaron a desarrollar el proyecto con este hardware fue su alta compatibilidad con gran cantidad de librerías que facilitarían el desarrollo del software del programa pero en contra encontramos graves inconvenientes que impidieron que fuera una vía acertada la de trabajar con esta opción.

Como hemos descrito en el capítulo 2, Arduino UNO es la placa programable más estándar de toda la gama de Arduino y por ello cuenta con muchas más facilidades para trabajar con ella (librerías, shields, versiones...), pero en contra encontramos que no es tan potente como otras versiones (Imagen 2.4).

Al empezar a desarrollar el software vimos que el principal problema que nos íbamos a encontrar era la memoria de la placa (memoria FLASH de 32 KB y memoria RAM de 2KB), ya que con programas de prueba para el reconocimiento ya nos estaban saltando avisos de que la memoria de la placa estaba casi al 100%. Esto nos empezó a pasar con el reconocimiento de 6 insectos y con un reconocimiento mucho más simple que el que acabamos teniendo ya que la transformada de Fourier que realizábamos era con una ventana de 256 por la memoria de la Arduino UNO. Este tipo de ventana en la transformada de Fourier nos daba como resultado un array de 128 posiciones para basar nuestro reconocimiento por lo que los saltos en frecuencia en el reconocimiento eran demasiado bruscos y para insectos con frecuencias parecidas el programa era incapaz de diferenciarlos.

En el capítulo 2 se puede ver la placa Arduino UNO junto con sus características.

4.2. Arduino DUE

Como solución al problema de la memoria en la placa programable decidimos cambiar el tipo de placa por una con unas características más asociadas a nuestro código. Ya que de esta forma tendríamos un reconocimiento con mayor fiabilidad y un número de insectos a reconocer más amplio.

Con esta nueva placa pudimos ampliar la ventana de la transformada de Fourier a 2048, lo que nos da un array de reconocimiento de 1024 posiciones que nos permite saltar en frecuencia cada 8 Hz, esto hace que se pueda reconocer cualquier insecto por muy similar que sean.

Al ampliar la memoria en la placa programable (memoria FLASH de 512KB y memoria RAM de 96KB) nos ha permitido desarrollar un código más específico en la identificación de insecto y con un mayor número de insectos a reconocer.

En el capítulo 2 se puede ver la placa Arduino DUE junto con sus características

4.3. Micrófono

El micrófono utilizado para el proyecto es un electret MAX4466.

Se trata de un micrófono conectado a un pequeño amplificador con control de ganancia ajustable de 25x a 125x. Ello significa que podemos tener a la salida señales desde los 200mV, lo cual es suficiente para señales de entrada de línea con hasta 1Vpp.

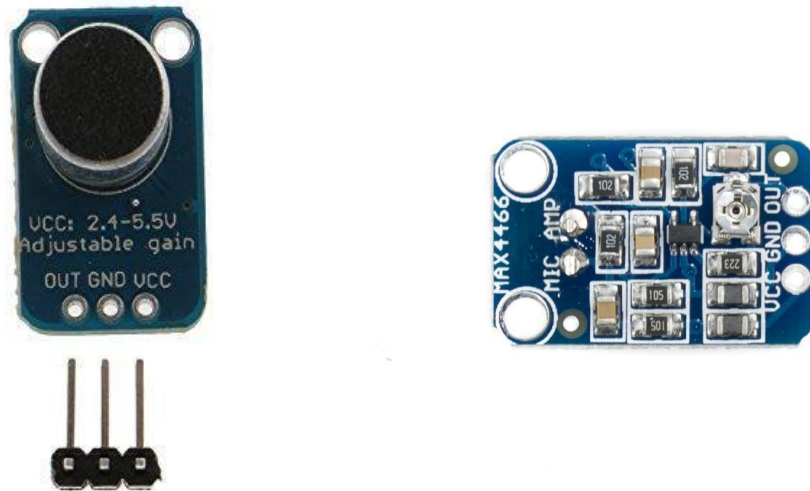


Imagen 4.1. Izquierda, parte frontal del micrófono. Derecha, parte trasera del micrófono.

Las características de este dispositivo se detallan a continuación:

- Alimentación: 2,4V~5V.
- Rango de frecuencia: 20Hz~20KHz.
- Nivel de salida: hasta 5Vp-p.
- Ganancia ajustable: 25x~125x.

5.IMPLEMENTACIÓN DE LA SOLUCIÓN (I): ANALIZANDO INSECTOS EN MATLAB

En este capítulo expondremos la parte dedicada al análisis del reconocimiento con el programa Matlab y la forma en la que hemos desarrollado el código. El objetivo es crear un programa que emule el comportamiento de la placa Arduino para después buscar la mejor solución a implementar en esta.

5.1. Código en Matlab

En este apartado se detalla la explicación del código desarrollado en el programa y el porqué se ha realizado de esa forma. Para una mejor comprensión dividiré el programa en diferentes partes.

5.1.1. *Primera parte: Analizando la señal de entrada*

En primer lugar debemos recoger la señal analógica 'x' y la frecuencia de muestreo 'fs' del archivo de audio del insecto que queremos analizar. Para ello se utiliza la siguiente función:

```
[x,fs]=audioread('Acheta_domesticus.wav');
```

Al hacer esto recibimos la señal analógica en estéreo (2 canales) por lo que la variable 'x' sería una matriz de $2 \times (\text{longitud de la señal})$. De esta forma no es cómodo hacer un análisis de la variable por lo que pasamos la señal a mono (1 canal) de la siguiente forma:

```
x1=x(1,:);  
x2=x(2,:);  
x=(x1+x2)/2;
```

De esta forma nos queda la señal de audio en un solo array.

Si queremos añadir ruido blanco gaussiano a la señal utilizamos la función:

```
x = awgn(x,10);
```

La cual nos devuelve una relación señal/ruido (SNR) de 10 dB en el ejemplo anterior.

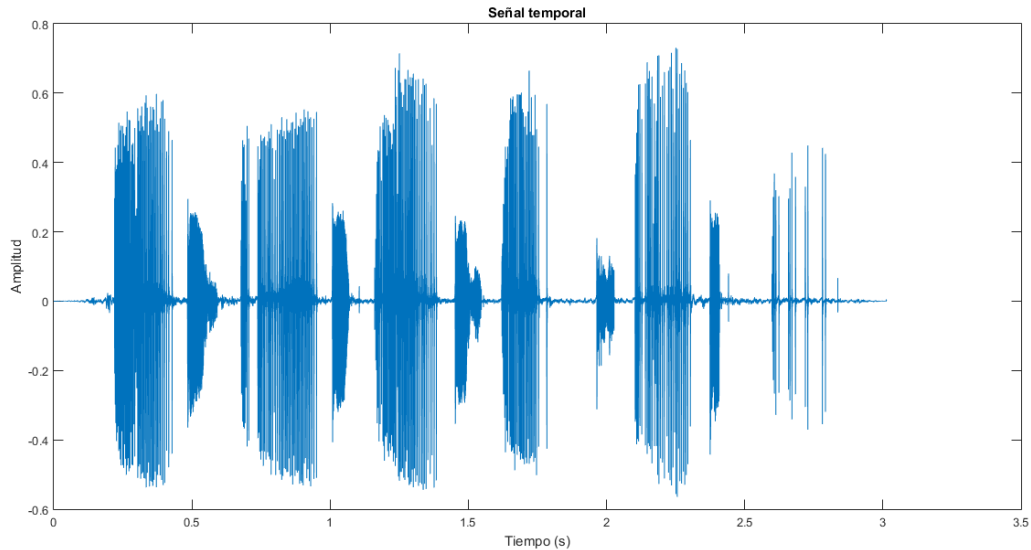


Imagen 5.1. Señal analógica del insecto Acherontia atropos.

Una vez tenemos la variable de la señal analógica pasamos a calcular la frecuencia de dicha señal. Para ello utilizaremos la Transformada de Fourier vista en el capítulo anterior.

5.1.2. Segunda parte: Realizando la transformada de Fourier

Primeramente se deberá obtener la señal en el dominio de la frecuencia, para ello utilizaremos la función de Matlab 'fft'.

Dado que Arduino no es tan potente como para poder hacer una transformada de Fourier de todo el espectro temporal, deberemos dividir el cálculo en ventanas para ello utilizaremos potencias de 2, en nuestro caso, gracias a los test realizados en el capítulo 7, se ha llegado a la conclusión de que la mejor opción es realizar la transformada de Fourier con una ventana de 2048 y una frecuencia de muestreo de la señal de 24000 Hz.

```
X = fft(x,NFFT);  
f = fs/2*linspace(0,1,NFFT/2+1);
```

Ahora en la variable 'X' tenemos la señal frecuencial de la cual obtendremos la posición del valor más representativo, es decir, la frecuencia de la señal.

```
[Amplitud,f_max] = max(abs(X));  
f0=f(f_max);
```

En 'f0' guardamos la frecuencia del valor máximo de la trama y en 'Amplitud' guardamos su amplitud.

```
Amplitud_log=20*log10(Amplitud);
```

En 'Amplitud_log' guardaremos los dBv del valor máximo de la trama y de esta forma podremos aplicar una puerta de ruido, en la que los valores que nos sean lo suficientemente representativos queden omitidos.

Por cada valor que vamos obteniendo vamos decidiendo qué tipo de insecto es. Esto se hace agrupando los valores de frecuencia máxima de 10 tramas y calculando la mediana, para evitar que alguna trama haya sido muy poco representativa y corrompa la media.

Cuando sabemos la mediana de la señal nos centramos en el tipo de insecto que es:

Si la mediana está entre 120 y 400 Hz, nos centraremos en insectos de baja frecuencia y sólo recogeremos las frecuencias máximas de las tramas que estén dentro de este rango. Lo mismo pasará con los insectos de tipo grillo (4400 ~ 5200 Hz), de tipo cigarra (5200 ~ 6600) y para insectos de alta frecuencia (6600 ~ 8000 Hz).

Si el insecto no corresponde a ninguno de estos rangos también será analizado, aunque no se discriminaría la frecuencia en caso de que exista alguna trama con una frecuencia máxima muy dispar.

5.1.3. Tercera parte: Histograma y número de tramas

Por cada trama analizada de la señal de entrada se obtendrá una frecuencia máxima y si su amplitud supera la puerta de ruido, la almacenaremos en un array 'y'.

Cuando toda la señal quede analizada pasamos a representar un histograma para ver qué aspecto tiene el conjunto de nuestros valores imagen 4.2.

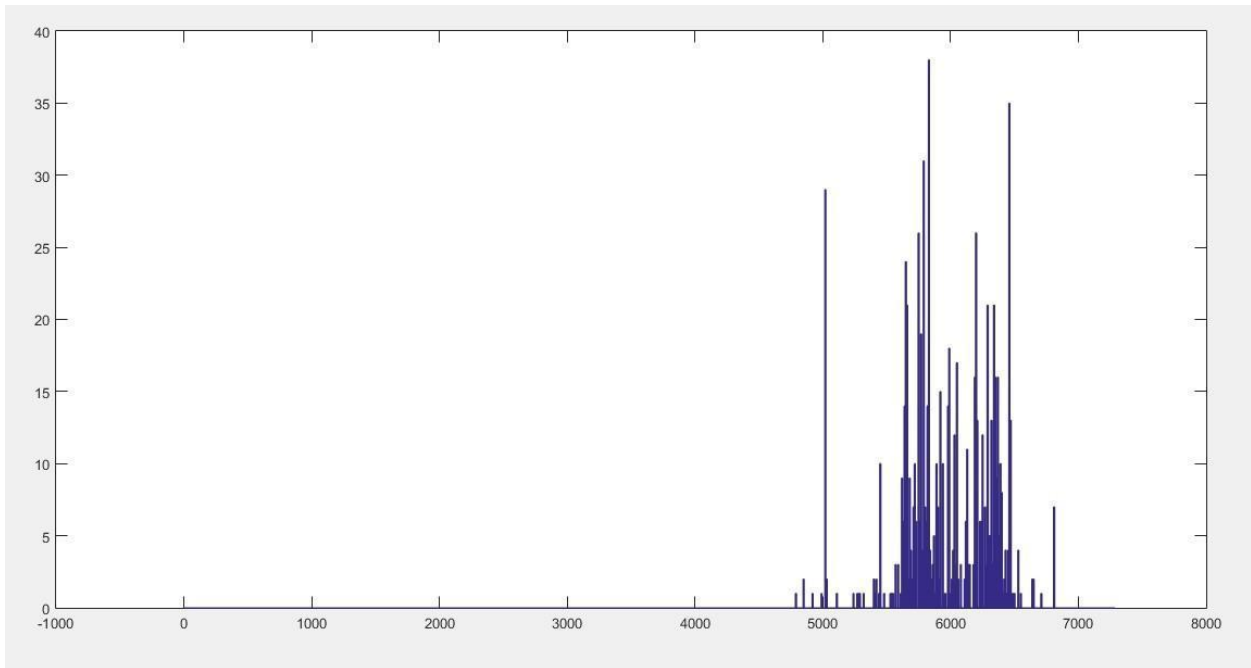


Imagen 5.2. Histograma del insecto Acherontia atropos

Una vez hecho esto, recortamos la señal 'y' para emular lo que sería nuestra placa Arduino, ya que en nuestra placa no analizamos la señal de entrada de esta forma, sino que recogemos un número de tramas estipulado hasta llegar a una decisión. Este número de tramas lo variamos en los test del capítulo 7 para poder saber qué número es el más efectivo.

5.1.4. Cuarta parte: Obteniendo los valores del reconocimiento

Pasamos a calcular los valores que utilizaremos para el reconocimiento de nuestra señal de entrada. Estos valores son la media y la desviación típica. Una vez hallado estos dos valores pasamos a representar la campana de Gauss, Imagen 4.3.

La campana se la señal de entrada se muestra junto a las campanas de Gauss de los insectos de su mismo tipo, los cuales hemos definido anteriormente. De esta forma podemos analizar el posible desplazamiento que haya podido sufrir nuestra señal de entrada o si ha aumentado el ancho de la campana debido al ruido añadido o a algún parámetro modificado.

Esto ha sido muy útil para poder decidirse por qué valores decantarse en la placa Arduino ya que se ha podido observar de qué forma le ha afectado una modificación a la representación original del insecto. Se ha creado un anexo que contiene algunas gráficas de las campanas de Gauss con sus modificaciones de parámetros de los test realizados en el capítulo 7.

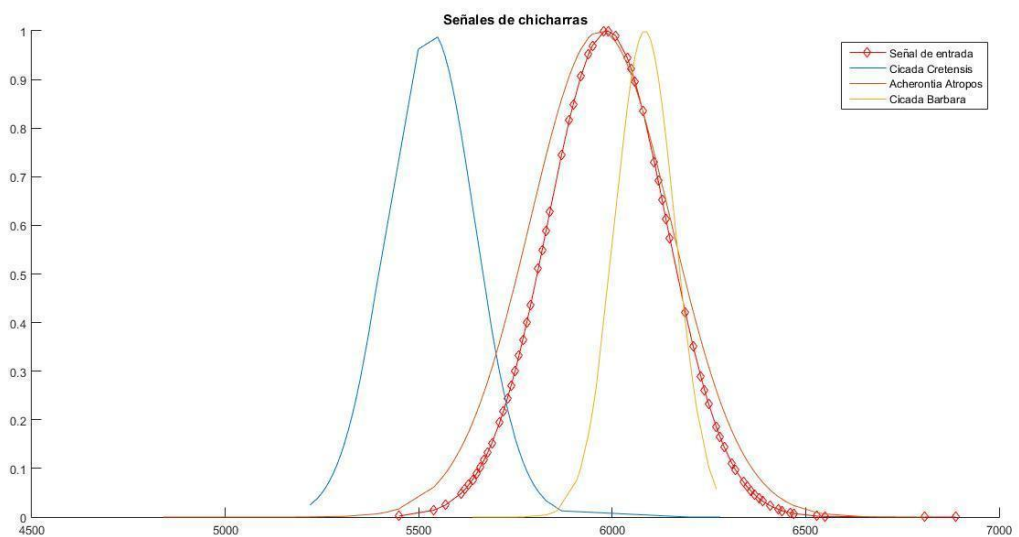


Imagen 5.3. Funciones gaussianas de las chicharras y de la señal de entrada.

5.1.5. Quinta parte: Identificación del insecto

Una vez tenemos los valores de media y desviación típica de nuestras tramas pasamos a compararlos con los del resto de insectos. Para ello se ha utilizado una fórmula y unos porcentajes de similitud que se explican de mejor forma en el subapartado “Tests de Optimización de Parámetros para el Reconocimiento de Insectos” hallado dentro del capítulo 7 de “Resultados”.

Cuando es analizada la similitud de cada insecto, se almacenan en un array en el cual se obtendrá el máximo para saber el insecto que ha sido más parecido con su porcentaje de similitud y su valor mínimo en el caso de que se quiera saber la función de similitud.

Al obtener el valor máximo, tenemos en cuenta también la posición en el array del valor para después mirar en la función de Matlab “insectos” a qué insecto pertenece dicha posición.

Al finalizar mostramos la diferencia de media y desviación típica de cada insecto, así como su porcentaje de similitud:

```
Eumodicogryllus Burdigalensis
La media tiene un valor de similitud de: 4.779410
La desviación típica tiene un valor de similitud de: 42.781046
PORCENTAJE TOTAL DE SIMILITUD: 78.731536
FUNCIÓN DE SIMILITUD: 204.468165
```

```
-----
Gryllotalpa Gryllotalpa
La media tiene un valor de similitud de: 2703.883545
La desviación típica tiene un valor de similitud de: 85.683873
PORCENTAJE TOTAL DE SIMILITUD: 0.000000
FUNCIÓN DE SIMILITUD: 231679.213302
```

```
-----
Gryllus Campestris
La media tiene un valor de similitud de: 13.548086
La desviación típica tiene un valor de similitud de: 72.593695
PORCENTAJE TOTAL DE SIMILITUD: 62.948344
FUNCIÓN DE SIMILITUD: 983.505597
```

Para valores en el porcentaje de similitud inferiores a 50 se toma la elección de insecto no reconocido ya que consideramos que la comparativa de insectos ha sido muy diferente.

5.2. Extrayendo el perfil de cada insecto

Para obtener la información que deseamos de cada insecto, guardamos previamente los valores de los insectos en archivos de extensión “.mat”.

Estos archivos se guardan a través de la siguiente función de Matlab:

```
save('Acherontia_atropos.mat','y');
```

De esta forma guardamos en un archivo la señal “y”, la cual explicamos en el apartado anterior que corresponde a la señal en el dominio de la frecuencia, en un archivo con el nombre “Acherontia_atropos.mat”.

Cuando se desee obtener la señal en el dominio de la frecuencia de esta señal, llamaremos desde Matlab a este archivo de la siguiente forma.

```
Acherontia_atropos=load('Acherontia_atropos.mat');  
y=Acherontia_atropos.y;
```

De esta forma recuperamos la señal “y” de todos los insectos que tengamos almacenados en archivos y llamando a la función “preparada” obtenemos los valores de: media de la frecuencia, desviación típica y función gaussiana de cada insecto.

```
[y1,med1,des1,gs1]=preparada(y);
```

Por lo que cada insecto tendrá asociada una media de la frecuencia, una desviación típica y una variable gaussiana que solo empleamos para su representación gráfica de la Imagen 4.3.

Con la media y la desviación típica de cada insecto se realiza la comparación que detallamos en el apartado anterior en su quinta parte.

6.IMPLEMENTACIÓN DE LA SOLUCIÓN (II): RECONOCIENDO INSECTOS EN ARDUINO

Este capítulo detalla el código fuente de la aplicación desarrollada y explica los aspectos cruciales del mismo, divididos en 4 bloques: estructura principal del programa Arduino, cómo el sonido es captado por el programa desde el micrófono, cómo la trama de sonido es procesada para poder trabajar con ella, y finalmente cómo se comparan las tramas analizadas con los insectos conocidos por el programa para su identificación.

6.1. Parte 1: Estructura principal

Como se mencionó en el Capítulo 2 en la introducción al lenguaje Arduino, la estructura principal de un programa en Arduino está compuesta de dos funciones: *setup* and *loop*. En esta estructura también incluimos la cabecera de nuestra aplicación, la cual (1) importa las librerías que serán utilizadas, que todas forman parte del núcleo estándar de C a parte de “kiss-fft”, que fue descrita en el capítulo 3, (2) e inicializa algunas constantes y variables globales, que son descritas en comentarios en el código fuente detallado abajo. Las constantes y variables globales especificadas en esta parte del código incluyen comentarios explicando a lo que se refieren.

En la función *setup* inicializamos los parámetros necesarios para la entrada de datos analógicos del micrófono. En la función *loop* se realizan dos tareas: (1) inicializamos los parámetros para cada trama, y (2) bucle *while* que itera indefinidamente intentando reconocer los insectos que se escuchan por el micrófono.

(1) Los parámetros inicializados para cada trama dentro de la función *loop* son los siguientes:

- `tramas[NT]` : Lista donde añadiremos las tramas analizadas hasta el momento (inicialmente vacía)
- `struct insecto insectos[num_insectos]`: Lista con todos los insectos y su información. El tipo “insecto” se detalla en la sección “Parte 4: Reconociendo insecto”.
- `rellenar_info_insectos(insectos)` : Llamada a la función “rellenar info insectos”, la cual rellena el array “insectos” con los insectos y su información para ser reconocidos.

- `num_tramas`: Inicializamos el número de tramas analizadas hasta el momento, que al ser el principio de la ejecución es igual a cero.
- `contador_reset_tipo`: Número de tramas fuera de la ventana de focalización (para cada tipo de insecto) que han sido tomadas de forma seguida hasta ahora. Si este contador llega a `max_contador_reset_tipo`, el reconocimiento de insecto se reiniciará. Por ejemplo, si estamos en el análisis de la trama número 15 y el programa de reconocimiento está focalizado en la ventana de grillos, y tras esto se suceden 10 tramas seguidas que no se corresponden con dicha ventana (porque por ejemplo esté sonando una cigarra), el programa se re-inicializará y probablemente se centrará en la ventana de cigarras. Si por el contrario solamente se sucedieran 7 tramas seguidas fuera de la ventana de grillos y la octava quedara dentro, este contador se inicializaría a 0.
- `max_contador_reset_tipo`: Como se ha explicado anteriormente, es el número máximo de tramas seguidas que el programa permite tomar fuera de la ventana esperada de insecto antes de re-inicializar el reconocimiento.

(2) El bucle `while` que itera indefinidamente ejecuta las siguientes tres operaciones principales: tomar trama, procesar la trama y reconocer insecto. Estas 3 operaciones principales se corresponden con los tres apartados de código fuente que detallamos tras esta sección. Tras esas tres tareas, lo único adicional que se ejecuta es el controlador para re-inicializar el reconocimiento y los parámetros por defecto en caso de que un número de tramas seguidas queden fuera de los valores esperados.

Finalmente, cuando el programa ejecuta este bucle `while` `NT` veces, resetea el número de tramas analizadas hasta el momento a cero (en `num_tramas = (num_tramas + 1) % NT;`) para comenzar el reconocimiento de insecto de nuevo.

```
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include <string.h>
#include <stdio.h>
#include <math.h>
```

```

#include "kiss_fft.h"

#define mic_adc A0 // Pin # del microfono ADC
#define N 2048 // Tamaño de la trama
#define Fs 24000 // Frecuencia de la muestra

// Inicializacion de la configuracion de kiss-fft: Tamaño de ventana es
2048 y lo // demás todo por defecto (no es transformada de Fourier inversa)
kiss_fft_cfg cfg = kiss_fft_alloc(N, false, 0, 0);
// Declaración de los arrays de entrada (in) y salida (out) para la FFT
kiss_fft_cpx in[N], out[N];

// Array donde se guarda la onda de sonido capturada desde el micrófono
double onda_sonido[N];

// Num. tramas que tomaremos en total para reconocer cada insecto
int NT = 30;
// Num. de especies de insectos a reconocer
int num_insectos = 14;

// aperturas de los intervalos de reconocimiento por defecto (para la
función de // similitud, ver la sección de reconocimiento de insecto más
adelante
int apertura_media = 2000;
int apertura_desv = 500;
int muestra_minima = 150;
int muestra_maxima = 8400;

... // Código de otras funciones del programa: muestreo, procesamiento y
reconocimiento de insectos

//*****FUNCIONES BASE DE
ARDUINO*****
void setup(){
    Serial.begin(115200);
    analogReadResolution(12);
}

void loop(){
    int maxFrec;

```

```

// Variables de inicializacion para reconocimiento de insectos
int tramas[NT]; // Array con todas las tramas tomadas hasta ahora
struct insecto insectos[num_insectos];
rellenar_info_insectos(insectos); // Rellenamos los parametros para
reconocimiento de insectos

```

```

int num_tramas = 0; // Num tramas tomadas hasta ahora
int contador_reset_tipo = 0;
int max_contador_reset_tipo = 10;

```

```

while (1){
    tomar_trama(onda_sonido, in);
    kiss_fft(cfg, in, out);
    // Post-procesamiento de la transformada
    maxFrec = obtenerFrecuenciaMasAlta(out);
    if(maxFrec >= muestra_minima && maxFrec <= muestra_maxima){
        Serial.print((char*)"Valor de la trama: ");
        Serial.println(maxFrec);
        contador_reset_tipo = 0;
        tramas[num_tramas] = maxFrec;
        num_tramas = (num_tramas + 1);
        if(num_tramas > 1){
            reconocer_insecto(tramas, num_tramas, insectos, num_insectos);
        }
        Serial.println(num_tramas);
    }else{
        contador_reset_tipo = contador_reset_tipo + 1;
        if(contador_reset_tipo == max_contador_reset_tipo){
            Serial.println((char*)"Reiniciando el reconocimiento...");
            num_tramas = 0;
            contador_reset_tipo = 0;
            apertura_media = 2000;
            apertura_desv = 500;
            muestra_minima = 200;
            muestra_maxima = 8400;
        }
    }
    if(num_tramas == NT){
        Serial.println((char*)"Reconocimiento de insecto finalizado!");
        num_tramas = 0;
    }
}

```

```

    apertura_media = 2000;
    apertura_desv = 500;
    muestra_minima = 200;
    muestra_maxima = 8400;
    delay(5000);
};
}
}

```

6.2. Parte 2: Muestreando sonido

La función que hemos implementado para muestrear sonido es relativamente sencilla. Simplemente llamamos a la función de Arduino “analogRead” pasándole el pin donde está conectado nuestro micrófono un número N de veces para así tomar N muestras. Recordamos que N es una constante inicializada al comienzo del programa y cuyo valor es 2048 (teniendo así una trama de mayor tamaño comparado con lo que podíamos hacer en Arduino UNO).

```

inline void tomar_trama (double *onda_sonido, kiss_fft_cpx *in){
    double espera_entre_tramas = (1000000.0 / Fs);
    int inicio = micros();
    for (int i = 0; i < N; i++){
        while (micros() - inicio < espera_entre_tramas);
        onda_sonido[i] = analogRead(mic_adc);
        inicio = micros();
    }
    for (int i = 0; i < N; i++){
        in[i].r = onda_sonido[i];
        in[i].i = 0;
    }
}

```

El valor del parámetro `in` es modificado por la función cada vez que es llamada. Este parámetro es un array que contiene elementos del tipo `kiss_fft_cpx`. Este tipo define un número complejo, compuesto por una parte real y una imaginaria. En el último bucle de esta función rellenamos los valores reales y complejos de cada uno de los valores del array `in`. En nuestro caso, los valores imaginarios no son necesarios, ya que realizaremos una transformada solamente con números reales, por lo que la parte imaginaria

de los números complejos la inicializamos a 0 en todos los casos. La parte real se corresponde con la onda de sonido que el micrófono ha capturado.

Dicha variable `in`, ya con el valor de la trama tomada, será la que pasemos a la función `kiss_fft(cfg, in, out)` de la librería *kiss-fft* para ser procesada. En la siguiente sección explicaremos cómo realizamos ésta operación y cómo procesamos esta trama para obtener su frecuencia máxima en hercios.

6.3. Parte 3: Procesando la trama

La trama es procesada en la función `loop` de la estructura principal detallada anteriormente en esta sección. Dicho procesamiento consiste en dos partes principales: (1) extraer las frecuencias fundamentales de la trama mediante la Transformada de Fourier, usando la librería *kiss-fft* y (2) obtener la frecuencia máxima de entre estas para usarla en la identificación del insecto que está siendo escuchado en la trama. Estos dos pasos son realizados por las siguientes dos líneas de código dentro de la función `loop` especificada en la Parte 1 de esta sección:

```
kiss_fft(cfg, in, out); // Transformada Fourier rapida sobre la trama
maxFrec = obtenerFrecuenciaMasAlta(out); // Post-procesamiento de la
transformada
```

La primera línea hace la llamada a la función `kiss_fft(cfg, in, out)` de la librería *kiss_fft*, pasando como uno de los parámetros la trama a analizar `in` que recolectamos en la sección anterior, y obteniendo como resultado la transformada en `out`, que también se pasa como parámetro.

Tras esta primera parte, la trama analizada `out` se pasa a nuestra función `obtenerFrecuenciaMasAlta`, que simplemente recorre los espacios relevantes del array `out` y devuelve la frecuencia más alta.

```
inline double obtenerFrecuenciaMasAlta(kiss_fft_cpx *out){
    double maxk = -1.0;
    double maxvalue = 30;
    for (int i = N/2; i < N; i++){
        double frecuencia = abs((i-N)*Fs*.791/N); // Frecuencia en i
```

```

    double temp = sqrt(out[i].r * out[i].r + out[i].i * out[i].i);
    if (temp > maxvalue){
        maxk = frecuencia;
        maxvalue = temp;
    }
}
return maxk;
}

```

En esta función también analizamos la frecuencia máxima de la trama así como la amplitud de la trama escogida en dBv, usando la frecuencia de muestreo F_s y el número total de muestras N . Gracias a los tests previos de calibración que realizamos encontramos un ligero desajuste ocasionado por el micrófono (por ejemplo, su conexión) o sensibilidad. Analizando este desajuste en todo el rango de frecuencias utilizado por los insectos analizados, encontramos que el error es un factor constante de 0.791 por lo que introducimos este valor constante en el cálculo de la frecuencia máxima.

Otro punto a tener en cuenta en esta función es que el valor mínimo de amplitud (también llamado puerta de ruido) que reconoceremos es de 30 dBv, por lo que de esta forma descartamos cualquier inferencia de ruido de fondo que pueda haber. Este parámetro fue introducido con el fin de minimizar el ruido de fondo que constantemente distorsionaba el análisis durante los tests que presentaremos en el siguiente capítulo. Introduciendo esta puerta de ruido, la precisión en el reconocimiento tanto en MatLab como en Arduino incrementó considerablemente.

Finalmente, la función devuelve la frecuencia máxima de la trama analizada que es la que nos interesa para el reconocimiento del insecto.

6.4. Parte 4: Reconociendo insecto

Esta parte del programa en principio parece la más compleja. Sin embargo, el haberla dividido en varias funciones sencillas hace más fácil su explicación. Este “módulo” puede dividirse en tres partes principales:

1. Inicializando insectos: Definición e inicialización de los insectos a reconocer
2. Análisis de tramas: Cálculo de la similitud entre estos insectos y los valores obtenidos

3. Ventanas y similitud: Selección de una ventana de frecuencias para reconocer insectos dentro de un tipo general

Estas tres partes se corresponden con las tres sub-secciones que vienen a continuación describiendo cada una de las características del módulo de reconocimiento dentro del programa.

6.4.1. *Inicializando insectos*

En esta parte del código se declara el tipo “insecto”, el cual se compone de 4 atributos:

- nombre: Es el nombre del insecto, de hasta 22 caracteres. Para mostrar todos por pantalla junto con las similitudes calculadas de forma adecuada, los 22 caracteres son rellenados con puntos
- frecuencia media: Es la media de las frecuencias detectadas para este insecto
- frecuencia desviación estándar: Es la desviación estándar de las frecuencias detectadas para este insecto
- tipo: Es un entero que define la especie o tipo de frecuencia del insecto y que utilizamos para focalizar la apertura de la ventana de frecuencias que tomamos (las frecuencias que quedan fuera de esta ventana son descartadas como vimos en la función principal de Arduino *loop*). Este tipo puede tomar 5 valores:
 - 0: si es un grillo
 - 1: si es una cigarra
 - 2: si es un insecto de baja frecuencia
 - 3: si es un insecto de alta frecuencia
 - 4: si es otro tipo que no entra en ninguno de los rangos anteriores

Seguidamente, en esta parte del código se declara la función `rellenar_info_insectos`, la cual ya vimos su funcionamiento cuando explicamos el bucle principal de Arduino en la Parte 1 de este capítulo. Las medias y desviaciones estándar para cada insecto son las que han sido extraídas del procesamiento en MatLab explicado en el capítulo anterior.

```
//*****FUNCIONES DE RECONOCIMIENTO DE INSECTOS*****  
  
struct insecto{  
    char nombre[22];
```

```

float frecuencia_media;
float frecuencia_devst;
int tipo;
};

void rellenar_info_insectos(struct insecto *insectos){
    struct insecto acherontia_atropos={"acherontia atropos...", 5912.40,
234.19, 1};
    struct insecto acheta_domesticus= {"acheta domesticus...", 4622.60,
35.50, 0};
    struct insecto acrididae = {"acrididae (langosta).", 7303.50,
88.45, 3};
    struct insecto anthophila = {"anthophila (abeja)...", 189.31,
20.33, 2};
    struct insecto bombus = {"bombus (abejorro)...", 119.92,
5.61, 2};
    struct insecto cicada_barbara = {"cicada barbara.....", 6094.00,
151.23, 1};
    struct insecto cicada_cretensis = {"cicada cretensis.....", 5536.30,
243.91, 1};
    struct insecto culicidae = {"culicidae (mosquito).", 287.80,
25.05, 2};
    struct insecto eumodicogryllus = {"eumodicogryllus.....", 4879.00,
55.55, 0};
    struct insecto gryllotalpa = {"gryllotalpa.....", 2179.90,
30.47, 4};
    struct insecto gryllus_campestris={"gryllus campestris...", 4869.80,
41.37, 0};
    struct insecto muscidae = {"muscidae (mosca).....", 521.95,
9.46, 4};
    struct insecto vespula = {"vespula germanica....", 246.18,
21.97, 2};
    struct insecto saltamontes = {"saltamontes.....", 7719.90,
133.39, 3};

    insectos[0] = acherontia_atropos;
    insectos[1] = acheta_domesticus;
    insectos[2] = acrididae;
    insectos[3] = anthophila;
    insectos[4] = bombus;
}

```



```

insectos[5] = cicada_barbara;
insectos[6] = cicada_cretensis;
insectos[7] = culicidae;
insectos[8] = eumodicogryllus;
insectos[9] = gryllotalpa;
insectos[10] = gryllus_campestris;
insectos[11] = muscidae;
insectos[12] = vespula;
insectos[13] = saltamontes;
return;
}

```

6.4.2. *Análisis de tramas*

Esta parte contiene la que sería la “función principal” de nuestro módulo de reconocimiento: `reconocer_insecto`. Dicha función toma 4 parámetros: el número y frecuencia de las tramas tomadas hasta ahora y el número y frecuencias medias de los insectos a reconocer. El cuerpo de la función prácticamente se explica por sí solo. Los siguientes puntos lo explican línea a línea:

- Primero calculamos la media de las frecuencias máximas tomadas hasta ahora, usando la función `calcular_media` detallada debajo.
- Seguidamente, calculamos su desviación estándar, usando la función `calcular_desv`, también detallada más abajo.
- Después, ordenamos el array de tramas usando la función estándar de C “`qsort`”, para así calcular su mediana utilizando la función `calcular_mediana`, también detallada en el código debajo. Como hemos visto anteriormente, la mediana se usa como valor más fiable para seleccionar el tipo de insecto y focalizar la ventana de frecuencias para el reconocimiento ya que la media puede desviarse bastante si una o dos frecuencias analizadas son erróneas o con mucho ruido son tomadas al principio de la ejecución del reconocimiento.
- Por último, llamamos a la función `mostrar_insectos_mas_parecidos` para focalizar la ventana de frecuencias, calcular el insecto más parecido o finalizar el reconocimiento mostrando por pantalla, detallada en la sección anterior.

```

void reconocer_insecto(int *tramas, int num_tramas, struct insecto
*insectos, int num_insectos){
    float media = calcular_media(tramas, num_tramas);

```

```

float devst = calcular_desv(tramas, num_tramas, media);
qsort(tramas, num_tramas, sizeof(int), compare_int );
int mediana = calcular_mediana(tramas, num_tramas);
mostrar_insectos_mas_parecidos(num_tramas, media, devst, mediana,
insectos, num_insectos);
}

```

```

float calcular_media(int *tramas, int num_tramas){
    int suma = 0;
    for(int i=0; i<num_tramas; i++){
        suma = suma + tramas[i];
    }
    return (float)suma/num_tramas;
}

```

```

int compare_int( const void* a, const void* b )
{
    if( *(int*)a == *(int*)b ) return 0;
    return *(int*)a < *(int*)b ? -1 : 1;
}

```

```

float calcular_desv (int *tramas, int num_tramas, float media){
    int suma_cuadrados = 0;
    for(int i=0; i<num_tramas; i++){
        float diff = tramas[i] - media;
        suma_cuadrados = suma_cuadrados + (diff * diff);
    }
    suma_cuadrados = (float) suma_cuadrados/(num_tramas - 1);
    int desvst = sqrt(suma_cuadrados);
    if (isnan(desvst)){ // Comprobamos si desvst es NaN
        return 0;
    }
    return desvst;
}

```

```

float calcular_mediana (int *tramas, int num_tramas){
    if (num_tramas<=3){
        return tramas[num_tramas -1];
    }
    if (num_tramas% 2 == 0){

```

```

    return (tramas[(num_tramas/2)-1] + tramas[(num_tramas /2)])/2;
}else{
    return tramas[num_tramas /2];
}
}

```

6.4.3. Ventanas y similitud

Una vez que la media, desviación estándar y mediana de las tramas tomadas hasta ahora han sido calculadas, el programa llama a la función

`mostrar_insectos_mas_parecidos`, la cual realiza 3 tareas:

1. Si el número de tramas tomadas hasta ahora es menor que 5, utiliza la mediana para seleccionar la ventana de frecuencias dependiendo del valor de la misma. En esta parte del código, correspondiente en esta función con el `else` tras el primer `if` que comprueba si el número de tramas tomadas hasta ahora es mayor que 5, podemos ver que hay cuatro casos que se corresponden con las 4 ventanas de frecuencias utilizadas por el programa (a parte de la ventana por defecto que se inicializa al principio y tras cada reconocimiento de insecto como hemos visto en la Parte 1 de este capítulo). Dependiendo del valor de la mediana, el programa se centrará en una ventana o en otra, o en ninguna (ya que hay insectos como el *Gryllotalpa* o la *Mordoganensis* que no se corresponden con ninguna ventana como detalla el código que inicializa los insectos detallado anteriormente). En cada caso imprimimos por pantalla la ventana en la que nos hemos focalizado para que el usuario pueda notar este cambio de parámetros de la ventana por pantalla.
2. Si por el contrario el número de tramas tomadas hasta ahora es mayor que 5, ésta función recorrerá el array de insectos para calcular la similitud de cada insecto con las tramas analizadas hasta el momento (representadas por la media y la desviación estándar). Una vez calculadas las 14 similitudes (una por cada insecto), almacenamos el insecto con la similitud más alta. Si un insecto obtiene una similitud mayor del 50% será mostrado por pantalla. La función `calcular_similitud` se detalla a continuación después del código fuente de la función `mostrar_insectos_mas_parecidos`.
3. Finalmente, si el número de tramas analizadas es igual a `NT` (que es el número de tramas que tomaremos para dar por concluido el reconocimiento, y que definimos al

principio del programa, en la Parte 1 de este capítulo), simplemente imprimiremos por pantalla los detalles del insecto con la similitud más alta calculada anteriormente para mostrárselos al usuario. Si llegado este punto ningún insecto consiguiera obtener una similitud mayor de 0, el programa simplemente imprime esta eventualidad por pantalla y reinicia el reconocimiento de nuevo.

Es importante señalar el valor añadido y complejidad de que el reconocimiento de insectos se haga de forma dinámica. Es decir, en cada trama que se toma el programa calcula la similitud y la imprime por pantalla en vez de simplemente calcular la similitud una vez al final de la toma del número de tramas estipulado. Gracias a esta cualidad es posible observar cómo normalmente la similitud y por lo tanto confianza en el reconocimiento va aumentando conforme el número de tramas analizadas aumenta (hasta llegar a NT, punto en el que el reconocimiento se da por concluido).

```
void mostrar_insectos_mas_parecidos(int num_tramas, float media, float
devst, int mediana, struct insecto *insectos, int num_insectos){
    float max_similitud = -1;
    struct insecto insecto_reconocido;
    if (num_tramas > 5){
        for(int i=0; i<num_insectos; i++){
            float similitud = calcular_similitud(insectos[i], media, devst);
            if(similitud > 50){
                Serial.print(insectos[i].nombre);
                Serial.print(similitud);
                Serial.println("%");
            }
            if(similitud > max_similitud){
                max_similitud =similitud;
                insecto_reconocido = insectos[i];
            }
        }
    }else{
        if(mediana > 4400 && mediana < 5200){
            Serial.println((char*)"La funcion de similitud e intervalo de
muestreo se han centrado en GRILLOS");
            apertura_media = 500;
            apertura_desv = 100;
            muestra_maxima = 5200;
            muestra_minima = 4400;
```

```

    }
    if(media >= 5200 && media < 6400){
        Serial.println((char*)"La funcion de similitud e intervalo de
muestreo se han centrado en CIGARRAS");
        apertura_media = 1000;
        apertura_desv = 500;
        muestra_maxima = 6400;
        muestra_minima = 5280;
    }
    if(media >= 120 && media < 400){
        Serial.println((char*)"La funcion de similitud e intervalo de
muestreo se han centrado en INSECTO DE BAJA FRECUENCIA");
        apertura_media = 300;
        apertura_desv = 40;
        muestra_maxima = 380;
        muestra_minima = 150;
    }
    if(media >= 6400 && media < 8000){
        Serial.println((char*)"La funcion de similitud e intervalo de
muestreo se han centrado en INSECTO DE ALTA FRECUENCIA");
        apertura_media = 1500;
        apertura_desv = 600;
        muestra_maxima = 8100;
        muestra_minima = 5800;
    }
}
if(num_tramas == NT){
    if(max_similitud > 0){
        Serial.print((char*)"INSECTO RECONOCIDO: ");
        Serial.print(insecto_reconocido.nombre);
        Serial.print((char*)"certeza: ");
        Serial.print(max_similitud);
        Serial.println("%");
    }
    else{
        Serial.println((char*)"Ningun insecto almacenado cumple con el perfil
escuchado. Reiniciando escucha...");
    }
}
Serial.println("*****");

```

```

    }

    float calcular_similitud(struct insecto insecto, float media, float
desvst){
        float diff = 50*fabs(insecto.frecuencia_media -
media)/(apertura_media);
        diff = diff + 50*fabs(insecto.frecuencia_devst -
desvst)/(apertura_desv);

        float similitud = 100.0 - diff;
        if(similitud < 0){
            similitud = 0;
        }
        return similitud;
    }

```

Cabe destacar la función `calcular_similitud`, detallada en el código fuente arriba y que se sitúa en el centro del reconocimiento. Esta función toma 3 parámetros: el insecto para el cual queremos calcular la similitud y la media y desviación estándar de las tramas tomadas hasta ahora. La función se basa en la diferencia entre la media y la desviación estándar del insecto y de las tramas. Utilizando los parámetros (seleccionados para cada tipo de insecto junto con la ventana de frecuencias como vimos en el código de la sección anterior) `apertura_media` y `apertura_desv` pretendemos normalizar en el intervalo de 0 a 1 el valor de estas diferencias. Es importante notar que estos parámetros serán modificados dependiendo del rango de frecuencias del insecto que se haya reconocido tras tomar las 5 primeras muestras y dependiendo la mediana, lo que permitirá al sistema “**afinar el oído**” para diferenciar a insectos dentro de un rango de forma más fina, reduciendo estos valores que son los que dividen a las diferencias, permitiendo de esta forma al sistema obtener porcentajes de similitud más dispersos entre insectos del mismo rango de frecuencias. Después, las diferencias de la media y la desviación estándar son añadidas en forma de ecuación lineal, dando igual importancia a ambas para obtener una diferencia total. Nótese que al multiplicarlas por 50 estamos convirtiendo la diferencia en porcentaje. Podrían haberse dado otros valores para priorizar la media sobre la desviación estándar (por ejemplo 90 y 10). Habiendo implementado el programa de esta forma dejamos la puerta abierta para futuro trabajo en este área, pudiendo optimizar estos dos parámetros de la ecuación lineal.

Por último, definimos la similitud como la inversa de la diferencia total calculada, por lo tanto la restamos del porcentaje total 100. Si por algún motivo esta similitud fuera negativa, la reseteamos a cero.

7.RESULTADOS

En este capítulo explicamos los tests que realizamos sobre el sistema para entender su eficacia y precisión al reconocer insectos y detallamos los resultados obtenidos. Además, explicaremos los tests previos realizados con el simulador del sistema final que desarrollamos en MatLab y que explicamos en el capítulo 5. Los resultados de esos tests previos fueron los parámetros y metodologías que mejor precisión dieron en el reconocimiento y que por lo tanto luego utilizamos para desarrollar el sistema final que detallamos en el capítulo 6.

Los tests realizados en el sistema pueden dividirse en dos bloques principales: tests de optimización de los parámetros para el reconocimiento, y tests de evaluación final de los resultados. Los primeros fueron realizados en el entorno de simulación que desarrollamos en MatLab y que explicamos en el capítulo 5, mientras que los segundos fueron implementados en el sistema final en Arduino cuyo funcionamiento explicamos en el capítulo 6.

7.1. Tests de Optimización de Parámetros para el Reconocimiento de Insectos

En estas primeras rondas de tests nos centramos en encontrar los parámetros que mayor precisión podrían darnos en nuestra aplicación final. Para ello, realizamos dos rondas de tests con todos los insectos variando diferentes parámetros y observando los resultados. Los parámetros que intentamos optimizar fueron los siguientes:

7.1.1. *Tamaño de la ventana*

Con el tamaño de ventana se ven influidas la resolución del espectro en frecuencia y el coste computacional de la placa.

En la primera tanda de test hemos optado por un tamaño de ventana de 1024 y de 2048 muestras debido a que anteriormente se intentó trabajar con Arduino UNO con una ventana de 512 y los resultados no nos permitieron diferenciar de dos insectos que se situarán cerca en el espectro. Con Arduino DUE no podemos trabajar con más de 2048 de tamaño de ventana debido a que la memoria de la placa no permite hacer operaciones de tanto coste computacional.

7.1.2. Relación señal ruido (SNR)

La relación señal/ruido (SNR) se define como la proporción existente entre la potencia de la señal que se transmite y la potencia del ruido que la corrompe. Este margen es medido en decibelios.

Para nuestros test hemos optado por una SNR de 10 dB y 20 dB primeramente y en una segunda tanda de test se decidió aumentar la complejidad del reconocimiento haciendo este margen aún más pequeño con valores de 5 dB de relación Señal/Ruido.

7.1.3. Solapamiento

Con el solapamiento en la ventana de muestreo se perseguía reiterar los valores significativos de la señal de entrada. De esta forma se tendrían más tramas de una señal y por lo consiguiente un histograma con más valores y por lo tanto más representativo. Con esto también se intentaba que la campana de Gauss fuese lo más próximo posible a los valores del insecto, es decir, que tanto la desviación típica como la media de todas las tramas caracterizará lo mejor posible al insecto analizado.

7.1.4. Número de tramas

El número de tramas a analizar para el análisis de la señal de entrada es determinante, ya que influirá tanto en el tiempo de reconocimiento como en la precisión de este. En la segunda tanda de test se ha decidido optar por valores de 30 ó 40 tramas, ya que con menos no se tienen valores demasiados representativos para el cálculo de la media y de la desviación típica y con valores superiores a 40 estaríamos invirtiendo demasiado tiempo en el reconocimiento del insecto, sobre todo para insectos que no emitan sonidos constantemente.

7.1.5. Fórmula de similitud:

En este caso se propusieron dos posible fórmulas para medir la similitud entre la media y la desviación típica de las tramas tomadas hasta un cierto momento, y cada uno de los insectos:

Error Absoluto:

La primera era la raíz cuadrada de los cuadrados de las diferencias, conocida por sus siglas en inglés como RMSE [49]. Esta fórmula da igual valor a la media y a la desviación típica (50% a cada una) por lo que no permite optimizar esta parte del programa. Además, los valores que devuelve son números reales positivos, sin rango superior, por lo que su interpretación es relativamente compleja aunque sabemos cuanto mejor sea el resultado de RMSE, mejor es el error y por lo tanto mayor la precisión, no nos es posible comparar diferentes insectos ya que aunque los valores más acertados para un insecto de bajas frecuencias estén en unas cuantas decenas (entre 0 y 100), el valor más acertado para un insecto de altas frecuencias puede llegar a más de 4000. Esto hace imposible comparar en términos reales la precisión del sistema entre estos dos tipos de insectos (aunque a los dos tipos los reconozca acertadamente, no sabríamos decir cuál de los dos acierta con mayor precisión, ya que las frecuencias de los insectos se encuentran en rangos de números reales dispares y por lo tanto el error medido a través de RMSE también lo está. Otras fórmulas de medición de error en términos reales como MAE [50], MAD [51] o MSE [52] comparten los mismos aspectos negativos y producen resultados parecidos a RMSE. La fórmula de error que utilizamos para estos tests fue una variación de MAE en la que multiplicamos las diferencias de media y desviación estándar, sumándole 1 a cada diferencia para que en el caso de que solo uno de los parámetros la diferencia fuera 0 no establezca la similitud total a 0, tal y como se aprecia en la ecuación (1):

$$|\mu_i - \mu_t + 1| \cdot |\sigma_i - \sigma_t + 1| \quad (1)$$

Donde μ_i y σ_i son la media y desviación estándar de las frecuencias del insecto i con el que midiendo la similitud y μ_t y σ_t son la media y desviación estándar de las t tramas tomadas hasta ahora. Cuando $t = N$ (el número total de tramas a procesar), el reconocimiento se da por concluido.

Aperturas Normalizadas:

Esta segunda fórmula propuesta fue la finalmente elegida. Es simplemente una ecuación lineal en la que agregamos la media y la desviación típica, multiplicando cada una por un peso (encontramos que el mejor peso se encontraba en torno al 50% cada uno, pero fue de gran utilidad para este proyecto el ver como el peso de cada parámetro afectaba al resultado final), y dividiendo este resultado entre unos valores de apertura para normalizar esta suma en el intervalo [0, 100]. Después, el resultado de esta ecuación (que nos devuelve

la diferencia/error entre la muestra y cada insecto), es restado a 100 para convertirlo en similitud (así, consideramos que un 15% de error es igual a un 85% de precisión/acierto). Esta fórmula está detallada en código fuente en la sección “Ventanas y Similitud” del capítulo anterior, la cual describimos matemáticamente a continuación:

$$1 - \left(p_{\mu} \frac{|\mu_i - \mu_t|}{R_{\mu}} + (1 - p_{\mu}) \frac{|\sigma_i - \sigma_t|}{R_{\sigma}} \right) \quad (2)$$

Donde μ_i, σ_i, μ_t y σ_t tienen el mismo significado que en ecuación (1).

$p_{\mu} \in [0,1]$ es el peso relativo que la media tiene sobre la desviación típica. En nuestro caso, al querer trabajar en porcentajes en vez de en el intervalo normalizado $[0, 1]$, el valor de p_{μ} se ha fijado en 50 (la media y la desviación típica tienen el mismo peso relativo), por lo que luego nos ahorramos multiplicar por 100 el resultado. Esta cualidad no podíamos encontrarla en las fórmulas de medición de error en términos reales presentadas anteriormente, y es lo que la convierte en ideal para comparar la actuación del sistema entre insectos de distinto rango y así ver si la precisión es parecida a través de todos los rangos e insectos o si debíamos optimizar el reconocimiento de alguna parte del sistema.

Por último, R_{μ} y R_{σ} son los parámetros de apertura, que se fijan dependiendo del rango de frecuencias en las que el sistema esté focalizado (baja frecuencias, grillos, etc.). Estos parámetros permiten al sistema “**afinar el oído**” cuando identifica el rango de frecuencias en las que se encuentra el insecto escuchado actualmente. Esto lo conseguimos modificando las variables `apertura_media` y `apertura_desv`, respectivamente, también descritas en la sección “Ventanas y Similitud” del capítulo anterior. Su funcionamiento se basa en reducir estos valores por los cuales dividiremos una vez que el rango de frecuencia sea reconocido, dando así el efecto de que el sistema es más exigente a la hora de diferenciar insectos de un mismo rango (por ejemplo, grillos). Esta cualidad tampoco podíamos encontrarla en las fórmulas de error, que son propensas a dar valores demasiado parecidos entre insectos de una misma frecuencia, generando confusión en el sistema.

7.2. Tests realizados

En la primera ronda de tests para todos los insectos dejamos fijos los parámetros D y E para encontrar las situaciones en que A, B y C funcionaban mejor. En la segunda ronda de tests utilizamos todos los parámetros (sabiendo en qué sentido debíamos continuar aumentando o disminuyendo A y B) menos C, ya que encontramos que éste tenía una influencia mínima en el resultado final.

7.2.1. Test realizados en Matlab

A continuación pasamos a detallar los resultados de concretos de las tres rondas de tests de optimización realizados en Matlab

7.2.1.1. Primera Ronda de Tests de Optimización

En esta ronda de tests se realizaron 2520 mediciones de reconocimiento de insectos. Estas mediciones estuvieron divididas en series de 10 mediciones para cada uno de los 14 insectos con cada una de las configuraciones resultantes de combinar los siguientes valores para los 3 parámetros medidos:

- A. Tamaño de la ventana: Valores de 1024 y 2048
- B. Relación señal ruido (SNR): Valores de 10dB, 20dB y sin ruido añadido.
- C. Solapamiento: 30%, 60% y 90%
- D. Número de tramas: Fijo a 30
- E. Fórmula: Aperturas Normalizadas - ecuación (b) explicada anteriormente

En el Anexo 1 se detallan todas las series de tests realizadas en esta primera ronda.

Resultados totales para cada parámetro por separado:

La siguiente tabla incluye la precisión media de todos los tests en los que participa cada valor de cada parámetro.

	Tam. Ventana		Rel. Señal-Ruido			Solapamiento		
	1024	2048	SR*	10dB	20dB	30%	60%	90%
Media total de todos los tests con este parámetro (%)	84.52	89.85	92.04	79.81	89.71	86.86	87.22	87.48

Tabla 7.1. Media total de precisión para cada valor de parámetro en los tests que participa

Como podemos observar en tabla 7.1, al aumentar el tamaño de la ventana de 1024 a 2048, la precisión total del sistema se incrementa en más de un 5%. Además, como era de esperar, teniendo una SNR sin ruido es la configuración que mejor funciona en cuanto a este parámetro. Por último, cabe detallar el mínimo impacto del solapamiento en la precisión del sistema. Es por eso que este parámetro será fijado a 0 e ignorado en la segunda ronda de tests de optimización que detallaremos más adelante.

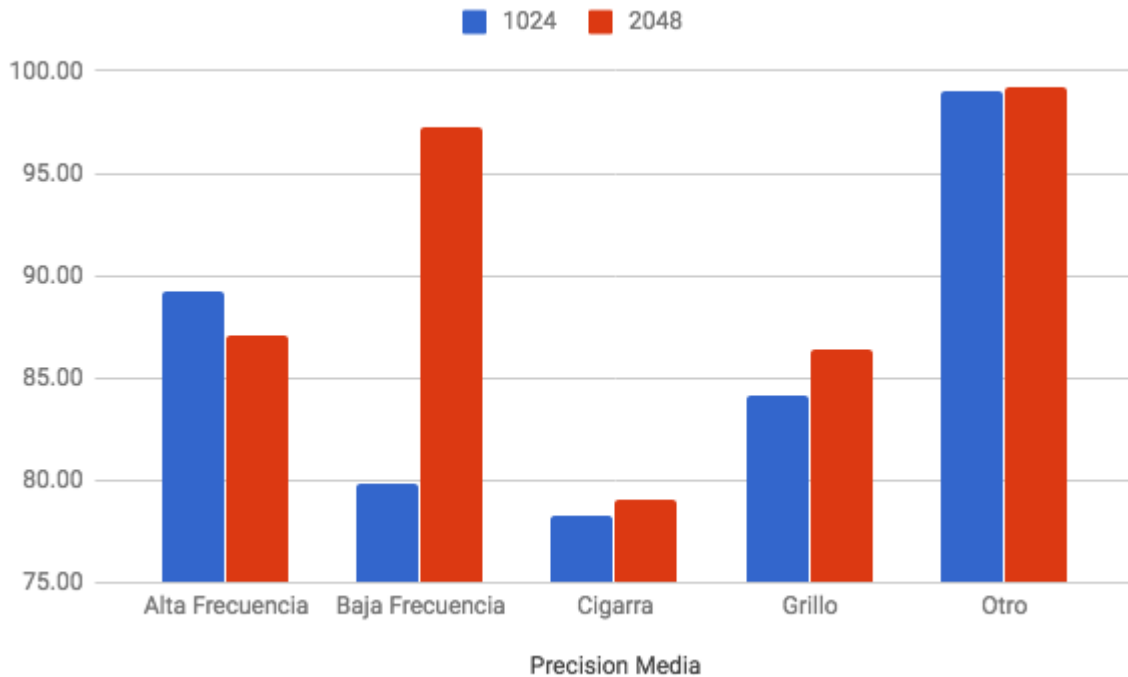


Gráfico 7.2. Precisión media por Rango de Frecuencia y Tamaño de Ventana

En la anterior gráfica se ha querido resaltar el comportamiento del tamaño de ventana en según qué tipo de insectos. Este parámetro es muy representativo para insectos de baja frecuencia debido a que se necesita una resolución mayor. Con un tamaño de ventana menor obtendremos una mayor separación de muestras en el espectro, lo que tiene una gran

repercusión para insectos que son similares en frecuencia., como lo son los de baja frecuencia.

Precisión total por cada valor de relación señal-ruido y rango de insecto

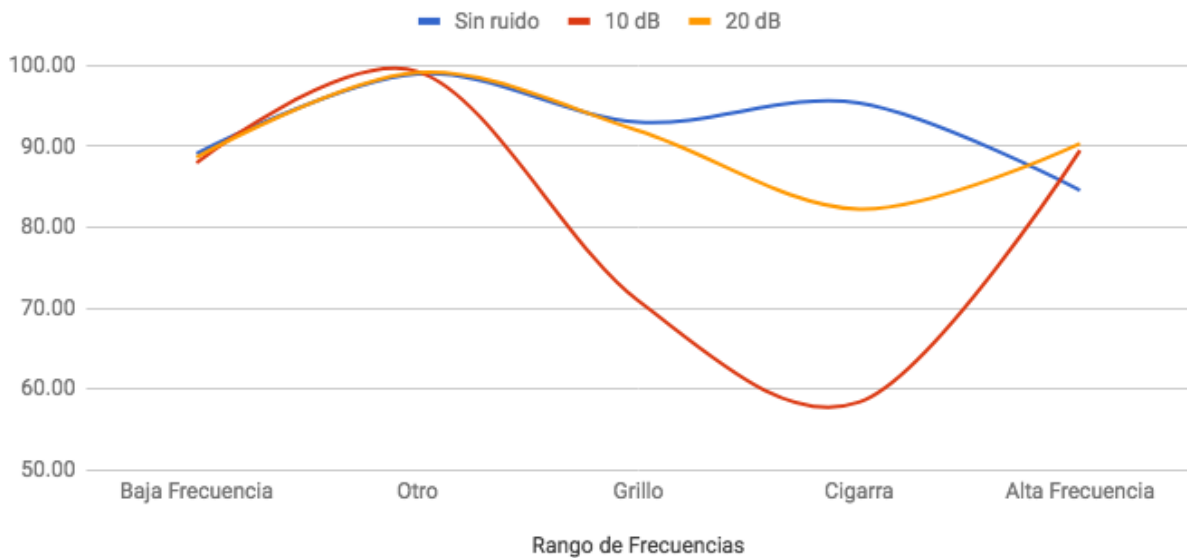


Gráfico 7.3. Precisión media por Rango de Frecuencia y SNR

En la anterior gráfica se observa como la SNR afecta al reconocimiento. El porcentaje de similitud cae considerablemente cuando la SNR es de tan solo 10 dB. También se aprecia como la similitud mejora con un rango mayor de relación señal a ruido, 20 dB.

En cuanto a la parte del espectro que más se ve afectada por la relación señal a ruido, podríamos decir que tanto como para las cigarras como para los grillos sufren más la aparición de ruido y esto se debe a que existen especies en este rango que son muy similares y que pueden llevar a confusión.

Por último, en el siguiente gráfico valoramos el comportamiento del reconocimiento de los insectos con el solapamiento. Viendo la anterior gráfica se puede apreciar como no existe mucha variabilidad entre los valores al modificar el parámetro de solapamiento. Es por esto por lo que se decidió eliminar el solapamiento y centrarnos en los atributos que tienen más peso a la hora de reconocer el insecto.

Precisión total por cada valor de solapamiento y rango de insecto

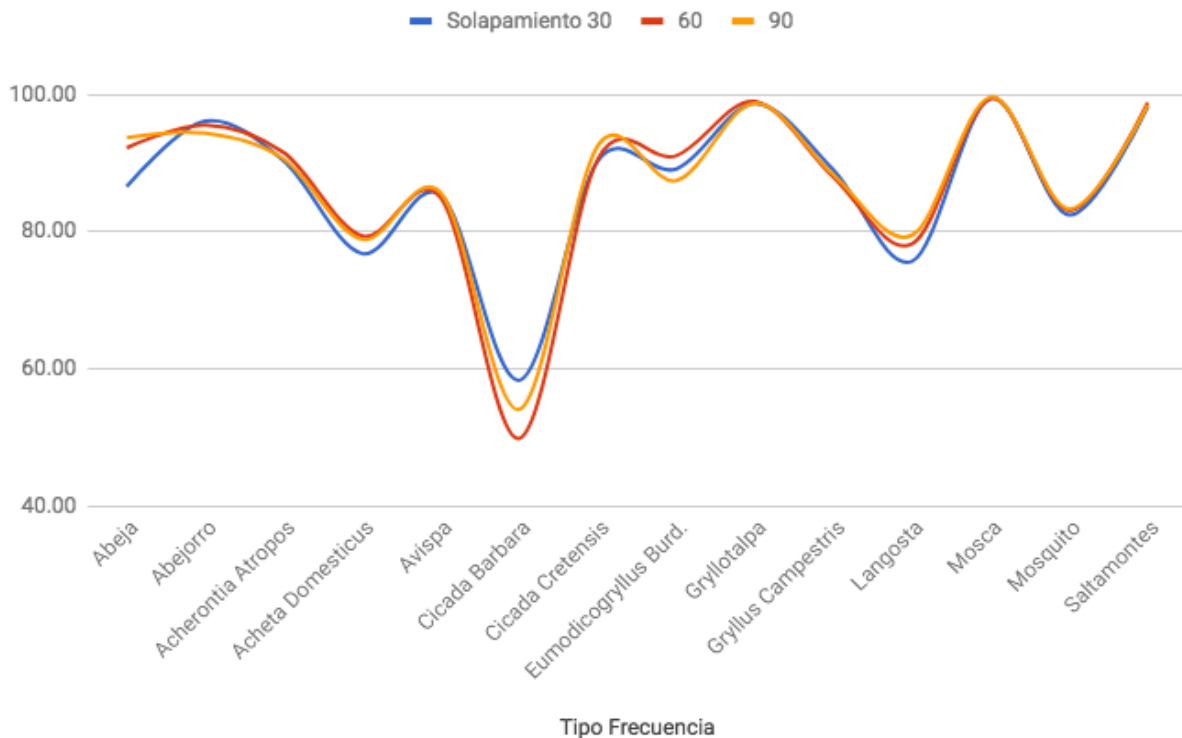


Gráfico 7.4. Precisión media por Rango de Frecuencia y Solapamiento

A pesar de estar todavía en fase de optimización, en la siguiente tabla podemos observar que el sistema ya era capaz de reconocer la mayoría de insectos, sobre todo en las configuraciones que mejor funcionaban (2048 de tamaño de venta y una SNR lo más amplia posible). La siguiente tabla muestra los resultados por cada insecto y tipo de frecuencias agregados de todos los tests.

El 50% de los insectos (7 de 14) obtuvieron precisiones medias mayores al 90%, y más del 78% (11 de 14) consiguieron ser reconocidos con una precisión de más del 80%. La precisión media de todo el sistema estuvo en torno al 87% (agregado de todos los tests realizados en esta ronda).

Tipo	Insecto	SNR: sin ruido			SNR: 10			SNR: 20		
		Media Precisión	Núm. Aciertos	Núm. Errores	Media Precisión	Núm. Aciertos	Núm. Errores	Media Precisión	Núm. Aciertos	Núm. Errores
Baja Frecuencia	Abeja	70.58	40	20	80.61	29	31	82.15	30	30
	Abejorro	98.59	60	0	98.38	50	10	98.52	60	0
	Avispa	90.33	60	0	92.54	56	4	89.50	59	1
	Mosquito	98.01	30	30	91.17	30	30	96.76	30	30
Otro	Gryllotalpa	85.19	60	0	85.32	60	0	85.35	60	0
	Mosca	82.94	60	0	82.86	60	0	82.95	60	0
Grillo	Acheta Domesticus	95.14	60	0	86.12	45	15	91.05	60	0
	Eumodicogryllus B.	96.39	60	0	2.39	58	2	63.50	60	0
	Gryllus Campestris	94.53	30	30	86.72	3	57	92.19	25	35
Cigarra	Acherontia Atropos	90.08	60	0	55.14	60	0	89.73	60	0
	Cicada Barbara	97.24	60	0	75.95	0	60	94.40	14	46
	Cicada Cretensis	91.69	60	0	81.68	49	11	91.72	59	1
Alta Frecuencia	Langosta	98.44	60	0	99.08	42	18	98.75	58	2
	Saltamontes	99.45	60	0	99.44	60	0	99.47	60	0
Total		92.04	760	80	79.81	602	238	89.72	695	145

Tabla 7.5. Precisión media de cada tipo de frecuencias y en cada insecto

En la anterior tabla (Tabla 7.5) es la tabla resultante de la primera tanda de test en Matlab. Aquí se perseguía comprobar cómo de importante era los parámetros de solapamiento, ruido y tamaño de ventana. De estos test se llegó a la conclusión de que la ventana de 2048 tenía mayor porcentaje de aciertos y que el solapamiento era irrelevante. También pudimos comprobar que el ruido era un factor determinante y que debíamos intentar reducirlo para tener una mayor tasa de aciertos, como se puede comprobar en la anterior tabla, a mayor relación Señal/Ruido menor número de errores en el reconocimiento.

A pesar de los buenos resultados obtenidos en esta primera ronda, consideramos que era necesario intentar mejorarlos optimizando el número de tramas a utilizar y probando distintas funciones de similitud y otros valores para los parámetros ya probados: subiendo el tamaño de la ventana y probando otras opciones para SNR. Estas adaptaciones se

realizaron como parte de una segunda ronda de tests de optimización, que pasamos a explicar a continuación.

7.2.1.2. *Segunda Ronda de Tests de Optimización*

En esta ronda de tests pretendemos no sólo optimizar los parámetros que comenzamos a probar en la primera ronda, sino que además queríamos ver los límites que el sistema podía tolerar en cuanto a ruido (cambiando los valores de SNR) y el tamaño de las tramas. Es por ello que en esta ronda de tests modificamos 4 de los 5 parámetros presentados anteriormente:

- A. Tamaño de la ventana: Valores de 2048 y 4096
- B. Relación señal ruido (SNR): Valores de 5dB y 10dB
- C. Solapamiento: No aplica, tal y como detallamos en los tests anteriores
- D. Número de tramas: Valores 30 y 40
- E. Fórmula: Error absoluto al cuadrado [74] (ecuación (a)) y Aperturas Normalizadas (ecuación (b)) - ambas explicadas al inicio de este capítulo

La segunda ronda de tests se compuso un total de 672 mediciones, divididas en series de 8 tests para cada una de las 6 configuraciones de parámetros que establecimos, realizadas todas para cada uno de los insectos:

1. 30 tramas, SNR de 5 dB y tamaño de la trama de 2048
2. 30 tramas, SNR de 10 dB y tamaño de la trama de 2048
3. 40 tramas, SNR de 5 dB y tamaño de la trama de 2048
4. 40 tramas, SNR de 10 dB y tamaño de la trama de 2048
5. 40 tramas, SNR de 5 dB y tamaño de la trama de 4096
6. 40 tramas, SNR de 10 dB y tamaño de la trama de 4096

Resultados totales para cada parámetro por separado:

La siguiente tabla incluye la precisión media de todos los tests en los que participa cada valor de cada parámetro.

	Tam. Ventana		SNR		Núm. Tramas	
	2048	4096	5 dB	10dB	30	40
Media total de todos los tests con este parámetro (%)	84.34	85.13	78.62	90.59	83.39	85.21

Tabla 7.6. Media total de precisión para cada valor de parámetro en los tests que participa

La tabla anterior incluye los resultados al aplicar la fórmula de similitud (b), que llamamos Aperturas Normalizadas, ya que así podemos comparar en términos porcentuales (normalizado) la precisión de cada parámetros por separado y sin otorgar mayor peso a los insectos de frecuencias más altas, como pasaría con la fórmula de error absoluto.

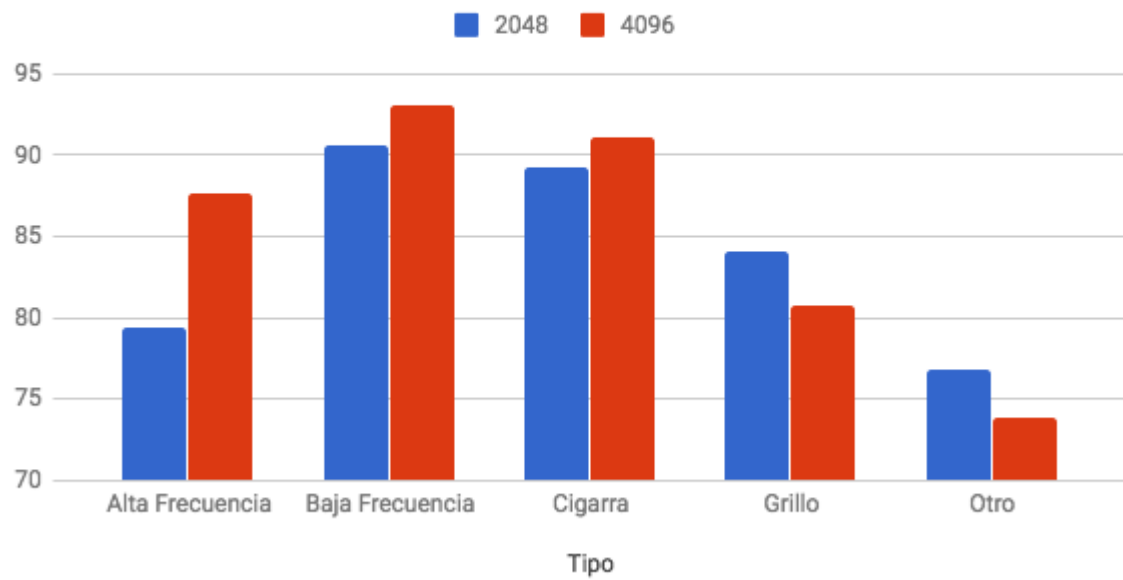
Como podemos observar en la tabla 7.6, al aumentar el tamaño de ventana a 4096, la precisión media sube ligeramente; quizás menos de lo esperado. Lo cual nos hace replantearnos si un 0.8% de mejora total del sistema es suficiente razón para contrarrestar el aumento en el tiempo de procesamiento de las tramas, que al ser el doble de tamaño, el sistema en Arduino tardara mucho más en procesarlas.

Por otro lado, vemos que número de tramas también aumenta sólo ligeramente la media de la precisión total. Esto también nos hace plantearnos si es justificable incrementar el reconocimiento de cada insecto en 1.3 segundos (lo que se tarda en procesar 10 tramas en una ventana de 2048) para algo menos de 2% de mejora.

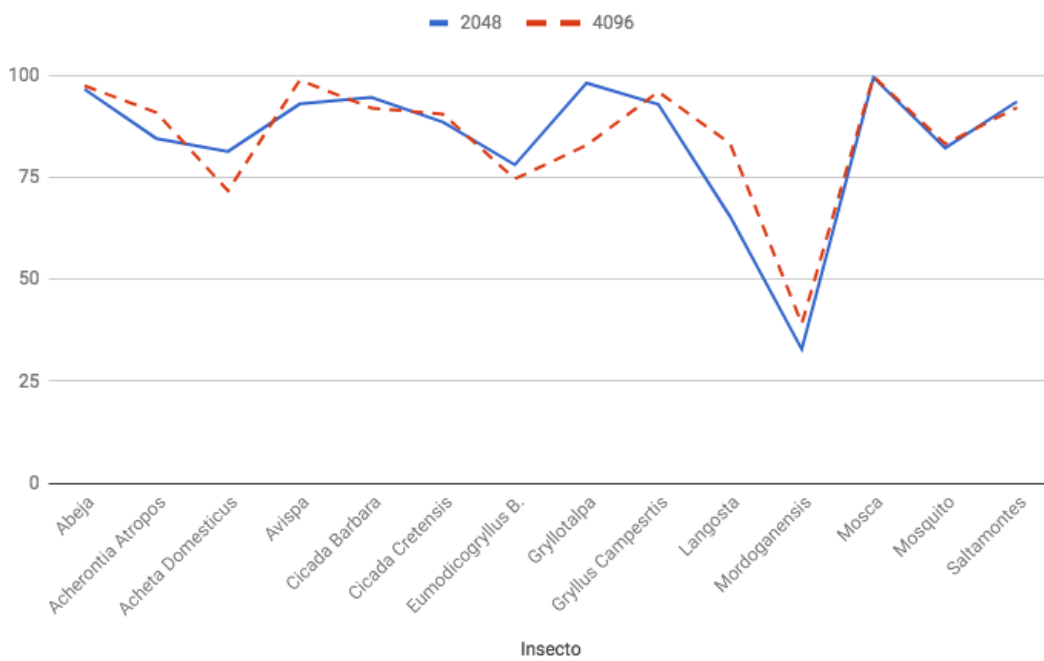
Finalmente, para simular un entorno real, hemos ido reduciendo la SNR para ver el comportamiento del sistema y conocer sus límites en algunos de los insectos.

Los gráficos presentados a continuación detallan estos resultados totales y comentan el porqué de su comportamiento.

Precision vs. Tamaño de la Trama, por rango de frecuencias de insectos



Tamaño de la trama: 2048 vs. 4096



Gráfica 7.7. Media total de precisión por cada rango de frecuencia e insecto y tamaño de trama

En la gráfica 7.7 podemos observar la diferencia entre realizar la transformada de Fourier con una ventana de 2048 o de 4096 y una frecuencia de muestreo de 24000 Hz. El resultado nos dice que no existe una gran diferencia entre ambas, ya que con una ventana de 2048 el reconocimiento sigue siendo notoriamente bueno y no merece la pena añadir coste computacional a la placa para obtener una mejor resolución en el espectro ya que aplicar una ventana de mayor tamaño influiría entre otras cosas en el tiempo de análisis de trama.

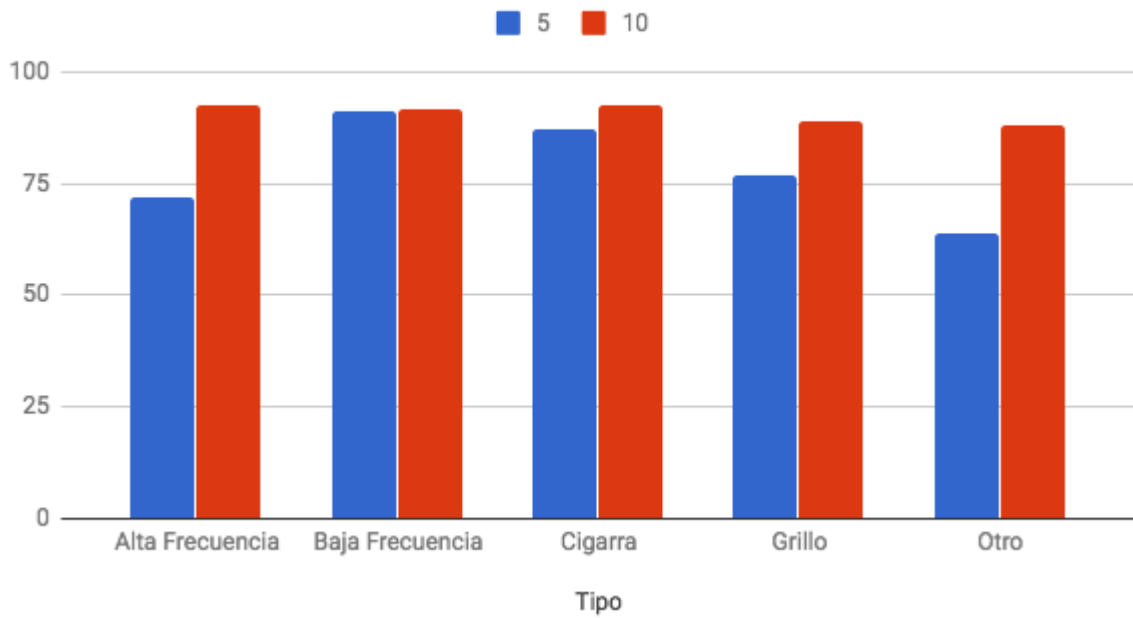
Además los insectos que más han notado el aumento de ventana son los de altas frecuencias los cuales han tenido un buen porcentaje de acierto, como se puede apreciar en la imagen superior de la gráfica 7.7, y no ha existido confusión entre ellos.

Por último, con la ventana de 2048 analizamos las tramas a una velocidad de 130 ms y eso nos permite obtener muchos más resultados y por lo tanto valores más representativos para el correcto reconocimiento.

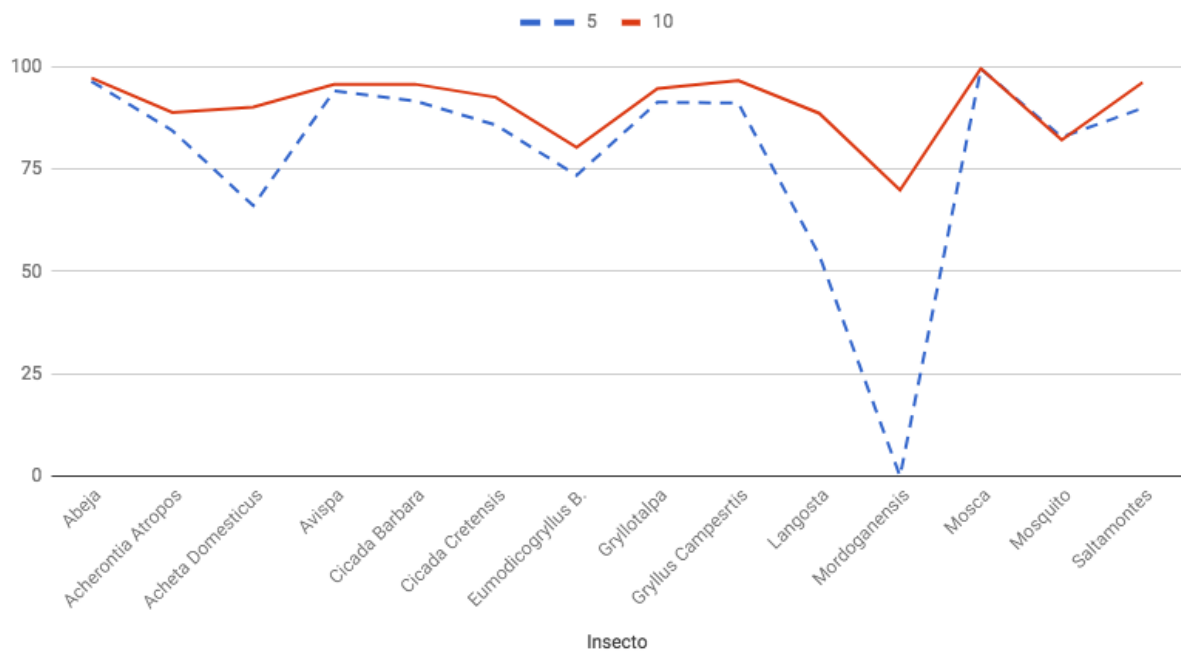
En la gráfica que se aprecia a continuación, gráfica 7.8, muestra cómo afecta la relación SNR al reconocimiento de insectos.

Se ha querido aumentar la dificultad del reconocimiento respecto a los test anteriores con una SNR de 5 dB. Los resultados nos han hecho ver que, como es lógico, al aumentar esta relación existe un mayor porcentaje de éxito. Sin embargo, el sistema es capaz de reconocer gran cantidad de insectos aun teniendo unas condiciones más hostiles.

Precision vs. SNR, por rango de frecuencias de insectos



Precisión vs. SNR: 5 y 10 dB

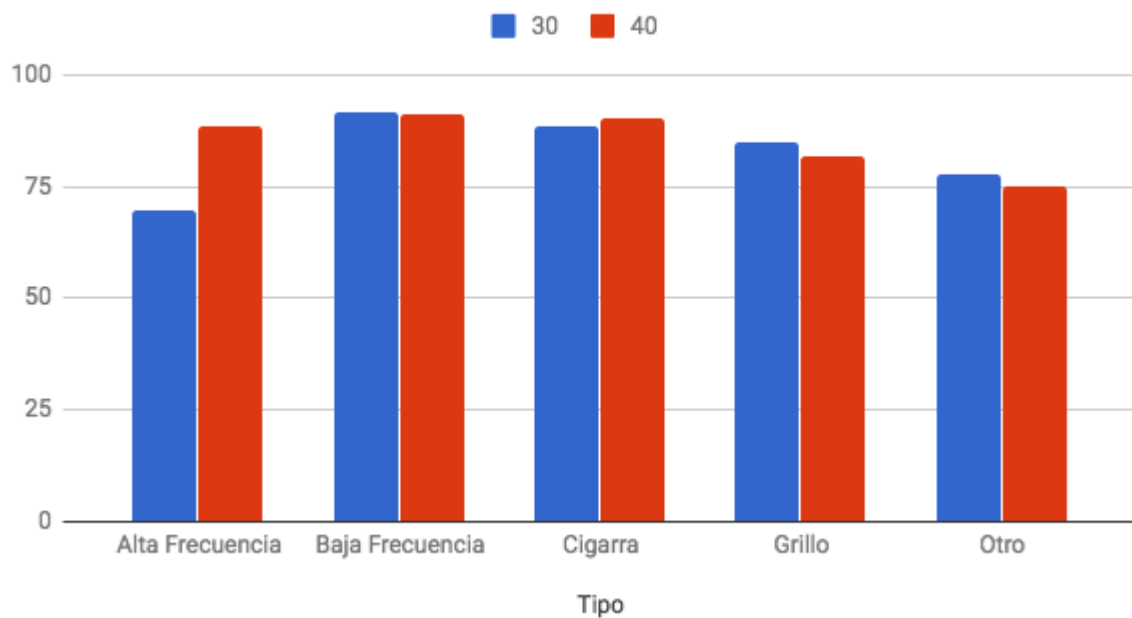


Gráfica 7.8. Media total de precisión por cada rango de frecuencia e insecto y SNR

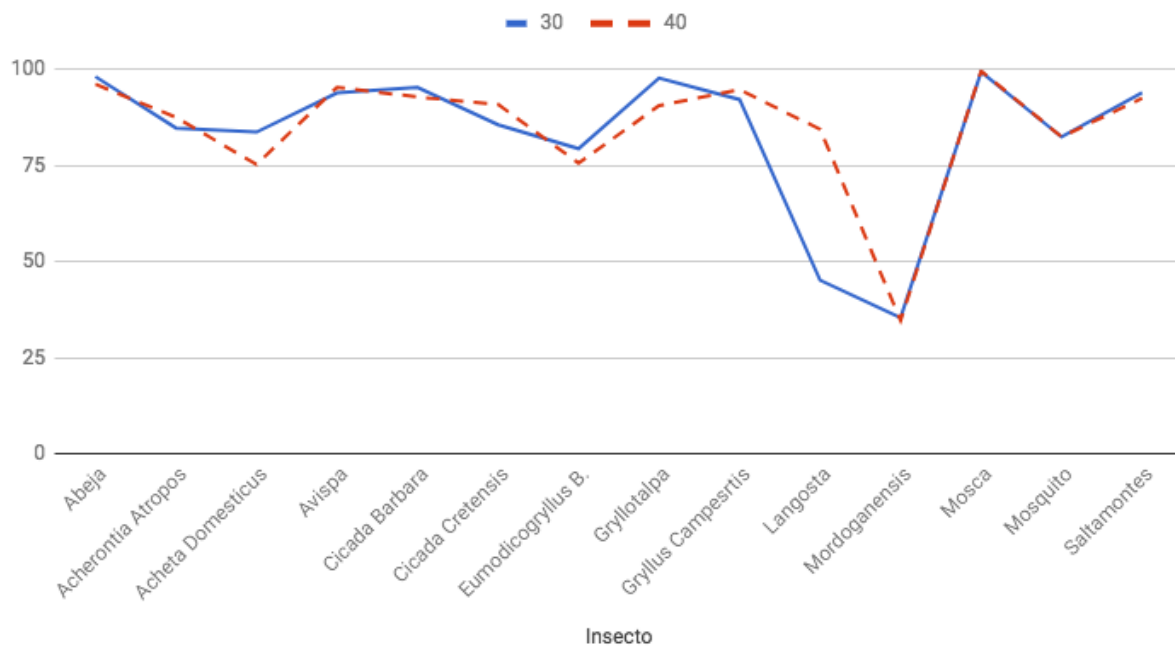
Para una mejor implementación del sistema en Arduino intentaremos disminuir el posible ruido de fondo que pudiera captar el micrófono, para ello algo muy importante es una buena colocación y situarlo en una zona lo menos ruidosa posible.

Por último pasamos a valorar el reconocimiento según el número de tramas y por lo que podemos ver en los datos de la gráfica 7.9, no existe una gran diferenciación entre reconocer a 30 tramas o a 40. Por lo tanto en Arduino optamos por trabajar con un reconocimiento de 30 tramas, ya que de esta forma el resultado lo obtendremos antes y la placa podrá reconocer insectos de manera más rápida que con tramas mayores.

Precisión vs. Número de Tramas: 30 y 40



Precisión vs. Número de Tramas: 30 y 40



Gráficas 7.9. Media total de precisión por rango de frecuencia e insecto y número de tramas

7.2.1.3. Tercera fase de test: Resultados aplicados

Tamaño de las tramas: 2048

Número de tramas: 30

Tipo	Insecto	SNR: 5			SNR: 10		
		Media Precisión	Núm. Aciertos	Núm. Errores	Media Precisión	Núm. Aciertos	Núm. Errores
Alta Frecuencia	Langosta	54.05	8	16	88.66	24	0
	Saltamontes	89.89	23	1	96.20	24	0
Baja Frecuencia	Abeja	96.45	24	0	97.26	24	0
	Avispa	94.09	18	6	95.71	20	4
	Mosquito	82.84	13	11	82.13	24	0
Cigarra	Acherontia Atropos	84.41	14	10	88.87	16	8
	Cicada Barbara	91.67	24	0	95.76	24	0
	Cicada Cretensis	85.78	23	1	92.57	24	0
Grillo	Acheta Domesticus	66.07	14	10	90.13	24	0
	Eumodicogryllus B.	73.49	24	0	80.30	24	0
	Gryllus Campestris	91.13	16	8	96.67	23	1
Otro	Gryllotalpa	91.37	19	5	94.66	20	4
	Mordoganensis	0.00	0	24	69.87	10	14
	Mosca	99.50	24	0	99.57	24	0
Total		78.62	244	92	90.60	305	31

Tabla 7.6. Precisión media de todos los tests realizados con 5 y 10 dB de SNR, todos con 2048 de tamaño de ventana y 30 tramas

De la anterior tabla (Tabla 7.6) se puede apreciar como el número de errores disminuye considerablemente cuando la relación Señal/Ruido (SNR) es más amplia. Esto nos hace ver, una vez más, que un factor muy importante para que el reconocimiento sea exitoso es que el ruido de fondo sea el menor posible respecto a nuestra señal de entrada del insecto por lo que debemos adaptar nuestra placa al entorno, buscando una situación con el menor ruido posible y con una buena orientación al insecto.

Media total de todos los tests con este parámetro (%)	SNR			Núm. Tramas	
	Sin Ruido	10dB	0dB	20	30
	100.00	93.20	54.87	83.06	82.32

Tabla 7.7. Media total de precisión para cada valor de parámetro en los tests que participa

En la tabla 7.7 se muestran los valores del último test realizado en Matlab en el que se ha intentado optimizar los parámetros que se deben utilizar. Para ello hemos supuesto unos casos más adversos que en los test anteriores disminuyendo la señal/ruido hasta 0 dB y suponiendo un menor número de tramas para el reconocimiento. Como se puede apreciar, el número de tramas no ha hecho disminuir el porcentaje de similitud, por lo que el sistema final de Arduino ya tendría valores significativos con 20 tramas pero final utilizaremos 30 tramas ya que el tiempo de procesado es muy rápido y con 30 tramas se podrían obtener valores más realistas.

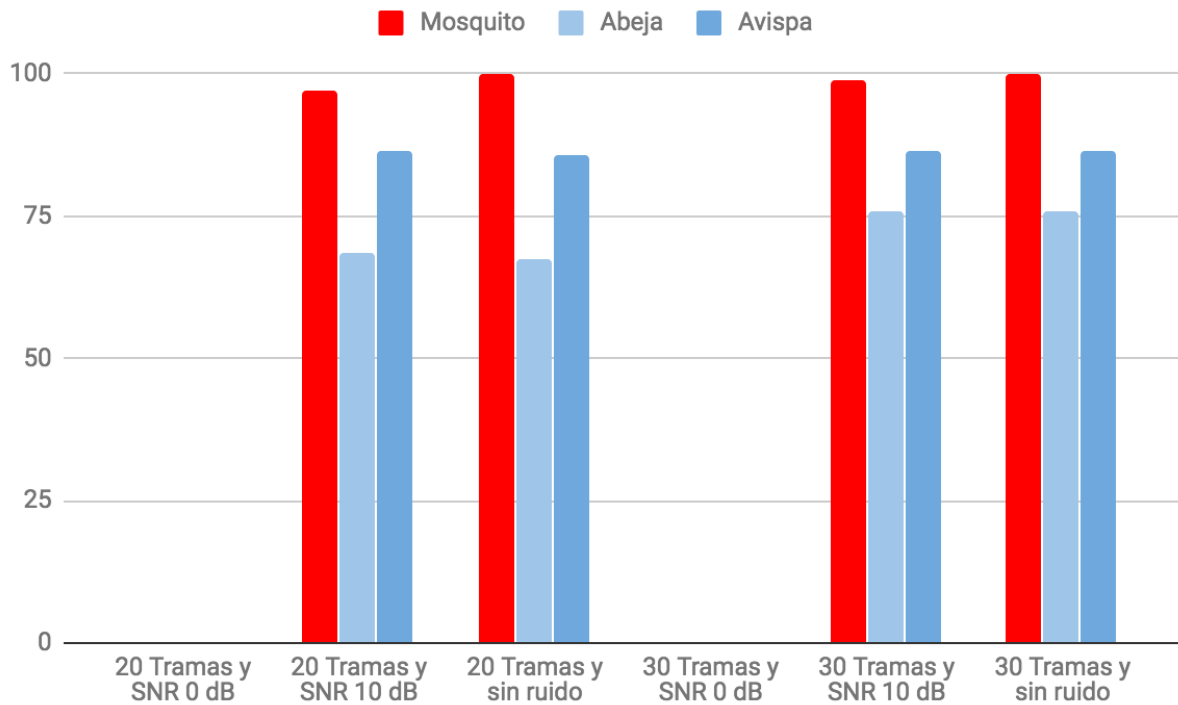
Respecto a la relación SNR se observa, una vez más, cómo el sistema es mucho eficaz con menor ruido.

Tipo	Insecto	SNR: 0			SNR: 10			SNR: Sin Ruido		
		Media Prec.	Núm. Aciert	Núm. Error	Media Prec.	Núm. Aciert	Núm. Error	Media Prec.	Núm. Aciert	Núm. Error
Alta Frec.	Langosta	0.00	0	8	89.43	8	0	100.00	8	0
	Saltamontes	72.31	7	1	98.02	8	0	100.00	8	0
Baja Frec.	Abeja	95.09	8	0	98.41	8	0	100.00	8	0
	Avispa	96.09	7	1	97.40	8	0	100.00	8	0
	Mosquito	0.00	0	8	98.94	8	0	100.00	8	0
Cigarra	Acherontia Atropos	9.77	0	8	84.48	6	2	100.00	8	0
	Cicada Barbara	64.10	4	4	91.89	8	0	100.00	8	0
	Cicada Cretensis	78.23	8	0	89.03	8	0	100.00	8	0
Grillo	Acheta Domesticus	33.31	5	3	93.48	8	0	100.00	8	0
	Eumodicogryllus B.	87.74	2	6	96.59	7	1	100.00	8	0
	Gryllus Campestris	17.74	4	4	86.71	8	0	100.00	8	0
Otro	Gryllotalpa	98.12	8	0	98.65	8	0	100.00	8	0
	Mordoganensis	0.00	0	8	84.64	8	0	100.00	8	0
	Mosca	97.82	8	0	99.57	8	0	100.00	8	0
Total		53.59	61	51	93.37	109	3	100.00	112	0

Tabla 7.8. Precisión media de todos los tests realizados con 5 y 10 dB de SNR, todos con 2048 de tamaño de ventana y 30 tramas

En la Tabla 7.8 se muestran los resultados de cada insecto en esta última tanda de test. En ella se puede destacar cómo algunos insectos tienen una tasa de error del 100% para una relación señal/ruido de 0 dB. Esto se debe a que la señal de entrada no tiene suficiente amplitud y que al incluir ruido la señal se distorsiona más que el resto de insectos. Estos errores no son debidos a confusiones con otros insectos sino que son debidos a que no tenemos una señal suficientemente fuerte como se puede apreciar en las tablas del anexo 3 donde se muestran todos los valores de todos los insectos.

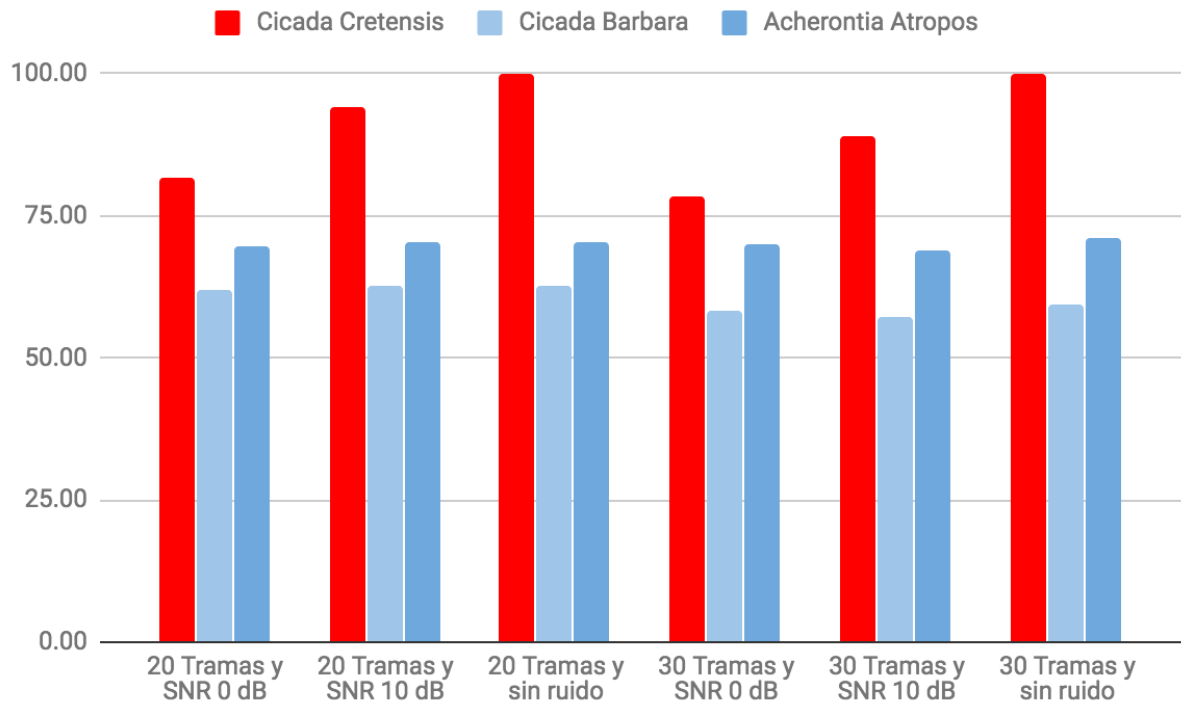
Los resultados anteriores nos hacen ver que existen 180 aciertos entre los parámetros de 0 dB y 10 dB. Esto nos dice que en total existe un porcentaje total de éxito del 75'85% para insectos que tengan una variabilidad de ruido que nos permita una relación SNR de entre 0~10 dB.



Gráfica 7.10. Reconocimiento de mosquito y como los otros insectos de baja frecuencia se comportan.

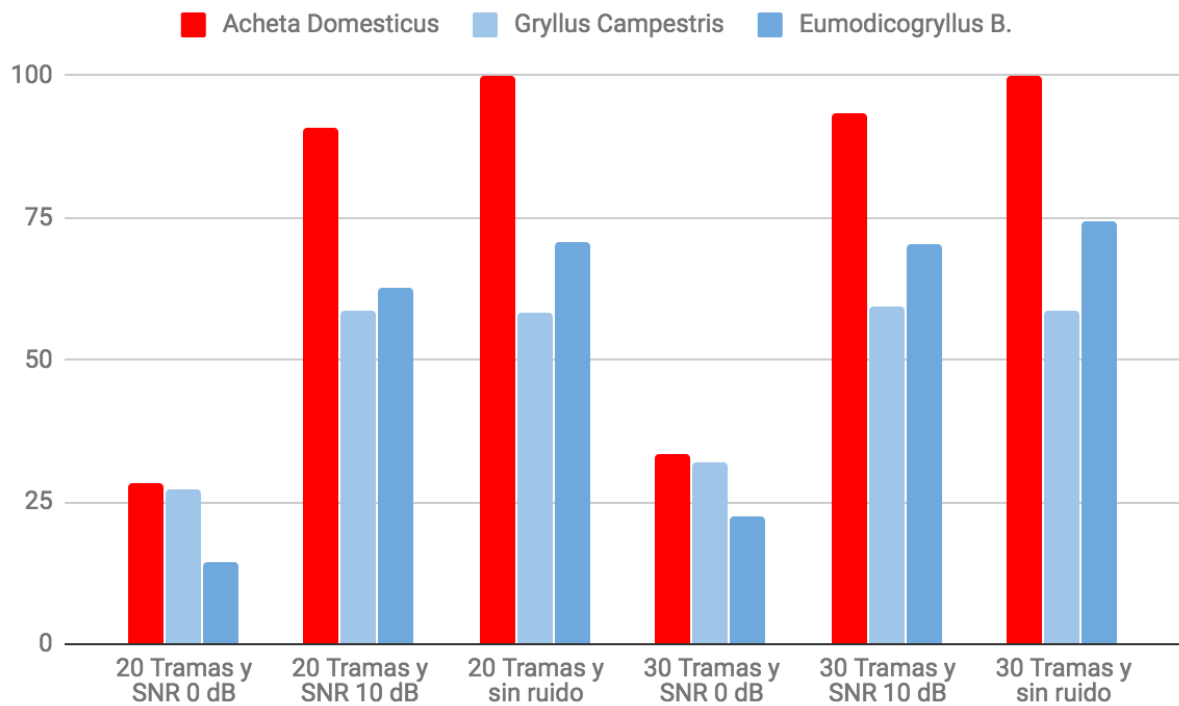
En esta última tanda de test también se ha querido comparar el porcentaje de similitud del insecto a reconocer junto con el de los insectos más similares en frecuencia, insectos de su mismo rango.

De la gráfica 7.10 se puede ver cómo se comporta el sistema al reconocer al insecto *mosquito*. Se puede ver cómo los insectos más cercanos. insectos de baja frecuencia, obtienen un porcentaje de similitud alto, pero siempre inferior al del insecto a reconocer.



Gráfica 7.11. Reconocimiento de cicada cretensis y como las otras cigarras se comportan.

De igual forma se muestra la gráfica de los porcentajes de similitud de los insectos de tipo *cigarra*. Una vez más, el insecto más similar es el que se quería reconocer ya que la *cicada cretensis* es el insecto con los porcentajes de similitud más altos. También podemos destacar cómo el número de tramas no influye en mucho para estos insectos.



Gráfica 7.12. Reconocimiento de *Acheta Domesticus* y como los otros grillos se comportan.

En el caso de los grillos se puede ver en la gráfica 7.12 cómo se comportan los demás grillos con una señal de entrada de la *Acheta Domesticus*. Todos tienen porcentajes de similitud altos pero destaca sobre ellos el porcentaje del insecto a reconocer.

7.2.2. Conclusión de los resultados de los tests de Matlab

Dado los resultados de todos test los realizados en Matlab se optará por la solución más eficiente y con mayor porcentaje de acierto. Como se ha explicado en el apartado anterior los parámetros a elegir son:

- Solapamiento:

Este parámetro lo descartamos en la segunda fase de test de Matlab puesto que no se veía una considerable mejora en el reconocimiento al aumentar el parámetro de

solapamiento. Por eso en la programación de la placa no se solapan los bloques del espectro temporal.

- Ventana al realizar la transformada de Fourier:

Este parámetro aporta resolución al espectro frecuencial como ya se ha detallado en este mismo capítulo. Los primeros test se realizaron con ventanas de tamaño 1024 y 2048 y se llegó a la conclusión de que con la ventana de 2048 teníamos un mayor porcentaje de aciertos, sobre todo en frecuencias más bajas. Por eso en la segunda fase de test de Matlab se aumentó el tamaño de la ventana (4096), pero estos segundos resultados no proporcionaron una mejora demasiado considerable y su coste de procesamiento hace que la placa se quede corta en memoria.

Por lo tanto la ventana que utilizaremos en Arduino será de 2048, la cual nos proporciona un espaciado entre muestras de 12 Hz. Este valor suficiente incluso para reconocer insectos de bajas frecuencias.

- Ruido en la señal:

Este ha sido el parámetro más determinante en nuestro reconocimiento. En la primera fase de test se decidió trabajar sin ruido, con una relación SNR de 20 dB y con una SNR de 10 dB. Como era de esperar los valores de reconocimiento más altos fueron los de sin ruido y 20 dB. Sin embargo, el sistema se comportó bien con una relación de 10 dB y por eso en la segunda fase de test se quiso incrementar la dificultad del reconocimiento y se disminuyó esta relación señal/ruido. Para estos segundos test se escogieron valores de 10dB de SNR y de 5dB. Se observó que aun teniendo una SNR tan baja se reconocían gran parte de los insectos.

Para la implementación de nuestra placa Arduino se buscará disminuir lo máximo posible el ruido no deseado por eso es muy importante que tengamos en cuenta el mayor número de factores posibles para disminuir este parámetro en nuestra placa de reconocimiento.

- Número de tramas para el reconocimiento final:

Este parámetro se decidió testear en la segunda fase de test, una vez que se comprobó el funcionamiento de otros parámetros. Con este valor esperamos encontrar el equilibrio entre tener datos representativos y no invertir demasiado tiempo en el reconocimiento.

Para ello se se decidió probar con valores de 30 y 40 tramas, ya que con menos de 30 los valores serían poco fiables y con más de 40 se emplearía mucho tiempo para poder reconocer insectos que no emitan sonidos periódicamente, como por ejemplo la langosta.

Al contrastar estos dos valores vimos que con 30 tramas obtenemos una representación acertada del insecto y por lo tanto escogimos este número para ahorrar tiempo de identificación.

En el siguiente apartado se aplican estos valores al sistema de Arduino y se realizan pruebas para analizar el funcionamiento.

7.3. Test de Evaluación sobre el Sistema Final

Tras haber finalizado los test de optimización de los parámetros que más podían influir en los resultados del sistema final, era hora de probar dichos parámetros en el sistema implementado en Arduino y entender cuál era el comportamiento del sistema final y comprobar cómo de similar era este comportamiento con el obtenido en los test de optimización realizados en el sistema simulado en Matlab.

7.3.1. Configuración de los test sobre el sistema final

En la elección de los parámetros que se utilizaron sobre el sistema final de Arduino se intentó equilibrar la precisión del sistema con el tiempo computacional en obtener el reconocimiento, tal y como se detalla en la sección anterior. En base a estos razonamientos, los parámetros sobre los que testeamos el sistema final son los siguientes:

- Tamaño de la trama o ventana: 2048 posiciones
- Puerta de ruido: 30 dB, para así evitar los peores resultados que obtuvimos cuando modificamos SNR
- Solapamiento: Descartado para su uso ya que demostró no influir en el resultado
- Número de tramas para el reconocimiento final: Aunque el sistema intenta reconocer el insecto tras procesar cada trama, éste dará por concluído el reconocimiento tras procesar la trama número 30
- Fórmula de similitud: Utilizaremos la fórmula de similitud que llamamos Aperturas Normalizadas (ecuación (b) explicada en la sección anterior), ya que así podremos

comparar en términos porcentuales el comportamiento del sistema a través de los diferentes rangos de frecuencias e insectos (algo que no podríamos hacer con otras fórmulas de error tradicionales ya que estas darían valores mucho más altos para los insectos de altas frecuencias comparados con los de bajas frecuencias y no entenderíamos en qué caso el sistema se comporta mejor)

En estas series de tests, utilizaremos esta configuración para tomar 20 mediciones de cada insecto, realizando en total 280 mediciones. A continuación observaremos los resultados globales de estas series de tests sobre cada insecto y rango de frecuencias.

7.3.2. *Conclusión de los resultados sobre los tests Evaluación sobre el Sistema Final*

Comenzaremos por describir los resultados globales, tal y como se muestran en la Tabla 7.7 debajo. Todos los resultados individuales de estas series de tests están disponibles en el Anexo 3.

Rango de Frecuencias	Media Precisión Total
<i>sin tipo</i>	79,96
<i>baja frecuencia</i>	87,10
<i>alta frecuencia</i>	87,53
<i>grillo</i>	88,31
<i>cigarra</i>	93,10
Media Total Sistema	87,17

Tabla 7.7 Precisión total del sistema y dividida entre los distintos rangos de frecuencias

En la tabla 7.7 se puede ver el porcentaje de similitud que se ha obtenido para cada tipo de insecto y el porcentaje total de reconocimiento del sistema, siendo un 87'17 % la

media total de todos los insectos para el sistema de reconocimiento de Arduino y con los parámetros antes citados.

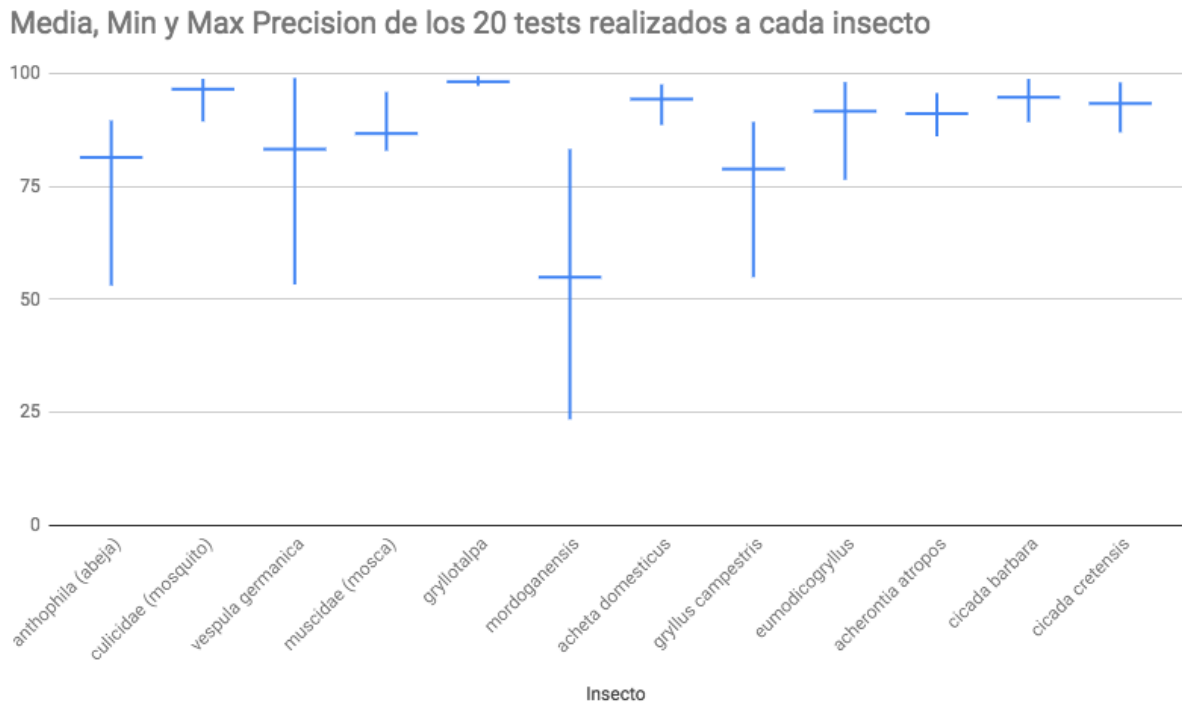


Gráfico 7.8 Precisión media, mínima y máxima para cada uno de los insectos testeados

El anterior gráfico (gráfico 7.8) representa los resultados de similitud de cada insecto. El eje vertical de cada cruz muestra el mayor y menor porcentaje de similitud del insecto y el eje horizontal que lo cruza representa la media del porcentaje de similitud de los test realizados.

De esa forma se puede apreciar cómo algunos insectos tienen mayores dificultades para ser reconocidos que otros.

Tipo	Insecto	Núm. Aciertos	Núm. Errores	Media Precisión	Max. Precisión	Min. Precisión
baja frecuencia	<i>anthophila (abeja)</i>	16	4	81,45	89,61	53,12
	<i>culicidae (mosquito)</i>	20	0	96,56	98,84	89,41
	<i>vespula germanica</i>	19	1	83,28	99,02	53,33
sin tipo	<i>muscidae (mosca)</i>	20	0	86,73	95,93	82,95
	<i>gryllotalpa</i>	20	0	98,22	99,43	97,31
	<i>mordoganensis</i>	20	0	54,94	83,28	23,49
grillo	<i>acheta domesticus</i>	20	0	94,35	97,6	88,64
	<i>eumodicogryllus</i>	18	2	91,68	98,1	76,52
	<i>gryllus campestris</i>	15	5	78,88	89,31	54,94
cigarra	<i>acherontia atropos</i>	18	2	91,14	95,7	86,12
	<i>cicada barbara</i>	20	0	94,76	98,81	89,3
	<i>cicada cretensis</i>	20	0	93,39	98,04	87
alta frecuencia	<i>acrididae (langosta)</i>	20	0	93,34	96,11	79,87
	<i>saltamontes</i>	17	3	81,71	88,92	63,79
MEDIA TOTAL		18,79	1,21	87,17	94,91	73,27

Tabla 7.9 Estadísticas totales de los 20 tests realizados para cada insecto

Como se puede observar en la Tabla 7.9, el número de errores medio entre los 20 test es de tan solo 1,21 (lo que es igual a tan solo un 6%). En muchos insectos vimos que el sistema fue capaz de acertar correctamente todas las veces. En este aspecto cabe destacar a “*mordoganensis*”, que a pesar de haber obtenido precisiones muy bajas en algunos de los test, en todos ellos fue el insecto con la precisión más alta.

Cabe destacar que los test se realizaron utilizando un altavoz para poder separar el micrófono del ordenador lo máximo posible y así evitar que el sonido del ordenador portátil desde el que leíamos los resultados introdujera demasiado ruido durante el reconocimiento. Por lo tanto, se puede considerar que estos test se realizaron bajo unas condiciones óptimas para el micrófono. Aun así, hemos podido observar que al ejecutar el sistema en otras condiciones no tan favorables, tal y como al aire libre y separando los altavoces del

micrófono y/o habiendo otro ruido de fondo, el rendimiento parece ser que no se ve demasiado afectado salvo para el caso de los insectos de bajas frecuencias.

8. CONCLUSIÓN

En este proyecto hemos demostrado que es posible desarrollar sistemas para la detección temprana de plagas con unos recursos hardware limitado y económico. Esta detección temprana se basa en el reconocimiento de insectos individuales que transiten cerca del micrófono del sistema, el cual analizará las señales de entrada e intenta reconocer de qué tipo de insecto se trata (primero focalizándose en la especie o rango de frecuencias para después intentar averiguar la especie del insecto).

Para diseñar este reconocimiento hemos desarrollado un entorno de optimización/ simulación en Matlab por el que testeamos los diferentes insectos para analizar el comportamiento del sistema con diferentes parámetros a calibrar (por ejemplo, el número de tramas a tomar para cada reconocimiento). De esta forma pudimos seleccionar los parámetros y valores que ofrecían un mayor rendimiento, siempre intentando establecer un equilibrio entre precisión y coste computacional. Estos parámetros y sus valores fueron luego trasladados al sistema hardware final en Arduino, del que obtuvimos resultados positivos bajo unas condiciones semi-ideales en cuanto a ruido de fondo y calidad del sonido. No obstante, tras otras mediciones ad-hoc hemos podido observar que el sistema sigue comportándose de forma fiable en la mayoría de los casos en estas condiciones no tan favorables (por ejemplo al aire libre).

A pesar de los resultados positivos, el sistema tiene varias posibilidades de mejora que no se han podido evaluar por falta de tiempo pero que al autor le gustaría remarcar como posible trabajo futuro. En el aspecto software, los pesos entre la media y la desviación típica de la fórmula de similitud o la puerta de ruido añadida para evitar el ruido de fondo podrían ser calibrados de la misma forma que se ha hecho con otros parámetros. También sería muy interesante evaluar otras especies y familias de insectos en rangos distintos a los evaluados en este proyecto. Por último, la parte software del sistema podría ser embebida en otro hardware con más recursos y especializado; por ejemplo: usar un micrófono con membrana para que el sonido se capta de forma más limpia y pueda ser focalizado, o una placa de desarrollo que incluyera una mini-pantalla donde visualizar los resultados sin depender de una conexión al ordenador.

9. REFERENCIAS

- [1] [https://es.wikipedia.org/wiki/Plaga#Plagas en la agricultura](https://es.wikipedia.org/wiki/Plaga#Plagas_en_la_agricultura)
- [2] [https://es.wikipedia.org/wiki/Acrididae#Plagas de los cultivos](https://es.wikipedia.org/wiki/Acrididae#Plagas_de_los_cultivos)
- [3] [https://es.wikipedia.org/wiki/Texas Instruments](https://es.wikipedia.org/wiki/Texas_Instruments)
- [4] <http://www.masadelante.com/faqs/memoria-rom>
- [5] <https://www.informatica-hoy.com.ar/aprender-informatica/Memoria-RAM-que-es.php>
- [6] <http://www.partesdeunacomputadora.net/procesador>
- [7] <http://smithsonianchips.si.edu/augarten/p38.htm>
- [8] [https://es.wikipedia.org/wiki/Perif%C3%A9rico de entrada/salida](https://es.wikipedia.org/wiki/Perif%C3%A9rico_de_entrada/salida)
- [9] [https://es.wikipedia.org/wiki/Raspberry Pi](https://es.wikipedia.org/wiki/Raspberry_Pi)
- [10] [https://es.wikipedia.org/wiki/ECB AT91](https://es.wikipedia.org/wiki/ECB_AT91)
- [11] [https://es.wikipedia.org/wiki/BASIC Stamp](https://es.wikipedia.org/wiki/BASIC_Stamp)
- [12] <https://aprendiendoarduino.wordpress.com/2015/03/23/shields-para-arduino/>
- [13] <http://www.monografias.com/trabajos4/lenguajec/lenguajec.shtml>
- [14] <https://www.arduino.cc/en/Main/ArduinoBoardDuemilanove>
- [15] <http://www.atmel.com/devices/atmega328p.aspx>
- [16] http://arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf
- [17] <http://www.atmel.com/devices/atmega2560.aspx>
- [18] http://www.arduino.cc/en/uploads/Main/arduino-mega2560_R3-sch.pdf
- [19] <http://www.atmel.com/devices/sam3x8e.aspx>
- [20] <http://www.atmel.com/devices/atmega32u4.aspx>
- [21] <https://en.wikipedia.org/wiki/E-textiles>
- [22] [https://es.wikipedia.org/wiki/Universal Asynchronous Receiver-Transmitter](https://es.wikipedia.org/wiki/Universal_Asynchronous_Receiver-Transmitter)
- [23] [https://es.wikipedia.org/wiki/Oscilador de cristal](https://es.wikipedia.org/wiki/Oscilador_de_cristal)
- [24] <https://es.wikipedia.org/wiki/JTAG>
- [25] <https://www.arduino.cc/reference/es/>
- [26] [https://es.wikipedia.org/wiki/C \(lenguaje de programaci%C3%B3n\)](https://es.wikipedia.org/wiki/C_(lenguaje_de_programaci%C3%B3n))

- [27] <https://es.wikipedia.org/wiki/C%2B%2B>
- [28] <https://www.arduino.cc/en/Guide/Introduction>
- [29] https://es.wikipedia.org/wiki/Bandas_cr%C3%ADticas
- [30] https://es.wikipedia.org/wiki/Curva_isof%C3%B3nica
- [31] <http://audiosocial.es/2013/06/09/la-membrana-basilar/>
- [32] <https://es.wikipedia.org/wiki/Arm%C3%B3nico>
- [33] <http://www-ma4.upc.edu/~lali/as/serie-fourier.pdf>
- [34] https://es.wikipedia.org/wiki/Frecuencia_fundamental
- [35] <http://www-ma4.upc.edu/~lali/as/serie-fourier.pdf>
- [36] https://es.wikipedia.org/wiki/Dicloro_difenil_tricloroetano
- [37] https://www.atsdr.cdc.gov/es/phs/es_phs35.html
- [38]

http://www.construmatica.com/construpedia/Los_Infrarrojos_como_Herramienta_Profesional_en_el_Control_de_Plagas

[39]

https://es.wikipedia.org/wiki/Diodo_organico_de_emision_de_luz

- [40] <https://es.wikipedia.org/wiki/Piezoelectricidad>
- [41] <http://www.anecpla.com/contenido-anecpla-1753>
- [42] <http://ieeexplore.ieee.org/document/5957464/?reload=true>
- [43] <https://mashable.com/2013/08/30/cicada-hunt-app/#OIUq8lJa2aqc>
- [44] https://es.wikipedia.org/wiki/Transformada_de_Fourier
- [45] <https://www.youtube.com/watch?v=spUNpyF58BY>
- [46] <https://github.com/bazaar-projects/kissfft>
- [47] <https://github.com/dujianyi/ardFFT>
- [48] https://github.com/bazaar-projects/kissfft/blob/master/kiss_fft.c#L339
- [49] https://en.wikipedia.org/wiki/Root-mean-square_deviation
- [50] https://en.wikipedia.org/wiki/Mean_absolute_error
- [51] https://en.wikipedia.org/wiki/Average_absolute_deviation
- [52] https://en.wikipedia.org/wiki/Mean_squared_error

10. PRESUPUESTO

10.1. Recursos materiales

Cantidad	Concepto	Precio (€)	Total (€)
1	Ordenador	434'5	434'50
1	Window 10	220'41	220'41
1	Matlab R2016a	2000'0	2000'00
1	Office Profesional 2016	425'81	425'81
1	Arduino DUE	40'50	40'50
1	Electret MAX4466	7'81	7'81
			3128'84

10.2. Honorarios

Cantidad (Horas)	Concepto	Precio/Hora (€/Hora)	Total (€)
110	Estudio del proyecto	15'00	1650'00
240	Implementación	25'00	6000'00
90	Redacción del proyecto	15'00	1350'00
			9000'00

10.3. Presupuesto total

Cantidad	Concepto	Total (€)
1	Recursos materiales	3128'84
1	Honorarios	9000'00
	Total sin IVA	12128'84
	IVA (21%)	2547'06
	Total con IVA	14675'90

Anexo 1: Detalle de la Primera Ronda de Tests de Optimización

<i>Acherontia Atropos</i>													
<i>(Cigarra)</i>													
<i>TEST</i>	<i>Porcentajes de similitud (%)</i>										<i>MEDIA</i>	<i>Desv. STD</i>	
Porcentaje de Solapamiento 30%, ventana 1024, sin ruido	4.20	4.20	4.20	4.20	4.20	4.20	4.20	4.20	4.20	4.20	4.20	4.20	0.00
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 10 dB	2.77	7.12	7.12	4.34	6.94	0.99	6.75	4.78	5.35	7.60	5.38	7.74	
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 20 dB	5.58	4.50	0.58	4.81	5.51	1.74	2.18	4.90	0.87	0.05	0.07	6.46	
Porcentaje de Solapamiento 30%, ventana 2048, sin ruido	5.33	5.33	5.33	5.33	5.33	5.33	5.33	5.33	5.33	5.33	5.33	0.00	
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 10 dB	5.40	9.67	0.68	0.99	3.23	0.14	4.21	3.24	9.36	8.70	6.56	6.94	
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 20 dB	5.62	9.40	5.56	9.64	4.77	0.05	4.65	7.49	6.87	6.67	0.07	6.23	
Porcentaje de Solapamiento 60%, ventana 1024, sin ruido	4.55	4.55	4.55	4.55	4.55	4.55	4.55	4.55	4.55	4.55	4.55	0.00	
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 10 dB	1.81	2.87	8.80	2.83	9.30	1.69	6.58	2.25	1.34	8.69	6.62	7.31	
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 20 dB	6.27	8.42	5.28	6.66	2.42	4.34	9.52	9.75	9.69	9.32	1.17	3.34	
Porcentaje de Solapamiento 60%, ventana 2048, sin ruido	6.30	6.30	6.30	6.30	6.30	6.30	6.30	6.30	6.30	6.30	6.30	0.00	
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 10 dB	0.39	7.62	4.42	1.72	3.16	7.16	1.07	9.16	4.06	1.24	7.00	7.80	
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 20dB	0.79	5.05	8.73	4.21	6.85	0.05	7.08	7.45	1.19	1.31	3.27	3.24	
Porcentaje de Solapamiento 90%, ventana 1024, sin ruido	4.36	4.36	4.36	4.36	4.36	4.36	4.36	4.36	4.36	4.36	4.36	0.00	
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 10 dB	3.99	4.21	1.77	3.81	6.05	2.20	0.31	9.59	1.46	0.11	2.35	4.79	

Porcentaje de Solapamiento 90%, ventana 1024, con SNR 20 dB	9.60	9.21	4.54	9.84	7.84	3.55	9.70	7.09	9.48	9.48	9.03	2.30
Porcentaje de Solapamiento 90%, ventana 2048, sin ruido	6.08	6.08	6.08	6.08	6.08	6.08	6.08	6.08	6.08	6.08	6.08	0.00
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 10 dB	8.67	1.29	4.61	8.56	7.19	6.34	8.46	6.00	7.17	9.75	8.80	2.59
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 20 dB	3.08	8.29	2.11	0.91	3.10	6.91	4.94	0.19	4.31	2.85	2.67	2.47

Mosquito (Baja Frecuencia)												
TEST	Porcentajes de similitud (%)										EDIA	^D esviacion Estandar
Porcentaje de Solapamiento 30%, ventana 1024, sin ruido	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	0.00
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 10 dB	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	0.00
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 20 dB	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	0.00
Porcentaje de Solapamiento 30%, ventana 2048, sin ruido	5.84	5.84	5.84	5.84	5.84	5.84	5.84	5.84	5.84	5.84	5.84	0.00
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 10 dB	6.48	5.53	6.48	5.61	5.55	7.31	7.24	5.68	5.35	3.92	5.91	1.01
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 20 dB	5.77	7.05	5.44	7.24	6.04	5.35	5.43	5.60	7.00	6.86	6.18	0.77
Porcentaje de Solapamiento 60%, ventana 1024, sin ruido	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	0.00
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 10 dB	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	0.00
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 20 dB	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	0.00
Porcentaje de Solapamiento 60%, ventana 2048, sin ruido	7.50	7.50	7.50	7.50	7.50	7.50	7.50	7.50	7.50	7.50	7.50	0.00

Porcentaje de Solapamiento 60%, ventana 2048, con SNR 10 dB	7.82	5.78	7.30	7.58	7.29	7.29	7.79	7.05	5.95	5.33	6.92	0.90
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 20dB	7.39	7.31	7.36	7.70	7.30	7.34	7.65	7.63	6.90	7.50	7.41	0.23
Porcentaje de Solapamiento 90%, ventana 1024, sin ruido	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	0.00
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 10 dB	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	0.00
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 20 dB	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	0.00
Porcentaje de Solapamiento 90%, ventana 2048, sin ruido	7.74	7.74	7.74	7.74	7.74	7.74	7.74	7.74	7.74	7.74	7.74	0.00
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 10 dB	7.64	7.81	7.94	6.97	7.48	8.07	7.96	8.00	7.71	8.10	7.77	0.34
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 20 dB	7.90	7.55	7.15	7.54	7.52	7.94	7.78	7.90	7.32	7.36	7.60	0.27

Abeja														
												MEDI	Desviacion	
TEST												A	Estandar	
Porcentajes de similitud (%)														
Porcentaje de Solapamiento 30%, ventana 1024, sin ruido	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	0.00
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 10 dB	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	0.00
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 20 dB	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	0.00
Porcentaje de Solapamiento 30%, ventana 2048, sin ruido	7.37	7.37	7.37	7.37	7.37	7.37	7.37	7.37	7.37	7.37	7.37	7.37	7.37	0.00
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 10 dB	9.64	9.64	9.51	4.11	0.15	8.86	9.65	6.64	9.63	9.64	7.75	3.24		
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 20 dB	7.49	7.35	7.34	7.46	7.34	7.46	7.70	7.76	7.56	7.13	7.46	0.18		
Porcentaje de Solapamiento 60%, ventana 1024, sin ruido	8.65	8.65	8.65	8.65	8.65	8.65	8.65	8.65	8.65	8.65	8.65	8.65	8.65	0.00

Porcentaje de Solapamiento 60%, ventana 1024, con SNR 10 dB	5.54	5.54	5.54	5.54	8.02	8.02	8.02	5.54	8.02	8.02	6.78	11.85
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 20 dB	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	0.00
Porcentaje de Solapamiento 60%, ventana 2048, sin ruido	7.19	7.19	7.19	7.19	7.19	7.19	7.19	7.19	7.19	7.19	7.19	0.00
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 10 dB	5.97	9.56	7.42	7.18	7.20	6.51	7.04	9.63	7.22	7.56	7.53	1.18
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 20dB	7.24	7.63	6.93	7.51	7.09	7.65	7.42	7.23	7.74	7.71	7.41	0.28
Porcentaje de Solapamiento 90%, ventana 1024, sin ruido	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	5.54	0.00
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 10 dB	8.02	8.02	8.02	8.02	8.02	8.02	8.02	8.02	8.02	8.02	8.02	0.00
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 20 dB	8.02	5.54	8.02	8.02	8.02	8.02	8.02	8.02	8.02	5.54	3.52	9.48
Porcentaje de Solapamiento 90%, ventana 2048, sin ruido	7.70	7.70	7.70	7.70	7.70	7.70	7.70	7.70	7.70	7.70	7.70	0.00
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 10 dB	9.65	9.54	9.64	9.65	9.64	9.65	9.61	9.63	9.62	9.63	9.63	0.03
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 20 dB	7.19	7.46	7.72	7.70	7.48	7.54	7.44	7.60	7.63	7.58	7.53	0.16

Cicada Barbara (Cigarra)													
TEST	Porcentajes de similitud (%)										MEDI A	Desviacion Estandar	
Porcentaje de Solapamiento 30%, ventana 1024, sin ruido	4.72	4.72	4.72	4.72	4.72	4.72	4.72	4.72	4.72	4.72	4.72	4.72	0.00
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 10 dB	R	R	R	R	R	R	R	R	R	R	R	.00	0.00
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 20 dB	6.17	4.28	8.81	7.91	6.00	8.21	0.58	7.95	8.64	6.53	2.51	7.32	
Porcentaje de Solapamiento 30%, ventana 2048, sin ruido	6.23	6.23	6.23	6.23	6.23	6.23	6.23	6.23	6.23	6.23	6.23	0.00	

Porcentaje de Solapamiento 30%, ventana 2048, con SNR 10 dB	7.63	5.63	R	R	R	R	R	R	R	R	.33	20.33
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 20 dB	8.27	R	4.36	0.27	R	6.43	9.46	4.85	1.19	6.75	7.16	36.42
Porcentaje de Solapamiento 60%, ventana 1024, sin ruido	7.97	7.97	7.97	7.97	7.97	7.97	7.97	7.97	7.97	7.97	7.97	0.00
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 10 dB	R	R	R	R	R	R	R	R	R	R	.00	0.00
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 20 dB	3.08	3.64	7.65	3.94	9.49	3.03	3.72	8.30	2.29	9.55	2.47	20.77
Porcentaje de Solapamiento 60%, ventana 2048, sin ruido	5.97	5.97	5.97	5.97	5.97	5.97	5.97	5.97	5.97	5.97	5.97	0.00
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 10 dB	5.96	R	R	R	R	R	R	R	R	R	.60	8.21
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 20 dB	R	R	1.95	R	8.24	5.34	.00	7.38	9.17	8.83	0.09	38.44
Porcentaje de Solapamiento 90%, ventana 1024, sin ruido	6.90	6.90	6.90	6.90	6.90	6.90	6.90	6.90	6.90	6.90	6.90	0.00
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 10 dB	R	R	R	R	R	R	R	.00	R	R	.00	0.00
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 20 dB	0.50	9.57	9.85	0.31	5.53	5.22	3.78	9.22	4.67	0.17	1.88	9.06
Porcentaje de Solapamiento 90%, ventana 2048, sin ruido	6.53	6.53	6.53	6.53	6.53	6.53	6.53	6.53	6.53	6.53	6.53	0.00
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 10 dB	R	R	R	R	4.07	R	.00	R	.00	.00	.41	7.61
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 20 dB	1.44	9.60	7.24	5.59	3.39	6.34	4.38	6.75	8.73	5.25	6.87	7.98

<i>Cicada Cretensis (Cigarra)</i>													
TEST	Porcentajes de similitud (%)										MEDIA	Desviaci	
Porcentaje de Solapamiento 30%, ventana 1024, sin ruido	4.76	4.76	4.76	4.76	4.76	4.76	4.76	4.76	4.76	4.76	4.76	4.76	0.00
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 10 dB	0.94	2.40	8.64	8.50	7.58	3.66	3.66	8.64	0.65	5.19	8.99	1.46	

Porcentaje de Solapamiento 30%, ventana 1024, con SNR 20 dB	6.93	7.21	8.30	0.14	7.35	4.57	0.92	4.62	1.25	0.69	9.20	4.93
Porcentaje de Solapamiento 30%, ventana 2048, sin ruido	6.29	6.29	6.29	6.29	6.29	6.29	6.29	6.29	6.29	6.29	6.29	0.00
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 10 dB	0.69	4.47	5.33	1.02	7.72	9.83	0.82	0.44	1.59	6.65	9.86	6.52
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 20 dB	1.83	5.84	6.06	9.11	3.80	3.64	3.39	5.86	2.16	5.41	2.71	3.12
Porcentaje de Solapamiento 60%, ventana 1024, sin ruido	3.65	3.65	3.65	3.65	3.65	3.65	3.65	3.65	3.65	3.65	3.65	0.00
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 10 dB	2.42	2.04	5.75	5.96	3.11	7.60	3.16	5.46	2.03	4.66	5.22	2.98
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 20 dB	0.54	3.98	2.11	4.21	1.19	5.90	1.30	6.05	4.51	9.62	2.94	2.28
Porcentaje de Solapamiento 60%, ventana 2048, sin ruido	1.27	1.27	1.27	1.27	1.27	1.27	1.27	1.27	1.27	1.27	1.27	0.00
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 10 dB	5.84	1.75	3.79	0.87	4.54	4.63	9.15	6.10	7.00	6.69	0.04	3.55
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 20dB	9.42	3.41	7.69	2.41	1.26	7.15	0.94	9.27	9.97	2.67	0.42	2.10
Porcentaje de Solapamiento 90%, ventana 1024, sin ruido	4.86	4.86	4.86	4.86	4.86	4.86	4.86	4.86	4.86	4.86	4.86	0.00
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 10 dB	5.88	7.73	1.31	5.33	1.22	7.02	4.18	6.77	8.43	5.56	7.34	2.40
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 20 dB	1.91	4.87	3.52	3.74	0.20	2.17	3.90	1.98	2.89	0.47	2.57	1.50
Porcentaje de Solapamiento 90%, ventana 2048, sin ruido	6.36	6.36	6.36	6.36	6.36	6.36	6.36	6.36	6.36	6.36	6.36	0.00
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 10 dB	0.16	3.41	8.18	2.98	6.39	0.40	6.44	5.88	5.39	9.25	8.85	2.89
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 20 dB	4.79	6.55	2.00	7.58	3.58	5.14	3.75	5.38	6.86	7.19	5.28	1.80

Avispa																					
TEST												Porcentajes de similitud (%)						MEDIA	Desviacion Estandar		

Porcentaje de Solapamiento 30%, ventana 1024, sin ruido	2.22	2.22	2.22	2.22	2.22	2.22	2.22	2.22	2.22	2.22	2.22	2.22	0.00
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 10 dB	5.01	3.22	5.89	1.94	3.50	1.28	4.59	4.01	3.12	3.31	3.59	1.37	
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 20 dB	3.31	2.91	2.24	1.93	2.22	3.14	1.93	3.94	2.08	3.07	2.68	0.69	
Porcentaje de Solapamiento 30%, ventana 2048, sin ruido	8.54	8.54	8.54	8.54	8.54	8.54	8.54	8.54	8.54	8.54	8.54	0.00	
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 10 dB	7.10	6.82	6.55	3.89	8.03	8.23	8.08	6.69	4.60	7.77	6.78	1.47	
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 20 dB	7.67	7.67	8.95	7.53	8.39	8.08	8.82	9.59	8.39	8.19	8.33	0.65	
Porcentaje de Solapamiento 60%, ventana 1024, sin ruido	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.0	
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 10 dB	1.45	1.23	1.31	1.63	1.87	0.47	2.61	1.29	1.39	0.92	1.42	0.5	
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 20 dB	0.36	9.71	0.76	0.76	1.15	0.17	0.89	1.05	0.43	0.66	0.59	0.4	
Porcentaje de Solapamiento 60%, ventana 2048, sin ruido	8.71	8.71	8.71	8.71	8.71	8.71	8.71	8.71	8.71	8.71	8.71	0.0	
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 10 dB	7.17	8.65	9.36	9.93	8.36	9.77	8.29	9.56	8.38	9.48	8.89	0.8	
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 20dB	9.69	9.11	8.94	9.46	9.72	9.60	8.94	9.16	9.74	8.56	9.29	0.4	
Porcentaje de Solapamiento 90%, ventana 1024, sin ruido	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	0.0	
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 10 dB	1.32	1.97	2.26	2.08	1.97	1.87	3.01	2.21	2.58	2.41	2.17	0.4	
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 20 dB	1.24	1.88	1.52	1.72	1.70	1.52	1.51	1.55	1.77	1.62	1.60	0.1	
Porcentaje de Solapamiento 90%, ventana 2048, sin ruido	9.76	9.76	9.76	9.76	9.76	9.76	9.76	9.76	9.76	9.76	9.76	0.0	
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 10 dB	8.00	9.58	9.00	9.16	9.23	9.09	8.76	9.71	9.01	9.42	9.10	0.4	
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 20 dB	9.69	9.89	9.77	9.75	9.85	9.57	9.46	9.57	9.78	8.97	9.63	0.2	

Langosta																					
<i>TEST</i>	<i>Porcentajes de similitud (%)</i>										<i>EDIA</i>	<i>Desviacion Estandar</i>									
Porcentaje de Solapamiento 30%, ventana 1024, sin ruido	8.32	8.32	8.32	8.32	8.32	8.32	8.32	8.32	8.32	8.32	8.32	8.32	0.00								
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 10 dB	9.74	8.41	9.63	4.78	2.00	0.86	7.42	1.57	8.94	9.06	9.24	7.48									
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 20 dB	0.27	9.70	5.49	7.56	5.09	7.34	5.88	6.31	7.61	6.24	2.15	6.45									
Porcentaje de Solapamiento 30%, ventana 2048, sin ruido	7.54	7.54	7.54	7.54	7.54	7.54	7.54	7.54	7.54	7.54	7.54	0.00									
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 10 dB	6.59	1.41	2.02	5.77	4.63	8.67	6.93	3.56	1.60	2.31	2.35	6.82									
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 20 dB	2.44	5.59	7.92	6.65	0.43	8.05	9.28	8.81	9.69	4.94	4.38	13.15									
Porcentaje de Solapamiento 60%, ventana 1024, sin ruido	8.17	8.17	8.17	8.17	8.17	8.17	8.17	8.17	8.17	8.17	8.17	0.00									
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 10 dB	9.77	6.56	0.57	1.33	7.53	0.41	6.30	5.91	2.25	8.97	6.96	9.64									
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 20 dB	8.83	1.07	8.04	2.54	7.34	1.63	2.29	5.18	8.02	9.32	4.43	4.60									
Porcentaje de Solapamiento 60%, ventana 2048, sin ruido	3.12	3.12	3.12	3.12	3.12	3.12	3.12	3.12	3.12	3.12	3.12	0.00									
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 10 dB	2.04	9.45	1.11	6.02	9.99	6.02	6.70	1.14	6.49	1.84	5.08	4.36									
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 20dB	0.60	8.21	6.52	6.90	3.65	8.43	2.90	9.68	5.61	1.90	1.44	4.19									
Porcentaje de Solapamiento 90%, ventana 1024, sin ruido	7.36	7.36	7.36	7.36	7.36	7.36	7.36	7.36	7.36	7.36	7.36	0.00									
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 10 dB	4.28	0.98	1.47	7.50	8.68	5.55	4.94	5.21	9.37	5.71	7.37	6.17									
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 20 dB	3.40	6.42	8.90	5.83	4.79	2.93	9.44	6.08	8.69	3.39	5.99	2.41									
Porcentaje de Solapamiento 90%, ventana 2048, sin ruido	8.96	8.96	8.96	8.96	8.96	8.96	8.96	8.96	8.96	8.96	8.96	0.00									
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 10 dB	4.75	4.99	8.55	4.06	3.25	5.74	1.81	7.68	6.54	9.23	2.66	4.01									
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 20 dB	9.65	0.10	6.18	2.86	1.73	3.80	5.40	6.47	4.16	4.57	4.49	2.67									

Saltamontes													
<i>TEST</i>	<i>Porcentajes de similitud (%)</i>										<i>MEDIA</i>	<i>Desviacion Estandar</i>	
Porcentaje de Solapamiento 30%, ventana 1024, sin ruido	7.68	7.68	7.68	7.68	7.68	7.68	7.68	7.68	7.68	7.68	7.68	7.68	0.00
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 10 dB	9.33	9.79	9.01	8.20	6.24	7.69	8.99	7.74	8.54	9.36	8.49	1.06	
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 20 dB	8.39	7.99	7.60	8.12	7.68	6.82	7.68	7.68	5.83	6.82	7.46	0.76	
Porcentaje de Solapamiento 30%, ventana 2048, sin ruido	8.69	8.69	8.69	8.69	8.69	8.69	8.69	8.69	8.69	8.69	8.69	0.00	
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 10 dB	6.71	9.60	8.53	6.32	9.95	8.05	9.38	8.86	9.60	8.62	8.56	1.23	
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 20 dB	8.89	8.77	8.47	8.95	9.32	8.35	8.26	8.26	8.20	8.64	8.61	0.37	
Porcentaje de Solapamiento 60%, ventana 1024, sin ruido	8.80	8.80	8.80	8.80	8.80	8.80	8.80	8.80	8.80	8.80	8.80	0.00	
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 10 dB	8.80	8.05	9.29	6.08	8.86	9.11	8.68	8.46	7.14	8.97	8.34	1.01	
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 20 dB	8.68	8.80	8.80	8.80	8.80	8.80	8.68	8.46	8.80	8.96	8.76	0.13	
Porcentaje de Solapamiento 60%, ventana 2048, sin ruido	9.23	9.23	9.23	9.23	9.23	9.23	9.23	9.23	9.23	9.23	9.23	0.00	
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 10 dB	9.16	8.90	8.56	7.76	8.91	8.26	4.33	9.55	8.96	9.78	8.42	1.55	
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 20dB	9.51	9.54	9.69	9.42	9.08	8.54	9.90	8.13	8.87	9.71	9.24	0.57	
Porcentaje de Solapamiento 90%, ventana 1024, sin ruido	8.51	8.51	8.51	8.51	8.51	8.51	8.51	8.51	8.51	8.51	8.51	0.00	
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 10 dB	9.83	8.75	7.75	9.50	8.66	8.29	8.32	8.95	9.54	9.55	8.91	0.68	
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 20 dB	9.08	8.96	8.49	8.35	8.47	8.17	9.12	8.28	8.26	8.46	8.56	0.35	
Porcentaje de Solapamiento 90%, ventana 2048, sin ruido	8.65	8.65	8.65	8.65	8.65	8.65	8.65	8.65	8.65	8.65	8.65	0.00	

Porcentaje de Solapamiento 90%, ventana 2048, con SNR 10 dB	4.19	9.06	9.11	7.49	8.50	9.04	6.91	8.36	8.28	4.64	7.56	1.80
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 20 dB	8.76	9.04	8.53	8.08	8.98	8.38	8.15	8.02	8.03	8.95	8.49	0.42

Mosca												
<i>TEST</i>	<i>Porcentajes de similitud (%)</i>										<i>MEDIA</i>	<i>Desviacion Estandar</i>
Porcentaje de Solapamiento 30%, ventana 1024, sin ruido	8.97	8.97	8.97	8.97	8.97	8.97	8.97	8.97	8.97	8.97	8.97	0.00
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 10 dB	9.14	9.00	9.46	9.18	9.37	9.46	9.14	8.52	9.21	8.40	9.09	0.36
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 20 dB	9.10	8.97	8.97	9.07	8.98	9.00	9.07	8.97	9.00	9.41	9.05	0.13
Porcentaje de Solapamiento 30%, ventana 2048, sin ruido	9.77	9.77	9.77	9.77	9.77	9.77	9.77	9.77	9.77	9.77	9.77	0.00
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 10 dB	9.78	9.78	9.88	9.67	9.82	9.83	9.79	9.66	9.70	9.64	9.76	0.08
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 20 dB	9.95	9.78	9.84	9.75	9.79	9.72	9.85	9.73	9.77	9.71	9.79	0.07
Porcentaje de Solapamiento 60%, ventana 1024, sin ruido	9.12	9.12	9.12	9.12	9.12	9.12	9.12	9.12	9.12	9.12	9.12	0.00
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 10 dB	8.92	8.48	8.91	9.36	9.39	9.39	8.56	8.78	8.94	8.88	8.96	0.33
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 20 dB	9.04	9.16	9.05	9.11	9.03	9.30	9.35	8.91	8.94	9.49	9.14	0.19
Porcentaje de Solapamiento 60%, ventana 2048, sin ruido	9.62	9.62	9.62	9.62	9.62	9.62	9.62	9.62	9.62	9.62	9.62	0.00
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 10 dB	9.75	9.82	9.78	9.68	9.82	9.77	9.53	9.76	9.95	9.70	9.76	0.11
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 20dB	9.61	9.71	9.70	9.71	9.69	9.71	9.70	9.63	9.58	9.59	9.66	0.06
Porcentaje de Solapamiento 90%, ventana 1024, sin ruido	9.26	9.26	9.26	9.26	9.26	9.26	9.26	9.26	9.26	9.26	9.26	0.00
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 10 dB	9.24	8.99	8.97	9.29	9.12	9.63	9.06	9.23	9.36	9.07	9.19	0.20

Porcentaje de Solapamiento 90%, ventana 1024, con SNR 20 dB	9.31	9.16	9.35	9.43	9.21	9.27	9.26	9.25	9.42	9.32	9.30	0.09
Porcentaje de Solapamiento 90%, ventana 2048, sin ruido	9.93	9.93	9.93	9.93	9.93	9.93	9.93	9.93	9.93	9.93	9.93	0.00
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 10 dB	9.75	9.76	9.98	9.86	9.92	9.88	9.86	9.90	9.96	9.88	9.87	0.07
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 20 dB	9.88	9.90	9.89	9.83	9.84	9.98	9.88	9.90	9.95	9.91	9.90	0.04

<i>Gryllotalpa (Grillo topo)</i>												
<i>TEST</i>	<i>Porcentajes de similitud (%)</i>										<i>MEDIA</i>	<i>Desviacion Estandar</i>
Porcentaje de Solapamiento 30%, ventana 1024, sin ruido	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	8.84	0.00
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 10 dB	9.21	8.36	8.76	8.41	8.45	8.78	8.83	9.60	4.98	8.95	8.43	1.27
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 20 dB	9.44	9.66	8.93	8.71	9.32	8.97	8.71	9.32	9.05	9.87	9.20	0.39
Porcentaje de Solapamiento 30%, ventana 2048, sin ruido	8.12	8.12	8.12	8.12	8.12	8.12	8.12	8.12	8.12	8.12	8.12	0.00
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 10 dB	9.47	9.03	9.01	9.11	8.90	9.39	9.63	4.63	9.37	8.96	8.75	1.47
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 20 dB	8.25	8.62	8.72	8.60	8.12	8.45	8.41	8.54	8.21	8.38	8.43	0.20
Porcentaje de Solapamiento 60%, ventana 1024, sin ruido	8.79	8.79	8.79	8.79	8.79	8.79	8.79	8.79	8.79	8.79	8.79	0.00
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 10 dB	9.64	8.50	9.77	9.13	9.35	9.07	9.55	9.00	9.57	9.51	9.31	0.38
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 20 dB	8.76	8.92	9.77	8.87	9.13	9.40	9.01	9.05	9.52	8.56	9.10	0.37
Porcentaje de Solapamiento 60%, ventana 2048, sin ruido	8.58	8.58	8.58	8.58	8.58	8.58	8.58	8.58	8.58	8.58	8.58	0.00
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 10 dB	9.78	9.33	9.45	9.62	8.77	9.67	8.63	9.42	9.29	8.76	9.27	0.41
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 20dB	8.54	8.72	8.44	7.91	8.73	8.61	8.85	8.48	8.81	8.44	8.55	0.27

Porcentaje de Solapamiento 90%, ventana 1024, sin ruido	8.47	8.47	8.47	8.47	8.47	8.47	8.47	8.47	8.47	8.47	8.47	8.47	0.00
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 10 dB	9.46	9.50	8.89	9.17	9.21	9.59	9.94	9.81	9.64	9.53	9.48	9.48	0.31
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 20 dB	8.98	8.92	8.70	9.13	8.94	8.96	8.82	9.09	9.09	9.20	8.98	8.98	0.15
Porcentaje de Solapamiento 90%, ventana 2048, sin ruido	7.85	7.85	7.85	7.85	7.85	7.85	7.85	7.85	7.85	7.85	7.85	7.85	0.00
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 10 dB	7.16	9.86	9.53	9.86	9.41	9.87	8.73	9.83	8.76	9.53	9.25	9.25	0.85
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 20 dB	8.76	8.23	8.11	8.11	8.28	8.26	8.46	8.36	8.06	7.97	8.26	8.26	0.23

<i>Acheta Domesticus (Grillo)</i>													
TEST	Porcentajes de similitud (%)										EDIA	Desviación Estandar	
Porcentaje de Solapamiento 30%, ventana 1024, sin ruido	9.20	9.20	9.20	9.20	9.20	9.20	9.20	9.20	9.20	9.20	9.20	9.20	0.00
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 10 dB	9.69	R	9.96	3.37	9.12	1.83	6.11	8.86	0.85	4.61	2.44	2.44	23.51
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 20 dB	9.20	9.75	9.20	9.20	9.20	9.20	9.20	9.75	8.38	9.20	8.23	8.23	3.47
Porcentaje de Solapamiento 30%, ventana 2048, sin ruido	6.93	6.93	6.93	6.93	6.93	6.93	6.93	6.93	6.93	6.93	6.93	6.93	0.00
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 10 dB	5.37	0.57	R	2.86	5.31	R	R	1.05	3.47	R	5.86	5.86	42.27
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 20 dB	6.77	9.36	7.59	8.93	6.17	9.13	6.77	8.38	8.38	7.86	7.93	7.93	1.09
Porcentaje de Solapamiento 60%, ventana 1024, sin ruido	7.41	7.41	7.41	7.41	7.41	7.41	7.41	7.41	7.41	7.41	7.41	7.41	0.00
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 10 dB	1.33	0.64	R	3.04	7.90	0.25	0.81	3.06	8.75	8.59	9.44	9.44	25.89
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 20 dB	7.41	3.59	3.59	7.41	3.59	7.23	7.41	7.41	7.41	7.41	6.24	6.24	1.83
Porcentaje de Solapamiento 60%, ventana 2048, sin ruido	6.77	6.77	6.77	6.77	6.77	6.77	6.77	6.77	6.77	6.77	6.77	6.77	0.00

Porcentaje de Solapamiento 60%, ventana 2048, con SNR 10 dB	R	R	8.80	1.23	R	2.55	3.22	6.89	R	4.15	0.68	35.59
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 20dB	5.27	6.49	5.30	3.52	5.75	6.73	5.47	3.03	6.52	6.75	5.48	1.30
Porcentaje de Solapamiento 90%, ventana 1024, sin ruido	4.50	4.50	4.50	4.50	4.50	4.50	4.50	4.50	4.50	4.50	4.50	0.00
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 10 dB	9.04	8.00	9.92	0.88	2.17	6.39	6.04	4.96	8.51	2.21	8.81	6.20
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 20 dB	5.54	5.73	4.73	3.41	4.36	4.15	5.29	6.03	5.00	5.22	4.95	0.80
Porcentaje de Solapamiento 90%, ventana 2048, sin ruido	5.69	5.69	5.69	5.69	5.69	5.69	5.69	5.69	5.69	5.69	5.69	0.00
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 10 dB	R	5.79	R	0.48	R	9.38	R	6.87	1.41	2.18	3.61	29.79
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 20 dB	5.80	6.12	4.69	6.58	4.92	4.24	5.25	5.60	5.91	6.23	5.53	0.74

<i>Gryllus Campestris (Grillo)</i>													
<i>TEST</i>	<i>Porcentajes de similitud (%)</i>										<i>MEDIA</i>	<i>Desviacion Estandar</i>	
Porcentaje de Solapamiento 30%, ventana 1024, sin ruido	7.71	7.71	7.71	7.71	7.71	7.71	7.71	7.71	7.71	7.71	7.71	7.71	0.00
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 10 dB	5.82	9.16	4.18	0.24	3.56	7.77	4.18	5.26	1.47	8.73	3.04	11.24	
Porcentaje de Solapamiento 30%, ventana 1024, con SNR 20 dB	8.40	9.16	1.91	9.39	2.23	0.55	7.71	7.68	7.71	9.39	9.41	1.68	
Porcentaje de Solapamiento 30%, ventana 2048, sin ruido	7.08	7.08	7.08	7.08	7.08	7.08	7.08	7.08	7.08	7.08	7.08	0.00	
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 10 dB	9.25	4.91	4.18	3.79	9.01	7.34	9.12	4.04	4.30	5.85	0.18	13.44	
Porcentaje de Solapamiento 30%, ventana 2048, con SNR 20 dB	5.69	7.59	2.66	7.51	8.00	4.37	4.50	6.21	7.48	8.89	6.29	1.96	
Porcentaje de Solapamiento 60%, ventana 1024, sin ruido	3.05	3.05	3.05	3.05	3.05	3.05	3.05	3.05	3.05	3.05	3.05	0.00	
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 10 dB	2.41	9.76	4.61	8.26	6.86	7.35	2.74	6.84	1.14	7.91	1.79	5.70	

Porcentaje de Solapamiento 30%, ventana 2048, con SNR 20 dB	6.70	9.36	5.78	3.28	1.56	8.80	7.94	4.68	8.77	8.35	6.52	2.64
Porcentaje de Solapamiento 60%, ventana 1024, sin ruido	8.41	8.41	8.41	8.41	8.41	8.41	8.41	8.41	8.41	8.41	8.41	0.00
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 10 dB	8.66	1.63	2.40	9.82	4.85	7.58	7.89	8.66	4.48	6.25	9.22	3.74
Porcentaje de Solapamiento 60%, ventana 1024, con SNR 20 dB	6.73	1.71	7.85	3.41	5.19	8.82	5.66	3.59	2.28	6.24	5.15	2.36
Porcentaje de Solapamiento 60%, ventana 2048, sin ruido	9.28	9.28	9.28	9.28	9.28	9.28	9.28	9.28	9.28	9.28	9.28	0.00
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 10 dB	2.70	0.86	7.74	6.87	0.57	9.06	3.31	0.74	7.69	1.60	8.11	3.49
Porcentaje de Solapamiento 60%, ventana 2048, con SNR 20dB	6.01	4.30	6.12	7.22	4.79	4.82	8.27	6.76	6.74	6.68	6.17	1.23
Porcentaje de Solapamiento 90%, ventana 1024, sin ruido	4.73	4.73	4.73	4.73	4.73	4.73	4.73	4.73	4.73	4.73	4.73	0.00
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 10 dB	6.62	3.55	0.93	6.91	9.25	9.69	7.79	8.69	9.76	8.29	8.15	2.09
Porcentaje de Solapamiento 90%, ventana 1024, con SNR 20 dB	3.62	0.92	2.04	1.68	1.87	1.38	2.37	0.52	3.54	4.71	2.27	1.32
Porcentaje de Solapamiento 90%, ventana 2048, sin ruido	7.52	7.52	7.52	7.52	7.52	7.52	7.52	7.52	7.52	7.52	7.52	0.00
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 10 dB	4.69	7.91	5.53	6.44	7.20	9.96	4.45	5.31	6.17	7.81	7.55	19.59
Porcentaje de Solapamiento 90%, ventana 2048, con SNR 20 dB	4.30	5.15	5.01	4.35	3.39	6.26	3.91	1.53	4.25	4.29	4.24	1.24

Anexo 2: Detalle de la Segunda Ronda de Test de Optimización

Mosca

Media: 519,74 Hz Desviación típica: 8,88 Hz

TEST		Porcentajes de similitud (%)								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	99,57	99,81	99,43	99,03	99,08	99,09	98,73	99,48	99,27
Test 2	Muestras 30, SNR 10-15dB y Ventana 2048	99,67	99,40	99,83	99,67	97,58	100,00	99,67	99,78	99,45
Test 3	Muestras 40, SNR 5dB y Ventana 2048	99,78	99,68	99,26	99,80	99,07	99,76	99,90	99,16	99,55
Test 4	Muestras 40, SNR 10dB y Ventana 2048	99,58	99,59	99,75	99,83	99,58	99,65	99,03	99,47	99,56
Test 5	Muestras 40, SNR 5dB y Ventana 4096	99,69	99,73	99,69	99,69	99,69	99,69	99,69	99,69	99,68
Test 6	Muestras 40, SNR 10dB y Ventana 4096	99,69	99,73	99,73	99,69	99,69	99,69	99,69	99,69	99,70

TEST		Funcion de similitud								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	5,75	2,59	8,28	13,04	10,39	12,28	22,93	7,63	8,80
Test 2	Muestras 30, SNR 10-15dB y Ventana 2048	4,79	9,79	2,44	4,79	57,04	1,00	4,37	2,49	4,91
Test 3	Muestras 40, SNR 5dB y Ventana 2048	2,90	4,70	9,44	2,89	7,51	3,62	1,95	14,96	4,83
Test 4	Muestras 40, SNR 10dB y Ventana 2048	6,54	6,03	3,82	2,76	6,54	5,19	18,73	8,57	6,23
Test 5	Muestras 40, SNR 5dB y Ventana 4096	2,69	2,61	2,69	2,69	2,69	2,69	2,69	4,01	2,82
Test 6	Muestras 40, SNR 10dB y Ventana 4096	2,69	2,61	2,61	2,69	2,69	2,69	2,69	2,69	2,67

Gryllotalpa Gryllotalpa

Media: 2181,4 Hz Desviación típica: 20,57 Hz

TEST		Porcentajes de similitud (%)								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	96,69	97,37	96,68	96,87	97,29	97,84	96,66	96,88	97,04
Test 2	Muestras 30, SNR 10dB y Ventana 2048	98,04	98,74	99,14	98,34	97,65	99,52	98,71	98,08	98,53
Test 3	Muestras 40, SNR 5dB y Ventana 2048	97,88	97,93	98,63	96,49	97,92	97,55	97,38	98,47	97,78
Test 4	Muestras 40, SNR 10dB y Ventana 2048	98,87	99,69	98,65	98,45	99,17	99,07	98,85	99,37	99,02
Test 5	Muestras 40, SNR 5dB y Ventana 4096	NR	NR	96,22	67,24	74,41	NR	NR	NR	78,38
Test 6	Muestras 40, SNR 10dB y Ventana 4096	78,47	83,03	85,97	NR	NR	98,22	NR	NR	86,12

TEST		Funcion de similitud								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	110,38	41,15	124,49	140,32	75,58	68,43	52,47	22,30	68,42
Test 2	Muestras 30, SNR 10dB y Ventana 2048	27,70	28,90	16,72	20,88	68,72	7,54	9,88	35,54	21,79
Test 3	Muestras 40, SNR 5dB y Ventana 2048	66,96	26,87	21,04	128,27	17,83	20,86	16,78	22,70	30,18
Test 4	Muestras 40, SNR 10dB y Ventana 2048	24,11	3,39	26,78	26,87	9,81	18,62	13,91	9,34	13,90
Test 5	Muestras 40, SNR 5dB y Ventana 4096	NR	NR	25,07	4334,90	2620,04	NR	NR	NR	657,90
Test 6	Muestras 40, SNR 10dB y Ventana 4096	2146,87	1144,26	908,52	NR	NR	43,12	NR	NR	556,99

Cicada Mordoganensis

Media: 4167,0 Hz

Desviación típica: 83,06 Hz

TEST		Porcentajes de similitud (%)								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	NR	NR	NR	NR	NR	NR	NR	NR	#jNUM!
Test 2	Muestras 30, SNR 10-15dB y Ventana 2048	NR	NR	NR	54,62	80,31	98,77	62,25	58,20	69,05
Test 3	Muestras 40, SNR 5dB y Ventana 2048	NR	NR	NR	NR	NR	NR	NR	NR	#jNUM!
Test 4	Muestras 40, SNR 10-15 dB y Ventana 2048	NR	NR	NR	NR	NR	NR	70,29	50,96	59,85
Test 5	Muestras 40, SNR 5dB y Ventana 4096	NR	NR	NR	NR	NR	NR	NR	NR	#jNUM!
Test 6	Muestras 40, SNR 10dB y Ventana 4096	NR	NR	NR	NR	88,25	NR	50,50	95,69	75,27

TEST		Funcion de similitud								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	NR	NR	NR	NR	NR	NR	NR	NR	#jNUM!
Test 2	Muestras 30, SNR 10dB y Ventana 2048	NR	NR	NR	8560,61	2578,01	23,47	9406,77	10915,00	2213,90
Test 3	Muestras 40, SNR 5dB y Ventana 2048	NR	NR	NR	NR	NR	NR	NR	NR	#jNUM!
Test 4	Muestras 40, SNR 10-15dB y Ventana 2048	NR	NR	NR	NR	NR	NR	4215,57	16325,79	8295,94
Test 5	Muestras 40, SNR 5dB y Ventana 4096	NR	NR	NR	NR	NR	NR	NR	NR	#jNUM!
Test 6	Muestras 40, SNR 10dB y Ventana 4096	NR	NR	NR	NR	407,56	NR	#####	256,15	1289,47

Langosta

Media: 7291,1 Hz

Desviación típica: 70,54 Hz

TEST		Porcentajes de similitud (%)								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	NR	NR	NR	NR	NR	NR	NR	NR	#jNUM!
Test 2	Muestras 30, SNR 10dB y Ventana 2048	90,91	93,22	94,00	87,04	91,05	88,60	87,10	90,36	90,25
Test 3	Muestras 40, SNR 5dB y Ventana 2048	NR	NR	NR	82,95	NR	NR	NR	NR	82,95
Test 4	Muestras 40, SNR 10dB y Ventana 2048	91,64	86,74	92,62	93,51	91,70	78,05	87,43	84,92	88,19
Test 5	Muestras 40, SNR 5dB y Ventana 4096	79,22	77,27	78,39	85,87	NR	80,09	78,71	74,75	79,12
Test 6	Muestras 40, SNR 10dB y Ventana 4096	85,30	90,41	84,50	90,58	90,18	80,76	89,29	87,96	87,31

TEST		Formula de similitud								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	NR	NR	NR	NR	NR	NR	NR	NR	#jNUM!
Test 2	Muestras 30, SNR 10dB y Ventana 2048	2326,43	1542,74	1629,44	8266,20	229,41	5077,76	4350,47	769,81	1925,05
Test 3	Muestras 40, SNR 5dB y Ventana 2048	NR	NR	NR	3003,89	NR	NR	NR	NR	3003,89
Test 4	Muestras 40, SNR 10dB y Ventana 2048	1848,73	2368,30	2659,75	296,78	276,70	#####	3244,99	#####	2252,17
Test 5	Muestras 40, SNR 5dB y Ventana 4096	1922,43	#####	#####	5135,98	NR	4119,18	#####	4319,35	7020,15
Test 6	Muestras 40, SNR 10dB y Ventana 4096	6646,11	1263,83	6983,23	2223,49	1813,55	#####	1666,77	3726,43	3418,10

Saltamontes

Media: 7710,9 Hz

Desviación típica: 141,32 Hz

TEST		Porcentajes de similitud (%)								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	75,52	84,46	89,43	87,43	NR	97,60	99,05	95,52	89,51
Test 2	Muestras 30, SNR 10dB y Ventana 2048	97,55	92,64	98,36	90,06	98,44	98,03	98,52	98,29	96,44
Test 3	Muestras 40, SNR 5dB y Ventana 2048	89,49	88,82	82,18	81,27	89,62	83,19	93,74	96,18	87,91
Test 4	Muestras 40, SNR 10dB y Ventana 2048	95,06	99,33	98,30	99,56	96,35	98,55	97,91	98,91	97,99
Test 5	Muestras 40, SNR 5dB y Ventana 4096	74,65	95,76	93,15	90,08	81,07	92,38	92,31	95,45	89,06
Test 6	Muestras 40, SNR 10dB y Ventana 4096	97,54	98,02	98,33	91,16	98,08	89,98	87,43	97,48	94,66

TEST		Funcion de similitud								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	31449,23	#####	6652,57	6550,40	NR	440,20	97,75	1892,17	2827,26
Test 2	Muestras 30, SNR 10dB y Ventana 2048	402,35	3459,33	80,23	3623,64	241,59	160,40	18,79	21,46	223,93
Test 3	Muestras 40, SNR 5dB y Ventana 2048	3925,60	5864,16	#####	#####	2661,40	#####	199,99	684,38	3589,68
Test 4	Muestras 40, SNR 10dB y Ventana 2048	1454,53	55,63	296,16	19,57	985,88	203,81	195,98	122,84	197,00
Test 5	Muestras 40, SNR 5dB y Ventana 4096	44978,91	757,87	400,79	6196,28	#####	445,90	3819,17	224,34	2263,79
Test 6	Muestras 40, SNR 10dB y Ventana 4096	555,45	332,38	52,46	5820,55	370,93	2492,30	8130,89	55,16	625,30

Mosquito

Media: 290,23 Hz

Desviación típica: 24,88 Hz

TEST		Porcentajes de similitud (%)								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	NR	NR	NR	NR	NR	99,61	67,81	NR	82,19
Test 2	Muestras 30, SNR 10dB y Ventana 2048	79,55	91,81	81,22	81,99	67,23	67,01	99,27	81,64	80,57
Test 3	Muestras 40, SNR 5dB y Ventana 2048	65,56	NR	NR	NR	92,96	NR	NR	73,60	76,55
Test 4	Muestras 40, SNR 10dB y Ventana 2048	78,68	80,34	79,46	77,90	97,21	79,88	97,71	99,19	85,83
Test 5	Muestras 40, SNR 5dB y Ventana 4096	83,40	98,30	83,16	89,14	64,52	98,39	84,32	98,20	86,71
Test 6	Muestras 40, SNR 10dB y Ventana 4096	72,47	73,93	63,66	96,32	84,70	82,52	83,97	73,41	78,32
TEST		Porcentajes de similitud (%)								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	NR	NR	NR	NR	NR	0,00	253,57	NR	#####
Test 2	Muestras 30, SNR 10dB y Ventana 2048	5,85	9,13	21,30	101,03	198,69	156,04	0,63	79,93	25,52
Test 3	Muestras 40, SNR 5dB y Ventana 2048	249,45	NR	NR	NR	27,86	NR	NR	218,07	114,86
Test 4	Muestras 40, SNR 10dB y Ventana 2048	46,09	3,92	77,53	71,86	6,86	96,86	2,19	0,51	12,85
Test 5	Muestras 40, SNR 5dB y Ventana 4096	14,14	3,06	28,38	19,17	213,46	2,50	5,04	2,36	10,52
Test 6	Muestras 40, SNR 10dB y Ventana 4096	61,17	118,81	163,77	10,20	3,28	42,02	11,96	91,85	34,13

Abeja

Media: 186,98 Hz

Desviación típica: 22,54 Hz

TEST		Porcentajes de similitud (%)								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	95,45	98,71	97,83	95,28	98,49	99,27	98,67	99,00	97,83
Test 2	Muestras 30, SNR 10dB y Ventana 2048	98,53	98,51	98,52	97,07	98,35	98,71	99,51	98,77	98,49
Test 3	Muestras 40, SNR 5dB y Ventana 2048	94,55	93,46	92,18	97,43	93,43	94,63	95,31	94,19	94,39
Test 4	Muestras 40, SNR 10dB y Ventana 2048	93,20	95,54	98,12	96,06	94,24	96,27	95,86	95,35	95,57
Test 5	Muestras 40, SNR 5dB y Ventana 4096	95,64	93,57	98,60	97,88	97,26	99,03	97,85	97,07	97,10
Test 6	Muestras 40, SNR 10dB y Ventana 4096	97,57	99,96	93,85	98,18	99,33	99,75	94,61	98,43	97,69
TEST		Formula de similitud								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	15,71	2,01	3,53	17,27	2,02	0,62	1,87	0,84	2,80
Test 2	Muestras 30, SNR 10dB y Ventana 2048	2,41	2,67	2,31	3,78	3,20	0,35	0,26	1,72	1,52
Test 3	Muestras 40, SNR 5dB y Ventana 2048	3,07	17,05	0,54	2,38	25,77	20,25	0,68	19,67	5,12
Test 4	Muestras 40, SNR 10dB y Ventana 2048	15,34	5,69	2,94	5,70	12,81	1,81	2,86	10,88	5,66
Test 5	Muestras 40, SNR 5dB y Ventana 4096	13,48	23,85	1,27	2,22	5,50	0,81	2,07	9,60	4,10
Test 6	Muestras 40, SNR 10dB y Ventana 4096	0,26	0,00	25,80	0,57	0,54	0,04	21,61	0,78	0,49

Avispa

Media: 238,04 Hz

Desviación típica: 18,58 Hz

TEST		Porcentajes de similitud (%)								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	98,35	95,07	95,19	79,52	94,96	98,75	80,99	99,71	92,50
Test 2	Muestras 30, SNR 10dB y Ventana 2048	98,13	93,81	96,07	85,25	99,30	95,83	94,57	97,30	94,94
Test 3	Muestras 40, SNR 5dB y Ventana 2048	89,45	89,04	93,77	94,35	84,45	95,54	94,34	88,02	91,05
Test 4	Muestras 40, SNR 10dB y Ventana 2048	93,95	92,60	89,79	95,03	96,10	91,30	89,86	94,29	92,84
Test 5	Muestras 40, SNR 5dB y Ventana 4096	98,95	99,55	96,88	97,83	99,28	97,44	98,45	98,26	98,33
Test 6	Muestras 40, SNR 10dB y Ventana 4096	99,64	99,12	99,81	98,95	99,55	99,25	98,62	99,04	99,25
TEST		Formula de similitud								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	3,26	15,49	6,47	28,97	6,80	1,77	107,07	0,10	5,74
Test 2	Muestras 30, SNR 10dB y Ventana 2048	0,52	14,59	3,23	20,75	0,51	6,57	25,99	4,03	4,54
Test 3	Muestras 40, SNR 5dB y Ventana 2048	77,83	48,48	38,59	26,26	59,08	21,68	33,99	61,89	42,33
Test 4	Muestras 40, SNR 10dB y Ventana 2048	32,87	40,76	33,92	22,21	16,16	54,53	47,94	30,43	32,67
Test 5	Muestras 40, SNR 5dB y Ventana 4096	0,82	0,24	3,95	3,13	0,52	1,81	1,54	1,64	1,24
Test 6	Muestras 40, SNR 10dB y Ventana 4096	0,03	0,60	0,02	0,00	0,00	0,53	0,42	0,29	#####

Cicada Cretensis

Media: 5511,8 Hz

Desviación típica: 239,38 Hz

TEST		Porcentajes de similitud (%)								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	78,91	88,80	74,32	79,65	92,62	85,44	79,04	84,33	82,70
Test 2	Muestras 30, SNR 10dB y Ventana 2048	77,07	85,08	89,64	92,06	97,58	91,68	94,74	79,59	88,16
Test 3	Muestras 40, SNR 5dB y Ventana 2048	91,01	78,44	92,01	89,40	92,41	88,93	81,78	88,71	87,70
Test 4	Muestras 40, SNR 10dB y Ventana 2048	95,06	94,42	87,78	94,47	98,11	98,19	97,97	93,81	94,92
Test 5	Muestras 40, SNR 5dB y Ventana 4096	87,57	81,50	89,56	85,33	79,17	93,65	85,38	90,66	86,49
Test 6	Muestras 40, SNR 10dB y Ventana 4096	94,57	94,39	95,47	96,92	91,76	96,29	94,70	90,41	94,29
TEST		Funcion de similitud								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	#####	6418,12	33103,61	#####	2790,53	#####	#####	9448,68	12407,13
Test 2	Muestras 30, SNR 10dB y Ventana 2048	#####	#####	5182,08	3217,66	107,25	3269,42	1360,53	#####	3869,71
Test 3	Muestras 40, SNR 5dB y Ventana 2048	4094,59	#####	3119,21	5777,01	2798,15	6186,25	#####	6112,68	6443,98
Test 4	Muestras 40, SNR 10dB y Ventana 2048	322,14	1441,06	6499,90	1592,40	201,69	170,53	207,23	1621,99	696,76
Test 5	Muestras 40, SNR 5dB y Ventana 4096	7656,42	#####	5339,98	#####	20393,58	1844,04	#####	4435,67	7833,21
Test 6	Muestras 40, SNR 10dB y Ventana 4096	1557,59	1168,98	1067,86	219,90	3512,67	586,71	787,66	4742,88	1160,60

Acherontia Atropos

Media: 5823,8 Hz

Desviación típica: 115,33 Hz

TEST		Porcentajes de similitud (%)								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	83,61	NR	81,08	89,49	76,27	77,34	NR	83,09	81,70
Test 2	Muestras 30, SNR 10-15dB y Ventana 2048	87,29	86,09	82,52	85,00	94,76	89,27	87,52	89,79	87,71
Test 3	Muestras 40, SNR 5dB y Ventana 2048	NR	80,28	80,84	82,00	83,81	NR	79,95	84,74	81,92
Test 4	Muestras 40, SNR 10dB y Ventana 2048	88,44	90,07	86,42	89,87	82,99	NR	81,36	NR	86,46
Test 5	Muestras 40, SNR 5dB y Ventana 4096	88,90	87,01	90,96	88,42	89,16	92,39	87,90	91,20	89,48
Test 6	Muestras 40, SNR 10dB y Ventana 4096	91,03	91,61	94,49	90,68	89,13	94,82	88,60	97,98	92,25
TEST		Funcion de similitud								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	7324,68	NR	14075,11	1712,34	20505,06	#####	NR	6592,87	8314,49
Test 2	Muestras 30, SNR 10-15dB y Ventana 2048	6769,12	6683,29	8420,56	10495,21	1240,46	2559,50	4566,47	3552,81	4615,49
Test 3	Muestras 40, SNR 5dB y Ventana 2048	NR	810,53	10255,12	10382,25	2031,66	NR	#####	6325,01	4922,20
Test 4	Muestras 40, SNR 10dB y Ventana 2048	2495,74	116,39	8350,88	388,54	7009,21	NR	8884,09	NR	1971,35
Test 5	Muestras 40, SNR 5dB y Ventana 4096	1935,33	2029,37	3177,36	4088,35	4027,54	2078,42	1131,26	1339,35	2245,79
Test 6	Muestras 40, SNR 10dB y Ventana 4096	2581,31	1425,63	667,40	2744,12	4218,59	1101,99	4145,74	37,64	1219,44

Cicada Barbara

Media: 6061,6 Hz

Desviación típica: 155,42 Hz

TEST		Porcentajes de similitud (%)								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	94,02	92,58	98,19	95,26	91,97	95,67	93,36	92,11	94,12
Test 2	Muestras 30, SNR 10dB y Ventana 2048	98,26	96,59	97,16	97,45	95,89	94,58	97,00	96,08	96,62
Test 3	Muestras 40, SNR 5dB y Ventana 2048	92,38	93,41	93,25	97,02	86,27	92,26	89,04	83,85	90,85
Test 4	Muestras 40, SNR 10dB y Ventana 2048	99,42	96,41	94,90	98,85	96,68	98,99	94,82	93,03	96,61
Test 5	Muestras 40, SNR 5dB y Ventana 4096	94,21	86,80	87,71	92,56	86,88	93,07	89,77	88,39	89,88
Test 6	Muestras 40, SNR 10dB y Ventana 4096	91,90	98,52	98,20	93,96	87,09	91,94	92,83	97,68	93,94
TEST		Funcion de similitud								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	1495,29	2815,09	140,76	1155,75	3137,63	987,07	1707,63	1909,79	1273,47
Test 2	Muestras 30, SNR 10dB y Ventana 2048	31,04	632,23	446,12	325,67	464,41	1549,02	390,89	823,20	400,36
Test 3	Muestras 40, SNR 5dB y Ventana 2048	2276,86	1699,49	568,82	38,94	9373,64	895,18	3569,70	9026,66	1481,24
Test 4	Muestras 40, SNR 10dB y Ventana 2048	14,48	349,91	1182,89	80,84	580,66	64,79	814,35	1846,31	268,98
Test 5	Muestras 40, SNR 5dB y Ventana 4096	1347,26	7571,46	5966,25	1633,50	6899,04	2239,83	5179,02	6816,89	3906,48
Test 6	Muestras 40, SNR 10dB y Ventana 4096	2482,12	130,38	182,65	1458,71	6462,28	3293,22	2647,51	285,72	1041,84

Eumodicoryllus Burgalensis

Media: 4880,5 Hz

Desviación típica: 63,47 Hz

TEST		Porcentajes de similitud (%)								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	88,10	81,30	76,25	83,59	65,74	64,18	72,55	73,82	75,28
Test 2	Muestras 30, SNR 10dB y Ventana 2048	87,19	83,37	66,88	72,93	87,94	97,91	79,96	88,97	82,62
Test 3	Muestras 40, SNR 5dB y Ventana 2048	78,65	73,06	67,70	71,16	87,53	65,29	74,60	78,64	74,29
Test 4	Muestras 40, SNR 10dB y Ventana 2048	71,60	76,06	81,30	74,61	84,85	77,83	85,22	78,73	78,65
Test 5	Muestras 40, SNR 5dB y Ventana 4096	80,76	64,61	59,73	83,32	58,68	74,93	71,17	68,46	69,69
Test 6	Muestras 40, SNR 10dB y Ventana 4096	89,89	75,90	79,90	86,94	77,97	69,18	78,28	73,78	78,73
TEST		Funcion de similitud								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	24,80	568,87	83,06	81,65	460,97	1213,90	1361,62	775,26	294,46
Test 2	Muestras 30, SNR 10dB y Ventana 2048	490,05	580,90	2401,75	211,19	457,10	32,85	1343,45	162,44	384,04
Test 3	Muestras 40, SNR 5dB y Ventana 2048	1015,98	1358,44	480,45	488,69	196,00	1816,47	324,31	714,73	635,91
Test 4	Muestras 40, SNR 10dB y Ventana 2048	991,25	549,40	277,17	1770,21	827,23	212,10	263,04	204,47	473,38
Test 5	Muestras 40, SNR 5dB y Ventana 4096	1261,85	2575,90	2500,32	757,71	937,50	1032,62	1601,75	712,62	1270,87
Test 6	Muestras 40, SNR 10dB y Ventana 4096	49,16	394,75	439,63	293,66	494,01	805,10	265,45	483,79	326,12

Acheta Domesticus

Media: 4628,1 Hz

Desviación típica: 32,11 Hz

TEST		Porcentajes de similitud (%)								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	74,54	52,49	NR	79,41	NR	77,08	NR	NR	69,96
Test 2	Muestras 30, SNR 10dB y Ventana 2048	97,16	96,35	97,21	96,73	94,45	94,77	97,40	98,59	96,57
Test 3	Muestras 40, SNR 5dB y Ventana 2048	NR	NR	63,90	NR	NR	51,84	68,15	NR	60,89
Test 4	Muestras 40, SNR 10dB y Ventana 2048	97,49	96,49	96,21	96,37	94,61	97,28	96,65	96,75	96,48
Test 5	Muestras 40, SNR 5dB y Ventana 4096	71,35	NR	52,93	57,32	81,37	61,87	62,21	75,25	65,37
Test 6	Muestras 40, SNR 10dB y Ventana 4096	84,10	56,31	51,83	92,57	74,47	95,80	64,56	98,97	75,32
TEST		Funcion de similitud								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	412,80	403,44	NR	42,18	NR	386,99	NR	NR	228,34
Test 2	Muestras 30, SNR 10dB y Ventana 2048	50,65	58,07	40,77	29,13	20,52	141,32	44,68	20,25	41,71
Test 3	Muestras 40, SNR 5dB y Ventana 2048	NR	NR	2007,19	NR	NR	2329,26	1083,10	NR	1717,22
Test 4	Muestras 40, SNR 10dB y Ventana 2048	33,55	36,57	37,60	52,14	43,92	11,13	48,01	9,72	29,34
Test 5	Muestras 40, SNR 5dB y Ventana 4096	1020,43	NR	1753,86	1723,37	313,70	1653,29	1854,22	57,27	776,26
Test 6	Muestras 40, SNR 10dB y Ventana 4096	446,31	2295,91	2855,22	81,43	79,91	44,25	2202,01	10,20	256,82

Gryllus Campestris

Media: 4871,7 Hz

Desviación típica: 33,66 Hz

TEST		Porcentajes de similitud (%)								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	92,69	95,22	94,25	79,27	85,14	70,89	90,03	85,50	86,25
Test 2	Muestras 30, SNR 10dB y Ventana 2048	97,07	98,90	94,55	98,71	99,00	97,78	97,52	98,59	97,76
Test 3	Muestras 40, SNR 5dB y Ventana 2048	80,87	94,52	93,04	96,69	93,81	91,67	95,94	89,37	91,86
Test 4	Muestras 40, SNR 10dB y Ventana 2048	94,17	94,92	93,73	94,86	94,17	97,69	96,55	95,02	95,13
Test 5	Muestras 40, SNR 5dB y Ventana 4096	95,63	97,41	97,13	88,59	94,25	98,90	93,34	92,96	94,73
Test 6	Muestras 40, SNR 10dB y Ventana 4096	97,67	99,15	99,32	98,45	89,95	93,41	99,74	99,16	97,05
TEST		Funcion de similitud								MEDIA
Test 1	Muestras 30, SNR 5dB y Ventana 2048	118,24	28,50	29,74	335,52	158,50	1016,65	99,72	309,21	142,19
Test 2	Muestras 30, SNR 10dB y Ventana 2048	22,32	12,83	36,16	15,51	12,83	12,53	19,20	14,97	17,13
Test 3	Muestras 40, SNR 5dB y Ventana 2048	177,32	48,91	72,48	22,00	58,28	135,41	24,36	43,57	57,27
Test 4	Muestras 40, SNR 10dB y Ventana 2048	42,83	18,79	32,44	35,23	27,85	23,90	33,47	36,87	30,53
Test 5	Muestras 40, SNR 5dB y Ventana 4096	54,24	38,20	36,15	195,93	44,16	8,14	78,08	115,80	51,27
Test 6	Muestras 40, SNR 10dB y Ventana 4096	19,51	4,60	4,21	12,40	152,62	32,63	2,59	5,10	11,51

Anexo 3: Test finales de Matlab.

<i>Gryllus Campestris (Grillo)</i>									
TEST	<i>Gryllus Campestris. Porcentajes de similitud (%)</i>								MEDIA
20 Tramas y SNR 0 dB	46,73	49,58	33,71	52,93	0,00	49,85	38,39	30,01	37,65
20 Tramas y SNR 10 dB	50,19	64,15	79,59	73,23	74,97	90,65	94,06	95,02	77,73
20 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00
30 Tramas y SNR 0 dB	0,00	0,00	43,42	0,00	39,74	0,00	0,00	58,79	17,74
30 Tramas y SNR 10 dB	80,87	90,31	78,27	80,64	91,35	86,22	91,48	94,55	86,71
30 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00
TEST	<i>Acheta Domesticus. Porcentajes de similitud (%)</i>								MEDIA
20 Tramas y SNR 0 dB	5,13	8,19	5,02	22,96	0,00	8,89	4,11	0,00	6,79
20 Tramas y SNR 10 dB	8,58	22,55	37,99	36,14	33,36	49,05	52,45	53,42	36,69
20 Tramas y sin ruido	58,40	58,40	58,40	58,40	58,40	58,40	58,40	58,40	58,40
30 Tramas y SNR 0 dB	0,00	0,00	5,84	0,00	6,03	0,00	0,00	17,62	3,69
30 Tramas y SNR 10 dB	40,71	51,87	37,10	40,33	50,18	45,05	53,62	53,38	46,53
30 Tramas y sin ruido	58,83	58,83	58,83	58,83	58,83	58,83	58,83	58,83	58,83
TEST	<i>Eumodicogryllus Burgalensis. Porcentajes de similitud (%)</i>								MEDIA
20 Tramas y SNR 0 dB	24,23	27,29	14,44	33,66	0,00	27,99	19,12	10,74	19,69
20 Tramas y SNR 10 dB	27,69	41,65	57,09	53,95	52,47	68,15	71,56	72,52	55,64
20 Tramas y sin ruido	77,50	77,50	77,50	77,50	77,50	77,50	77,50	77,50	77,50
30 Tramas y SNR 0 dB	0,00	0,00	30,93	0,00	27,24	0,00	0,00	43,07	12,65
30 Tramas y SNR 10 dB	66,15	77,31	62,55	65,77	75,62	70,50	78,99	78,82	71,96
30 Tramas y sin ruido	84,27	84,27	84,27	84,27	84,27	84,27	84,27	84,27	84,27

Eumodicryllus Burgalensis (Grillo)

TEST	Eumodicryllus Burgalensis. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	95,34	88,26	91,31	91,78	93,39	92,89	79,53	70,52	87,88
20 Tramas y SNR 10 dB	96,61	98,31	90,88	94,51	95,88	96,20	96,01	92,33	95,09
20 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00
30 Tramas y SNR 0 dB	97,18	86,68	96,80	89,40	85,25	81,27	83,43	81,92	87,74
30 Tramas y SNR 10 dB	99,01	94,94	97,93	90,81	96,08	99,08	98,99	95,88	96,59
30 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00

TEST	Gryllus Campestris. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	82,16	89,24	86,08	85,61	83,24	84,61	97,33	93,02	87,66
20 Tramas y SNR 10 dB	80,89	79,08	86,62	82,99	81,62	81,30	81,49	82,69	82,09
20 Tramas y sin ruido	77,50	77,50	77,50	77,50	77,50	77,50	77,50	77,50	77,50
30 Tramas y SNR 0 dB	86,80	97,45	86,83	93,43	97,74	93,77	95,92	94,42	93,29
30 Tramas y SNR 10 dB	85,27	89,33	82,20	91,96	81,72	83,36	85,28	88,40	85,94
30 Tramas y sin ruido	84,27	84,27	84,27	84,27	84,27	84,27	84,27	84,27	84,27

TEST	Acheta Domesticus. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	73,91	69,15	72,31	72,79	75,15	73,78	61,07	51,42	68,70
20 Tramas y SNR 10 dB	73,72	72,34	71,78	74,42	72,62	73,16	73,46	75,71	73,40
20 Tramas y sin ruido	70,65	70,65	70,65	70,65	70,65	70,65	70,65	70,65	70,65
30 Tramas y SNR 0 dB	72,03	61,24	72,00	65,40	60,95	57,69	58,56	58,56	63,30
30 Tramas y SNR 10 dB	73,56	69,50	76,00	66,87	74,01	75,47	73,55	70,43	72,42
30 Tramas y sin ruido	74,56	74,56	74,56	74,56	74,56	74,56	74,56	74,56	74,56

Acheta Domesticus (Grillo)

TEST	Acheta Domesticus. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	0,00	21,75	0,00	57,88	0,00	52,03	70,73	24,66	28,38
20 Tramas y SNR 10 dB	97,95	98,47	53,08	89,30	98,42	97,45	95,26	95,84	90,72
20 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00
30 Tramas y SNR 0 dB	34,17	18,09	44,84	57,24	0,00	57,99	0,00	54,16	33,31
30 Tramas y SNR 10 dB	97,46	93,78	95,65	97,51	98,87	71,06	96,41	97,08	93,48
30 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00

TEST	Gryllus Campestris. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	9,13	22,45	0,00	58,47	0,00	41,96	60,65	26,32	27,37
20 Tramas y SNR 10 dB	56,35	57,73	48,71	65,76	58,26	60,51	61,20	61,70	58,78
20 Tramas y sin ruido	58,40	58,40	58,40	58,40	58,40	58,40	58,40	58,40	58,40
30 Tramas y SNR 0 dB	33,56	29,69	40,58	49,09	0,00	53,29	0,00	49,40	31,95
30 Tramas y SNR 10 dB	59,50	62,46	54,48	59,89	59,03	63,07	59,48	55,91	59,23
30 Tramas y sin ruido	58,83	58,83	58,83	58,83	58,83	58,83	58,83	58,83	58,83

TEST	Eumodicogryllus Burgalensis. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	2,39	3,18	0,00	39,20	0,00	22,69	41,38	7,05	14,48
20 Tramas y SNR 10 dB	69,90	72,18	29,44	59,95	69,07	68,11	65,91	66,49	62,63
20 Tramas y sin ruido	70,65	70,65	70,65	70,65	70,65	70,65	70,65	70,65	70,65
30 Tramas y SNR 0 dB	21,07	17,19	28,08	36,60	0,00	40,79	0,00	36,90	22,58
30 Tramas y SNR 10 dB	75,23	71,71	70,20	75,43	74,75	50,57	74,34	71,64	70,48
30 Tramas y sin ruido	74,56	74,56	74,56	74,56	74,56	74,56	74,56	74,56	74,56

Avispa

TEST	Avispa. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	99,14	98,10	96,14	92,92	99,20	98,59	94,28	93,22	96,45
20 Tramas y SNR 10 dB	100,00	100,00	98,90	97,63	98,62	97,51	99,79	94,79	98,41
20 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00
30 Tramas y SNR 0 dB	98,61	97,87	92,54	97,13	97,19	97,17	94,49	93,75	96,09
30 Tramas y SNR 10 dB	97,18	97,28	98,56	96,52	94,76	99,19	98,81	96,87	97,40
30 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00

TEST	Abeja. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	81,91	83,85	85,81	89,03	82,39	82,87	76,76	88,72	83,92
20 Tramas y SNR 10 dB	81,95	81,95	83,04	84,31	80,56	80,35	85,52	87,15	83,11
20 Tramas y sin ruido	81,95	81,95	81,95	81,95	81,95	81,95	81,95	81,95	81,95
30 Tramas y SNR 0 dB	89,37	87,53	82,20	89,29	86,85	86,83	84,15	83,41	86,20
30 Tramas y SNR 10 dB	89,33	89,44	91,10	86,17	86,92	88,85	89,77	89,03	88,83
30 Tramas y sin ruido	89,66	89,66	89,66	89,66	89,66	89,66	89,66	89,66	89,66

TEST	Mosquito. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	85,59	83,65	81,69	78,47	85,11	84,63	90,74	78,77	83,58
20 Tramas y SNR 10 dB	85,55	85,55	84,45	83,18	86,93	87,14	81,97	80,34	84,39
20 Tramas y sin ruido	85,55	85,55	85,55	85,55	85,55	85,55	85,55	85,55	85,55
30 Tramas y SNR 0 dB	86,61	88,46	93,41	84,17	89,14	89,16	91,83	92,58	89,42
30 Tramas y SNR 10 dB	83,74	83,61	84,89	89,81	81,69	87,14	86,21	83,91	85,13
30 Tramas y sin ruido	86,33	86,33	86,33	86,33	86,33	86,33	86,33	86,33	86,33

Mosquito

TEST	Mosquito. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
20 Tramas y SNR 10 dB	96,67	96,08	100,00	97,37	96,99	95,49	97,61	95,45	96,96
20 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00
30 Tramas y SNR 0 dB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
30 Tramas y SNR 10 dB	98,88	99,28	99,47	98,71	98,37	98,03	99,44	99,31	98,94
30 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00

TEST	Abeja. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
20 Tramas y SNR 10 dB	70,83	66,45	67,50	70,13	65,21	64,78	69,88	72,05	68,35
20 Tramas y sin ruido	67,50	67,50	67,50	67,50	67,50	67,50	67,50	67,50	67,50
30 Tramas y SNR 0 dB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
30 Tramas y SNR 10 dB	74,86	75,75	75,56	77,27	74,35	77,95	76,55	75,30	75,95
30 Tramas y sin ruido	75,99	75,99	75,99	75,99	75,99	75,99	75,99	75,99	75,99

TEST	Avispa. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
20 Tramas y SNR 10 dB	88,88	84,50	85,55	88,18	83,26	82,84	87,94	90,10	86,41
20 Tramas y sin ruido	85,55	85,55	85,55	85,55	85,55	85,55	85,55	85,55	85,55
30 Tramas y SNR 0 dB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
30 Tramas y SNR 10 dB	85,20	86,09	85,90	87,62	84,70	88,29	86,89	85,64	86,29
30 Tramas y sin ruido	86,33	86,33	86,33	86,33	86,33	86,33	86,33	86,33	86,33

Abeja

TEST	Abeja. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	91,28	95,65	92,40	94,66	88,74	91,25	92,27	95,70	92,74
20 Tramas y SNR 10 dB	92,20	98,37	98,88	91,78	93,93	98,91	99,29	92,70	95,76
20 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00
30 Tramas y SNR 0 dB	96,27	96,62	95,68	95,86	91,83	91,84	97,48	95,17	95,09
30 Tramas y SNR 10 dB	98,93	97,55	98,51	97,66	97,55	99,32	98,93	98,83	98,41
30 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00

TEST	Avispa. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	90,67	86,29	89,54	87,28	91,18	90,70	89,68	81,96	88,41
20 Tramas y SNR 10 dB	89,74	83,58	83,07	90,17	88,02	89,23	82,66	89,25	86,96
20 Tramas y sin ruido	81,95	81,95	81,95	81,95	81,95	81,95	81,95	81,95	81,95
30 Tramas y SNR 0 dB	87,00	91,05	87,37	87,79	82,93	82,22	89,41	86,14	86,74
30 Tramas y SNR 10 dB	89,66	88,64	90,08	90,65	88,64	90,17	89,66	90,83	89,79
30 Tramas y sin ruido	89,66	89,66	89,66	89,66	89,66	89,66	89,66	89,66	89,66

TEST	Mosquito. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	76,22	71,84	75,09	72,83	78,76	76,25	75,23	67,51	74,22
20 Tramas y SNR 10 dB	75,29	69,13	68,62	75,72	73,57	76,40	68,21	74,80	72,72
20 Tramas y sin ruido	67,50	67,50	67,50	67,50	67,50	67,50	67,50	67,50	67,50
30 Tramas y SNR 0 dB	73,33	79,37	73,70	74,12	69,26	68,54	75,73	72,47	73,31
30 Tramas y SNR 10 dB	75,99	74,97	76,41	78,33	74,97	76,50	75,99	77,16	76,29
30 Tramas y sin ruido	75,99	75,99	75,99	75,99	75,99	75,99	75,99	75,99	75,99

Mosca

TEST	Mosca. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	98,97	97,84	98,75	97,44	99,14	98,15	95,33	96,32	97,74
20 Tramas y SNR 10 dB	99,85	99,54	99,13	100,00	99,72	99,72	98,61	97,99	99,32
20 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00
30 Tramas y SNR 0 dB	98,92	99,57	97,77	99,39	96,46	98,37	98,20	93,86	97,82
30 Tramas y SNR 10 dB	99,34	99,82	99,57	99,44	99,46	99,67	99,69	99,59	99,57
30 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00

TEST	Mosquito. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	74,33	75,46	74,56	75,86	74,16	75,15	77,16	76,98	75,46
20 Tramas y SNR 10 dB	73,22	73,76	74,17	73,30	73,58	73,58	74,15	75,31	73,88
20 Tramas y sin ruido	73,30	73,30	73,30	73,30	73,30	73,30	73,30	73,30	73,30
30 Tramas y SNR 0 dB	73,89	73,25	75,04	73,42	76,36	74,45	74,61	77,10	74,77
30 Tramas y SNR 10 dB	73,47	72,99	73,24	73,38	73,35	73,14	73,13	73,22	73,24
30 Tramas y sin ruido	72,82	72,82	72,82	72,82	72,82	72,82	72,82	72,82	72,82

Mordoganensis

TEST	Mordoganensis. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
20 Tramas y SNR 10 dB	91,01	91,23	63,24	85,81	93,01	95,89	65,41	85,57	83,90
20 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00
30 Tramas y SNR 0 dB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
30 Tramas y SNR 10 dB	97,19	78,14	90,39	90,77	89,48	91,15	68,74	71,26	84,64
30 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00

TEST	Acheta Domesticus. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
20 Tramas y SNR 10 dB	10,00	9,77	3,22	0,00	8,00	0,99	0,00	5,32	4,66
20 Tramas y sin ruido	1,01	1,01	1,01	1,01	1,01	1,01	1,01	1,01	1,01
30 Tramas y SNR 0 dB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
30 Tramas y SNR 10 dB	5,11	0,00	1,75	2,71	0,00	0,00	0,00	0,00	1,20
30 Tramas y sin ruido	7,20	7,20	7,20	7,20	7,20	7,20	7,20	7,20	7,20

Gryllotalpa (Grillo topo)

TEST	Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	98,84	98,84	98,84	98,84	98,84	98,84	98,84	98,84	98,84
20 Tramas y SNR 10 dB	99,21	98,36	98,76	98,41	98,45	98,78	98,83	99,60	98,80
20 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00
30 Tramas y SNR 0 dB	98,12	98,12	98,12	98,12	98,12	98,12	98,12	98,12	98,12
30 Tramas y SNR 10 dB	99,47	99,03	99,01	99,11	98,90	99,39	99,63	94,63	98,65
30 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00

Acherontia Atropos (Cigarra)

TEST	Acherontia Atropos. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	0,00	0,00	78,77	0,00	74,70	NR	NR	80,11	29,20
20 Tramas y SNR 10 dB	81,49	86,28	82,09	92,89	87,41	85,44	89,21	84,98	86,22
20 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00
30 Tramas y SNR 0 dB	0,00	0,00	0,00	0,00	0,00	0,00	78,18	0,00	9,77
30 Tramas y SNR 10 dB	89,08	86,65	80,26	79,81	85,23	84,21	88,78	81,82	84,48
30 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00

TEST	Cicada Cretensis. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	3,28	25,08	76,47	23,17	72,40	NR	NR	84,22	35,58
20 Tramas y SNR 10 dB	78,71	84,25	79,80	77,65	81,24	78,74	81,32	77,67	79,92
20 Tramas y sin ruido	70,53	70,53	70,53	70,53	70,53	70,53	70,53	70,53	70,53
30 Tramas y SNR 0 dB	1,49	0,00	0,00	0,00	32,38	18,87	78,88	0,00	16,45
30 Tramas y SNR 10 dB	75,13	76,41	75,83	75,38	80,80	83,32	76,14	77,39	77,55
30 Tramas y sin ruido	71,09	71,09	71,09	71,09	71,09	71,09	71,09	71,09	71,09

TEST	Cicada Barbara. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	0,00	0,00	74,86	0,00	77,19	NR	NR	72,33	28,05
20 Tramas y SNR 10 dB	84,04	78,50	80,12	85,10	81,51	84,01	81,42	80,85	81,94
20 Tramas y sin ruido	82,81	82,81	82,81	82,81	82,81	82,81	82,81	82,81	82,81
30 Tramas y SNR 0 dB	0,00	0,00	0,00	0,00	0,00	0,00	66,47	0,00	8,31
30 Tramas y SNR 10 dB	84,26	82,98	81,51	81,28	78,58	72,50	83,24	71,73	79,51
30 Tramas y sin ruido	82,57	82,57	82,57	82,57	82,57	82,57	82,57	82,57	82,57

Cicada Barbara (Cigarra)

TEST	Cicada Barbara. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	70,64	65,79	88,59	69,58	71,50	0,00	70,63	69,62	63,29
20 Tramas y SNR 10 dB	90,30	88,03	95,74	93,06	92,41	97,78	96,22	94,24	93,47
20 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00
30 Tramas y SNR 0 dB	76,66	74,35	79,95	72,74	0,00	64,30	69,38	75,42	64,10
30 Tramas y SNR 10 dB	89,61	93,15	93,58	95,11	91,87	90,86	89,14	91,81	91,89
30 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00

TEST	Cicada Cretensis. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	73,71	75,31	72,34	70,12	71,61	0,00	73,76	78,71	64,44
20 Tramas y SNR 10 dB	72,45	74,71	67,00	69,69	70,33	64,96	61,72	68,51	68,67
20 Tramas y sin ruido	62,75	62,75	62,75	62,75	62,75	62,75	62,75	62,75	62,75
30 Tramas y SNR 0 dB	67,64	68,16	74,77	74,19	0,00	70,39	70,01	70,56	61,97
30 Tramas y SNR 10 dB	69,78	66,23	65,81	64,27	67,52	68,53	70,25	67,58	67,50
30 Tramas y sin ruido	59,39	59,39	59,39	59,39	59,39	59,39	59,39	59,39	59,39

TEST	Acherontia Atropos Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	76,01	73,57	74,63	72,41	73,90	0,00	76,05	77,40	65,50
20 Tramas y SNR 10 dB	83,76	84,41	83,40	83,35	83,78	83,34	86,59	85,39	84,25
20 Tramas y sin ruido	82,81	82,81	82,81	82,81	82,81	82,81	82,81	82,81	82,81
30 Tramas y SNR 0 dB	72,08	72,60	79,21	78,63	0,00	74,82	74,45	75,00	65,85
30 Tramas y SNR 10 dB	81,76	81,82	80,88	81,34	81,93	79,81	76,98	79,65	80,52
30 Tramas y sin ruido	82,57	82,57	82,57	82,57	82,57	82,57	82,57	82,57	82,57

Cicada Cretensis (Cigarra)

TEST	Cicada Cretensis. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	79,08	75,36	90,72	82,63	72,84	85,35	85,74	81,49	81,65
20 Tramas y SNR 10 dB	94,68	83,18	96,99	95,83	95,25	90,70	96,59	98,37	93,95
20 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00
30 Tramas y SNR 0 dB	73,13	78,82	85,18	73,68	78,67	80,41	81,26	74,66	78,23
30 Tramas y SNR 10 dB	84,77	83,49	89,78	92,42	91,93	95,87	85,13	88,80	89,03
30 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00

TEST	Cicada Barbara. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	61,21	55,71	69,62	65,35	54,16	68,61	61,78	58,18	61,83
20 Tramas y SNR 10 dB	62,38	56,74	65,76	66,91	61,88	56,95	66,16	63,32	62,51
20 Tramas y sin ruido	62,75	62,75	62,75	62,75	62,75	62,75	62,75	62,75	62,75
30 Tramas y SNR 0 dB	58,79	57,16	60,82	52,88	58,29	60,93	57,76	58,31	58,12
30 Tramas y SNR 10 dB	59,59	55,76	55,84	57,16	58,99	60,00	54,24	54,97	57,07
30 Tramas y sin ruido	59,39	59,39	59,39	59,39	59,39	59,39	59,39	59,39	59,39

TEST	Acherontia Atropos Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	68,99	63,50	77,40	73,14	61,94	76,39	69,57	65,96	69,61
20 Tramas y SNR 10 dB	70,17	64,53	73,55	74,70	69,66	64,73	73,95	71,11	70,30
20 Tramas y sin ruido	70,53	70,53	70,53	70,53	70,53	70,53	70,53	70,53	70,53
30 Tramas y SNR 0 dB	70,50	68,87	72,53	64,58	70,00	72,64	69,47	70,02	69,83
30 Tramas y SNR 10 dB	71,30	67,47	67,55	68,87	70,70	71,70	65,95	66,68	68,78
30 Tramas y sin ruido	71,09	71,09	71,09	71,09	71,09	71,09	71,09	71,09	71,09

Langosta

TEST	Langosta. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
20 Tramas y SNR 10 dB	94,03	97,80	88,82	96,71	97,39	86,86	88,97	95,45	93,25
20 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00
30 Tramas y SNR 0 dB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
30 Tramas y SNR 10 dB	94,03	94,65	92,63	94,22	78,52	87,32	87,33	86,74	89,43
30 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00

TEST	Saltamontes. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
20 Tramas y SNR 10 dB	87,47	81,36	83,91	84,79	83,18	82,24	82,41	86,05	83,93
20 Tramas y sin ruido	81,50	81,50	81,50	81,50	81,50	81,50	81,50	81,50	81,50
30 Tramas y SNR 0 dB	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
30 Tramas y SNR 10 dB	84,40	86,61	85,00	84,76	86,15	80,43	79,71	80,86	83,49
30 Tramas y sin ruido	81,25	81,25	81,25	81,25	81,25	81,25	81,25	81,25	81,25

Saltamontes

TEST	Saltamontes. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	93,65	96,26	82,65	83,02	83,32	0,00	68,87	70,05	72,23
20 Tramas y SNR 10 dB	98,89	98,29	99,15	99,56	98,66	98,68	98,66	99,24	98,89
20 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00
30 Tramas y SNR 0 dB	74,98	82,16	82,78	0,00	80,76	87,98	89,63	80,15	72,31
30 Tramas y SNR 10 dB	97,42	99,71	97,48	97,15	98,76	97,24	97,58	98,82	98,02
30 Tramas y sin ruido	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00

TEST	Langosta. Porcentajes de similitud (%)								MEDIA
20 Tramas y SNR 0 dB	78,12	77,75	69,46	73,74	72,95	0,00	63,34	64,83	62,52
20 Tramas y SNR 10 dB	81,83	79,78	82,35	81,94	81,90	81,72	80,16	81,16	81,35
20 Tramas y sin ruido	81,50	81,50	81,50	81,50	81,50	81,50	81,50	81,50	81,50
30 Tramas y SNR 0 dB	67,07	71,23	69,35	0,00	69,93	74,44	73,28	68,90	61,77
30 Tramas y SNR 10 dB	83,84	81,54	83,78	82,53	81,35	82,04	83,67	82,43	82,65
30 Tramas y sin ruido	81,25	81,25	81,25	81,25	81,25	81,25	81,25	81,25	81,25

Anexo 4: Tests de Evaluación Sistema Final

Insecto	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	Test 10	Test 11	Test 12	Test 13	Test 14	Test 15	Test 16	Test 17	Test 18	Test 19	Test 20	Aciertos	Err	Media	Max	Min
<i>acherontia atropos</i>	92,13	95,70	92,94	92,44	93,47	89,44	92,06	88,50	93,86	89,03	92,89	86,87	94,45	87,3	89,04	89,34	91,73	93,44	92,31	86,12	18	2	91,14	95,70	86,12
<i>acheia domesticus</i>	89,27	95,89	97,6	93,46	88,64	92,02	95,01	95,52	97,38	95,04	91,23	93,76	95,98	94,82	96,91	95,95	96,08	92,63	96,53	93,36	20	0	94,35	97,6	88,64
<i>acrididae (langosta)</i>	94,76	93,97	93,25	96,11	94,63	95,12	92,36	95,42	94,79	90,68	88,95	95,16	79,87	92,99	95,78	94,07	94,27	95,01	95,03	94,5	20	0	93,34	96,11	79,87
<i>anthophila (abeja)</i>	87,83	88,49	84,39	85,57	87,11	87,7	60,53	80,4	53,12	85,67	85,87	86,78	84,18	72,2	89,61	83,84	84,01	83,4	73,87	84,5	16	4	81,45	89,61	53,12
<i>mordoganensis</i>	60,21	39,27	28,34	61,25	56,24	56,19	75,38	53,16	60,76	23,49	67,91	60,64	29,27	29,37	83,28	56,61	71,71	52,78	58,3	74,65	20	0	54,94	83,28	23,49
<i>cicada barbara</i>	98,73	93,03	97,2	96,11	92,91	96,09	90,03	94,2	91,38	97,66	97,43	91,07	97,51	98,81	92,58	98,37	90,84	89,3	93,81	98,16	20	0	94,76	98,81	89,3
<i>cicada cretensis</i>	94,68	87,52	97,14	95,34	91,5	95,66	94,04	94,38	93,59	87	94,6	90,57	98,04	92,16	94,83	96,28	94,61	91,13	89,45	95,26	20	0	93,39	98,04	87
<i>culicidae (mosquito)</i>	98,65	96,71	89,41	98,29	97,15	96,57	96,9	96,72	95,91	90,72	96,08	98,53	97,86	98,54	97,42	98,81	98,84	97,29	98,36	92,35	20	0	96,56	98,84	89,41
<i>eumodicoryllus</i>	90,83	92,2	83,22	91,01	91,34	97,35	96,41	97,46	90,39	84,03	91,21	91,47	91,56	98,1	89,43	97,82	96,7	91,35	95,2	76,52	18	2	91,68	98,1	76,52
<i>grylotalpa</i>	97,46	97,83	97,75	99,36	98,11	97,59	97,31	98,02	99,43	97,95	97,35	98,65	97,76	98,42	98,19	98,23	98,33	98,88	98,75	98,94	20	0	98,22	99,43	97,31
<i>gryllus campestris</i>	82,09	76,92	84,09	82,13	67,61	78,43	73,16	83,22	85,39	83,13	84,77	74,44	82,9	89,31	87,04	79,94	80,83	69,95	54,94	77,36	15	5	78,88	89,31	54,94
<i>muscidae (mosca)</i>	88,16	85,83	95,93	85,61	82,97	84,93	84,55	83,51	94,39	87,2	91,22	83,53	82,95	83,86	86,33			eum.(82, eumo.(86,72)			20	0	86,73	95,93	82,95
<i>vespula germanica</i>	99,02	94,95	93,91	93,8	94,37	54,16	53,33	95,25	96,18	97,58	92,26	98,5	93,75	97,74	54,65	80,86	93,84	73,37	54,54	54,37	19	1	83,28	99,02	53,33
<i>saltamontes</i>	78,01	73,44	88,85	86,24	86,53	70,17	87,13	63,79	88,11	87,5	87,95	88,83	85,34	71,53	88,67	69,62	84,41	71,76	88,92	87,48	17	3	81,71	88,92	63,79
						lang. 72,34					lang. 89,50														