

|  |                                      |         |            |
|--|--------------------------------------|---------|------------|
| <br>Institución Universitaria | INFORME FINAL DE<br>TRABAJO DE GRADO | Código  | FDE 089    |
|  |                                      | Versión | 03         |
|  |                                      | Fecha   | 2015-01-27 |

# TRADUCCIÓN Y DESARROLLO DEL SOFTWARE

## *ECOSONOS*

Manuel Alejandro Chalarca Botero

Ingeniería Mecatrónica

Director del trabajo de grado

Leonardo Duque Muñoz

**INSTITUTO TECNOLÓGICO METROPOLITANO**

Fecha

**20 de noviembre del 2023**

|   |                                      |         |            |
|---|--------------------------------------|---------|------------|
|  | INFORME FINAL DE<br>TRABAJO DE GRADO | Código  | FDE 089    |
|   |                                      | Versión | 03         |
|   |                                      | Fecha   | 2015-01-22 |

## RESUMEN

---

La ecoacústica es una disciplina que estudia la relación entre los seres vivos y el medio ambiente en el que habitan a través del sonido. Para estudiar el sonido, por lo general, se ubica un gran número de grabadoras en un ecosistema de interés con el fin de registrar el sonido por un tiempo definido. Esto genera una gran cantidad de información que debe ser almacenada y posteriormente procesada para reconocer información útil como el registro de sonotipos de animales, la estimación de lluvia en el ecosistema, entre otros. Es así como uno de los retos que plantea el estudio de la información acústica es el procesamiento de la información, puesto que los datos registrados pueden llegar a ocupar algunas teras en disco.

Atendiendo a lo anterior, en la fase de ejecución de este proyecto de investigación se procedió a la implementación en Python de *Ecosonos*, una nueva versión del software de reconocimiento de especies "AUREAS" previamente implementado en Matlab desde el año 2011 en la Universidad de Antioquia.

Mediante el proyecto de Minciencias RC 731-2020 "Programa Conservación Biológica usando Inteligencia Artificial" se planteó agregar funcionalidades a dicho software que permitiera compartirlo con la comunidad de biólogos y por eso surge la necesidad de implementación en un software de uso libre.

Es así como surge el módulo "Sonotipos" en el software *Ecosonos*, el cual permite la identificación de sonotipos mediante el uso de algoritmos de aprendizaje de máquina no supervisados. En concreto, en este se realiza el procesamiento de registros de audio a partir de algoritmos de agrupamiento (clustering), los cuales identifican los cantos de las especies o eventos de interés. Esta información es marcada en los metadatos de los audios, y sirve a los expertos quienes, mediante la información generada, pueden identificar si los eventos de interés se asocian a una especie en específico, se puede contar el número de cantos de dicha especie, si hay registro de lluvia, entre otros, lo cual permite estimar la salud de dicho ecosistema.

El siguiente módulo programado fue el módulo de reconocimiento, el cual, una vez identificado por parte de los expertos la información de los clústers, se prosigue a reconocer en audios no vistos la misma información, esto permite verificar si en otras zonas del ecosistema o en otros audios no analizados se encuentra la misma información que en los audios de entrenamiento.

|   |                                      |         |            |
|---|--------------------------------------|---------|------------|
|  | INFORME FINAL DE<br>TRABAJO DE GRADO | Código  | FDE 089    |
|   |                                      | Versión | 03         |
|   |                                      | Fecha   | 2015-01-22 |

Mediante la programación de este software se logró integrar de una mejor manera este módulo al software *Ecosonos*, en una distribución con software gratuito y tener las mismas funcionalidades del software original.

*Palabras clave:* Bioacústica, “Clustering”, Matlab, Python

## RECONOCIMIENTOS

---

Un agradecimiento a mi madre, Teresita, y a mi padre, Nicolás, por haberme brindado la oportunidad de iniciar y terminar esta carrera, así como a mi hermano, mis tías y demás familiares por su valioso apoyo y por estar al pendiente de mi avance en ésta, semestre a semestre.

Al grupo de investigación “Máquinas Inteligentes y Reconocimiento de Patrones” - MIRP, por haberme abierto las puertas a una oportunidad de aprendizaje y de formación académica y profesional.

A mi compañera, Diana, por haber estado conmigo, hombro a hombro, desarrollando este proyecto y al profesor Leonardo, por haberme acompañado durante los últimos meses de formación, dándome acompañamiento y siendo una acertada guía.

Por último, a todos aquellos colegas que conocí a lo largo de la carrera y ahora puedo llamar amigos.

A todos los mencionados y a quienes haya podido olvidar, solo me resta decirles: Este logro no es solo mío, es nuestro, y haber podido coincidir y contar con ustedes es algo que siempre voy a apreciar.

|   |  |         |            |
|---|--|---------|------------|
|  | <b>INFORME FINAL DE<br/>TRABAJO DE GRADO</b> | Código  | FDE 089    |
|   |  | Versión | 03         |
|   |  | Fecha   | 2015-01-22 |

# ACRÓNIMOS

---

**MIRP:** Máquinas Inteligentes y Reconocimiento de Patrones

**STFT:** Transformada Rápida de Fourier.

# TABLA DE CONTENIDO

|  |               |
|--|---------------|
| <b>1. INTRODUCCIÓN</b> .....                                   | <b>pag 4</b>  |
| <b>2. MARCO TEÓRICO</b> .....                                  | <b>pag 7</b>  |
| <b>3. METODOLOGÍA</b> .....                                    | <b>pag 13</b> |
| <b>4. RESULTADOS Y DISCUSIÓN</b> .....                         | <b>pag 20</b> |
| <b>5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO</b> ..... | <b>pag 37</b> |
| <b>6. REFERENCIAS</b> .....                                    | <b>pag 40</b> |
| <b>7. APÉNDICE</b> .....                                       | <b>pag 41</b> |

## 1. INTRODUCCIÓN

---

### Generalidades

La ecoacústica es una ciencia que se ocupa del estudio de los sonidos del ambiente, los animales y las personas, así como del modo en que éstos se relacionan entre ellos. Por su

|   |  |         |            |
|---|--|---------|------------|
|  | <b>INFORME FINAL DE<br/>TRABAJO DE GRADO</b> | Código  | FDE 089    |
|   |  | Versión | 03         |
|   |  | Fecha   | 2015-01-22 |

parte, la bioacústica surge del estudio de los sonidos que emiten los animales y a éstos se les denomina biofonías.

Las biofonías resultan de gran valor para los biólogos, ya que, con base en ellas, pueden estudiar el comportamiento de las poblaciones de animales de forma poco invasiva. No obstante, el análisis de los audios, por lo general, resulta tedioso, repetitivo y demorado, pues, durante la fase de recolección de datos, se genera una gran cantidad de información que, de no ser analizada de manera automática, requeriría el trabajo de expertos durante cientos de horas de reproducción de audios. Por esto, surge este proyecto con la necesidad de analizar miles de audios captados por las grabadoras de diferentes universidades de Colombia con el fin de brindar soporte a los biólogos encargados de verificar los comportamientos de los ecosistemas monitoreados para así facilitar la creación de programas de conservación biológica, políticas públicas de conservación, entre otros.

El software producto de este proyecto comenzó a ser desarrollado por la universidad de Antioquia y, en años posteriores, se unió a él el Instituto Tecnológico Metropolitano, con la participación del grupo de Máquinas Inteligentes y Reconocimiento de Patrones (MIRP), en el lenguaje de Matlab. Como consecuencia de esta alianza surgió la necesidad de crear nuevos módulos, esta vez en un software gratuito, por lo que se decidió implementar, y se entregó una copia del código fuente del software con el fin de identificar el camino óptimo para realizar la traducción y complementar funcionalidades.

### **Objetivos:**

General:

|   |  |         |                |
|---|--|---------|----------------|
|  | <b>INFORME FINAL DE<br/>TRABAJO DE GRADO</b> | Código  | FDE 089        |
|   |  | Versión | 03             |
|   |  | Fecha   | 2015-01-2<br>2 |

- Desarrollar una nueva versión del software de reconocimiento de especies *Ecosonos*, en Python, para que más investigadores puedan acceder a la licencia del software libre.

Específicos:

1. Investigar el algoritmo de reconocimiento de especies del software “Ecosonos”, programado en Matlab, para proponer una metodología que permita realizar su traducción a Python.
2. Adaptar los resultados de reconocimiento de especies para proponer diagramas de patrones de actividad de aquellas.
3. Validar los resultados obtenidos en cada lenguaje de programación para asegurar la fiabilidad del nuevo producto, respecto al software inicial.

### **Organización de la tesis**

En el marco teórico se encuentra el contexto histórico y técnico de los métodos, conceptos y conocimientos específicos bajo los cuales se construyó este proyecto, con el fin de facilitar al lector el entendimiento de los procesos que se describirán a lo largo de este informe.

En la metodología se describe la secuencia lógica y el paso a paso mediante el cual se dedujo, organizó y procedió al desarrollo del proyecto de grado, así como los problemas encontrados y las soluciones aplicadas.

|   |  |         |                |
|---|--|---------|----------------|
|  | <b>INFORME FINAL DE<br/>TRABAJO DE GRADO</b> | Código  | FDE 089        |
|   |  | Versión | 03             |
|   |  | Fecha   | 2015-01-2<br>2 |

En los resultados se presentan la descripción de los códigos más relevantes, la información obtenida de éstos y el cómo abrieron la posibilidad para un desarrollo posterior en el grupo de MIRP.

## 2. MARCO TEÓRICO

---

Las señales acústicas son registros de audio disponibles en el entorno, hayan sido emitidos o no por seres vivos. En diversos grupos de animales, tras un proceso evolutivo, éstas han llegado a cumplir múltiples funciones como defender el territorio, alertar sobre la presencia de depredadores, mantener la cohesión de los grupos mixtos, atraer la atención de una posible pareja e, incluso, localizar a los individuos y detectar a las presas (Bradbury & Vehrencamp, 2011). Por su parte, la bioacústica es una disciplina que estudia la emisión de sonido por parte de los organismos, su difusión a través de un medio elástico y, finalmente, su recepción y los eventuales cambios en el comportamiento del receptor, dada la información contenida. (Martínez Medina et al., 2021). De forma complementaria, se encuentra la ecoacústica, una disciplina que utiliza los sonidos ambientales como herramienta para investigaciones en ecología y conservación y se compone de tres elementos existentes en los paisajes sonoros: biofonías (sonidos biológicos de animales), geofonías (sonidos ambientales no biológicos) y antropofonías/tecnofonías (sonidos generados por la actividad humana). Siendo de interés para el presente trabajo las biofonías.

Las enunciadas disciplinas son de gran utilidad en el ámbito del estudio y la conservación de la ecología en las zonas estudiadas, toda vez que, contribuyen a:

|   |                                      |         |                |
|---|--------------------------------------|---------|----------------|
|  | INFORME FINAL DE<br>TRABAJO DE GRADO | Código  | FDE 089        |
|   |                                      | Versión | 03             |
|   |                                      | Fecha   | 2015-01-2<br>2 |

**La detección de individuos y especies:** Permite la identificación de especies determinadas, partiendo del análisis de las biofonías en una zona específica.

**Localización de animales:** Las biofonías pueden usarse para determinar la ubicación de diferentes especies, lo que es ideal para el estudio de migraciones o patrones de movimiento.

**Monitoreo de población:** Permite establecer qué tal se encuentra la salud y las dinámicas de las diferentes especies de la zona.

Los registros de audio analizados en este proyecto fueron obtenidos a través de grabadoras “AudioMoth” ubicadas estratégicamente en ecosistemas de interés, con el fin de captar registros de audio en el espectro audible y ultrasónico. Así también, éstas pueden ser programadas para no generar información de manera continua, si no de acuerdo a la necesidad del estudio en específico. Por ejemplo, se pueden programar para grabar un minuto, cada 15 minutos, durante 7 días, lo cual generaría 11.2 horas de grabación en una sola grabadora.

Hasta ahora, los registros se han analizado por biólogos de manera manual, circunstancia que lleva a la fatiga del experto, puesto que solo una grabadora puede generar horas y horas de grabación. Por tanto, ha surgido la necesidad de que este proceso se realice mediante modelos automáticos. Como consecuencia de lo anterior, *Ecosonos* sugiere una caracterización de los audios en el espacio tiempo-frecuencia, mediante la transformada de Fourier en tiempo corto, y luego se realiza una segmentación de los audios mediante procesamiento de imágenes. Por su parte, estos segmentos alimentan a un algoritmo automático de “clustering”, para, finalmente, mediante la ayuda de un experto, analizar los clústers obtenidos y asociarlos a eventos de interés.

|   |                                      |         |            |
|---|--------------------------------------|---------|------------|
|  | INFORME FINAL DE<br>TRABAJO DE GRADO | Código  | FDE 089    |
|   |                                      | Versión | 03         |
|   |                                      | Fecha   | 2015-01-22 |

La transformada de Fourier de tiempo reducido (*Short-Time Fourier Transform, STFT*), al igual que la transformada de Fourier, permite transformar señales al dominio de tiempo y frecuencia, con base en la premisa de que cada señal puede interpretarse como un conjunto de señales sinusoidales. No obstante, la STFT se caracteriza porque tiene una resolución fija, definida por una ventana en la que el ancho es determinado por un intervalo de tiempo y el alto por uno de frecuencias, formando un espectrograma que posibilita tener una gráfica tridimensional (tiempo, frecuencia, intensidad) del audio analizado. Partiendo de esto se realiza un estudio, mediante visión artificial, para luego investigar las facilidades que brindan el aprendizaje no- supervisado (agrupación de elementos etiquetados) y los diferentes métodos de “clustering” (agrupación de elementos no etiquetados), como medios para analizar la información recolectada, antes de enviarla a los biólogos, para que estimen la utilidad de las detecciones.

La mayoría de los algoritmos se basan en la siguiente ecuación, o sus variantes, donde  $w(t)$  es la función ventana,  $x(t)$  es la señal a ser transformada y  $X(\tau, \omega)$  es la Transformada de Fourier de  $x(t)w(t - \tau)$

$$\{x(t)\} = X(\tau, \omega) = \int_{-\infty}^{\infty} x(t)w(t - \tau)e^{-j\omega t} dt$$

Fórmula 1. Ecuación de la transformada de Fourier de tiempo corto.

En el siguiente ejemplo se puede apreciar como las señales de audio pueden ser convertidas en imágenes, lo cual facilita su análisis. A modo de demostración, a continuación se presenta la la gráfica de la lectura del audio JAGUAS253\_20130117\_075446, el cual fue proporcionado como material de trabajo, junto con otros 29 audios, por el grupo MIRP y la Universidad de Antioquia.

|   |  |         |            |
|---|--|---------|------------|
|  | <b>INFORME FINAL DE<br/>TRABAJO DE GRADO</b> | Código  | FDE 089    |
|   |  | Versión | 03         |
|   |  | Fecha   | 2015-01-22 |

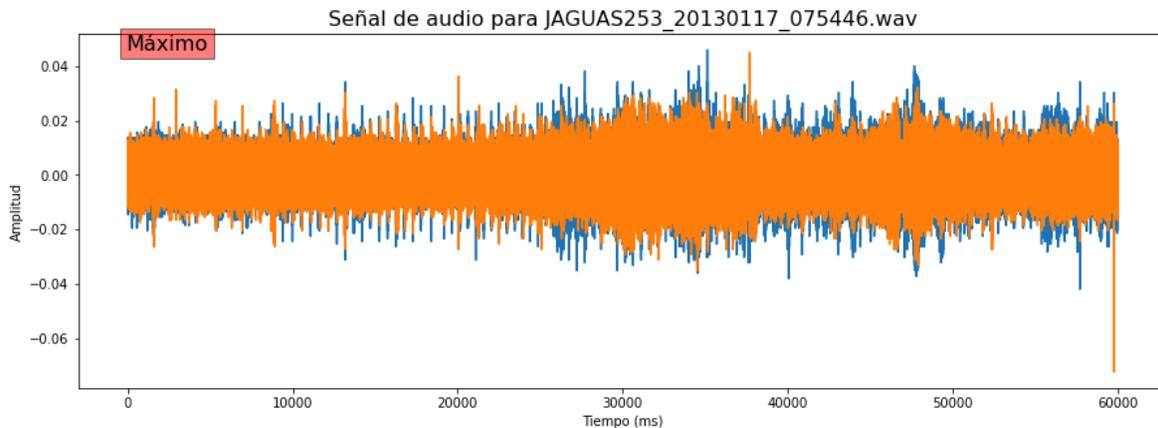


Fig. 1. Señal del audio de JAGUAS 253\_20130117\_075446.

La fig.1 presenta la señal analizada, es una señal estéreo, por lo cual consta de dos canales, esta señal consta de un minuto de grabación con una frecuencia de muestreo de 43.000Hz. Posteriormente, se promedian estas señales y se realiza la transformación STFT para obtener un resultado como el que se muestra a continuación en la Fig.2.

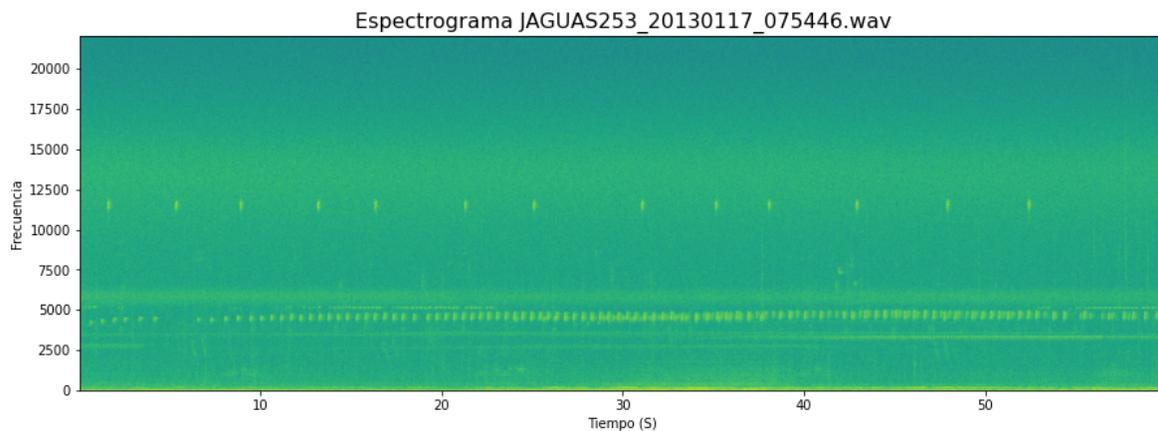


Fig. 2. Espectrograma para JAGUAS 253\_20130117\_075446.wav.

En la Fig.2 observamos un espectrograma de la señal de audio presentada e la fig 1. Se observa en el eje x la duración de la señal en seg y en el eje y la Frecuencia en Hz. Cada pixel de la imagen representa la intensidad (o energía) en un par coordenado (tiempo,

|   |                                      |         |                |
|---|--------------------------------------|---------|----------------|
|  | INFORME FINAL DE<br>TRABAJO DE GRADO | Código  | FDE 089        |
|   |                                      | Versión | 03             |
|   |                                      | Fecha   | 2015-01-2<br>2 |

frecuencia). Del mismo modo, este mapa permite observar, con mayor facilidad, los lugares en los que ocurre la actividad en el audio, siendo éstos los que figuran con más intensidad en la imagen, pues presentan mayor energía. Con esta información, según su frecuencia, forma y duración se puede identificar eventos de interés, como cantos de animales de diferentes especies, sonidos de antropofonía, lluvia, entre otros.

Estos datos por sí solos no son utilizables, por lo que es necesario usar técnicas de visión artificial en aras de poder extraer las características relevantes. A propósito de lo anterior, es importante anotar que la visión artificial es un campo de la inteligencia artificial que le permite a ordenadores y sistemas extraer información significativa partiendo de imágenes digitales, videos y otras entradas visuales, así como tomar medidas o realizar recomendaciones en función de aquella. En este caso se procesan los espectrogramas mediante algoritmos de tratamiento de imagen con procesos de binarizado en los que cualquier valor de un pixel de la imagen, bien sea que se encuentre dentro o fuera de un límite establecido, será convertido en blanco o negro con el fin de aplicar operaciones morfológicas que se basan en formas geométricas para cambiar los parámetros de las imágenes que, en este caso, se usan para eliminar interferencias en la imagen.

Una vez obtenida la imagen binarizada, se puede realizar el procedimiento de clustering, el cual permite obtener el agrupamiento de las características. Se trata de un procedimiento que consiste en reunir datos de grupos inconexos, de modo que los datos del mismo grupo son similares, pero, los de otros grupos, difieren, en mayor o menor medida. Esta capacidad de generar grupos de datos o “clústers” es lo que permite ordenar la información de manera automática, facilitando labores posteriores como el etiquetado.

Uno de los algoritmos de “clustering” más conocidos es el **K-means**, un método en el que se pre establece un número de clústers para cierta cantidad de observaciones y,

|   |                                      |         |                |
|---|--------------------------------------|---------|----------------|
|  | INFORME FINAL DE<br>TRABAJO DE GRADO | Código  | FDE 089        |
|   |                                      | Versión | 03             |
|   |                                      | Fecha   | 2015-01-2<br>2 |

dependiendo de la cercanía de los datos a estos clústers, se terminan agrupando. Posteriormente, el centro se recalcula con el promedio de los elementos que hayan entrado al clúster y, a partir de allí, se miden nuevamente las distancias a los elementos para asignarles un clúster y esto continúa hasta que no hay variaciones o hasta que se agreguen nuevos datos. Existe una variante del **K-means**, se trata del **Fuzzy K-Means**, el cual tiene un principio similar, pero agrega un indicador de pertenencia para cada elemento a los clústers, permitiendo clasificarlos de mejor manera.

Este agrupamiento, como se mencionó anteriormente, puede realizarse usando las frecuencias y duraciones de las biofonías detectadas en los audios para, de esta manera, agrupar los cantos similares (o eventos de interés que presenten características similares). Una vez hecho lo anterior, se generan metadatos por cada segmento de audio, en el cual queda registrado qué grupos contiene cada audio, la duración de los eventos de interés y las frecuencias en las que está cada característica agrupada. Esta información es analizada por los usuarios, quienes se encargan de filtrar las señales relevantes y generan etiquetas que, *a posteriori*, pueden ser usadas para una conseguir una clasificación más específica en otros audios. Respecto de este procedimiento, vale la pena indicar que todas las detecciones son guardadas en una base de datos para así guardar registro de los audios analizados y poder acceder a ellos sin necesidad de realizar los análisis auditivos nuevamente.

El software fue otorgado en el lenguaje de Matlab, con el fin de entender la estructura y las potenciales diferencias del código fuente, respecto a la futura contraparte en Python; entonces, en primer lugar, se comenzó a estudiar el programa para extraer la estructura del software, sus utilidades, funciones básicas y complementos y, en segundo lugar, se estableció un diagrama de flujo que sirvió como base para realizar la traducción de los elementos más relevantes, a saber: los algoritmos de detección y clasificación y los de



|  |                                      |         |           |
|--|--------------------------------------|---------|-----------|
| <br>Institución Universitaria | INFORME FINAL DE<br>TRABAJO DE GRADO | Código  | FDE 089   |
|  |                                      | Versión | 03        |
|  |                                      | Fecha   | 2015-01-2 |
|  |                                      |         | 2         |

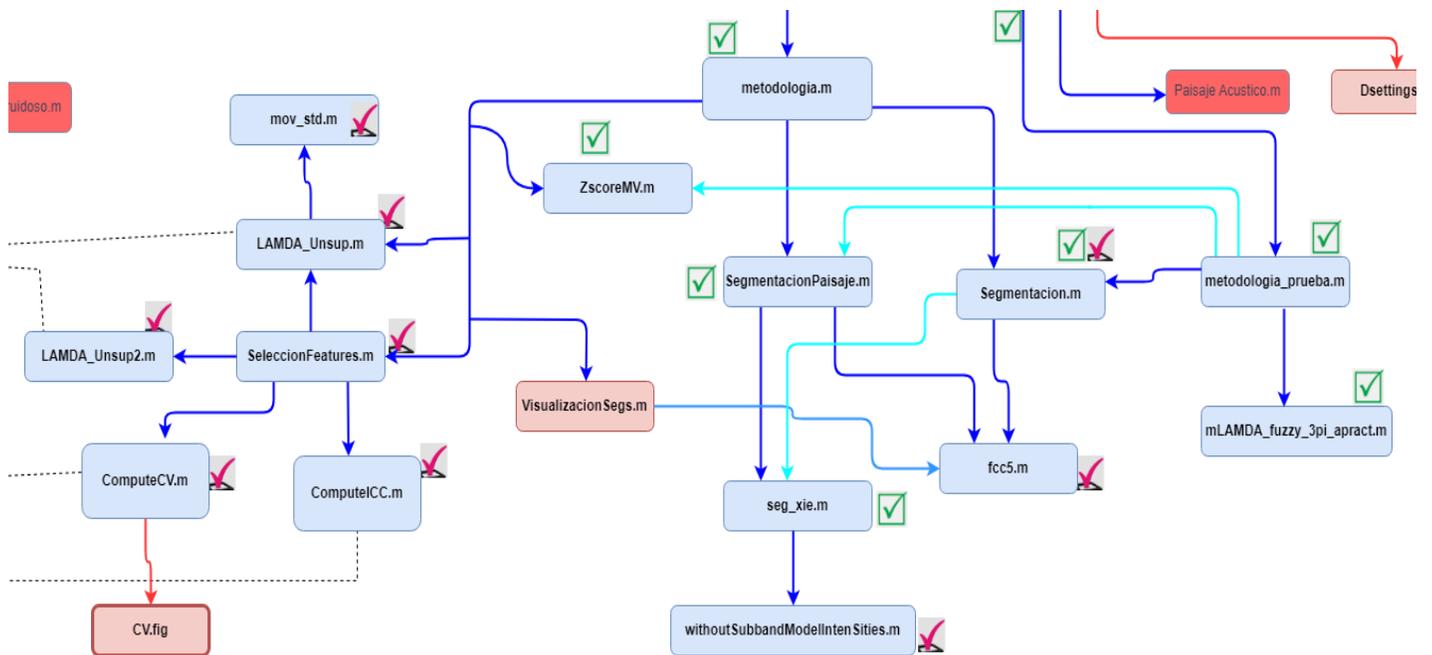


Fig. 3.1. Enfoque en las funciones de extraccion y analisis de informacion.

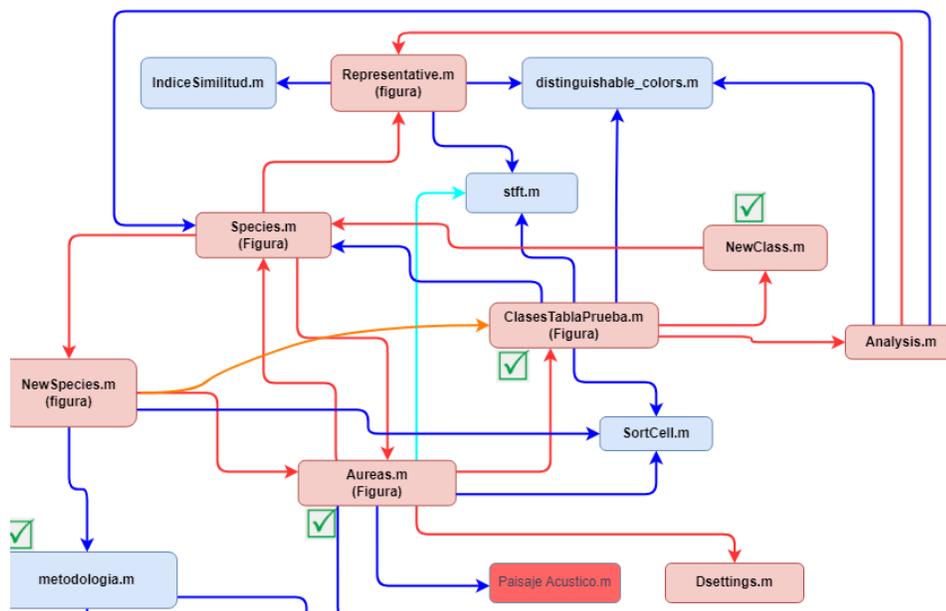


Fig. 3.2. Enfoque en las funciones de interfaz de matlab.

|   |                                      |         |                |
|---|--------------------------------------|---------|----------------|
|  | INFORME FINAL DE<br>TRABAJO DE GRADO | Código  | FDE 089        |
|   |                                      | Versión | 03             |
|   |                                      | Fecha   | 2015-01-2<br>2 |

Así pues, se identificaron las figuras de “NewSpecies” y “Aureas” como las principales encargadas de iniciar el procesamiento de la información, destacando la función de **metodología**, pues une los demás elementos de análisis del código. Así también, se pudo determinar que la información se procesa en diferentes etapas, utilizando los scripts previamente programados y siguiendo el proceso que se relaciona en lo sucesivo.

A través de las interfaces “NewSpecies” o “Aureas” se envía la dirección de una carpeta en la que se encuentran los audios a analizar, cada uno de éstos es convertido a un espectrograma, usando la STFT (Transformada Rápida de Fourier), y, acto seguido, se obtiene una matriz de tres dimensiones que indica el tiempo, la frecuencia y la intensidad; dicha matriz es luego procesada para eliminar ruido y convertirlo en decibelios. Más adelante, con el uso de visión artificial, se resaltan las zonas de alta actividad mediante un binarizado y, mediante operaciones morfológicas, se eliminan los ruidos. Posteriormente, se extraen coordenadas en un rango de tiempo y frecuencia, generando una *bounding box* que encierra todas las secciones. Por último, se aplica un filtro de tamaño y forma para asegurar la calidad de los segmentos y este se proceso se repite para todos elementos detectados en cada audio analizado.

Una vez obtenida esta información, se analiza dando datos adicionales, como valores estadísticos y una etiqueta preliminar que permite agruparlos en clústers para luego organizar la información con más información relevante como nombre del archivo al que pertenece el clúster, fecha de extracción, frecuencias máximas y mínimas, rango de frecuencia y tiempo de cada segmento y clúster al cual fue asociado, luego estos clústers pueden ser seleccionados y fusionados para crear una base de datos con la cual se puede dar un nombre propio a los clústers, por ejemplo, el nombre de una especie de rana, esto nos permite usar una nueva clasificación usando Áureas la cual al momento de clasificar nos permitirá buscar la presencia de las especies que ya hayamos identificado.

|   |  |         |            |
|---|--|---------|------------|
|  | <b>INFORME FINAL DE TRABAJO DE GRADO</b> | Código  | FDE 089    |
|   |  | Versión | 03         |
|   |  | Fecha   | 2015-01-22 |

Lo descrito anteriormente es una explicación sobre el funcionamiento del software *Ecosonos* y, con base en esta información, se abre la puerta a la creación de gráficas y otros componentes de análisis estadístico, como lo muestra la fig.2

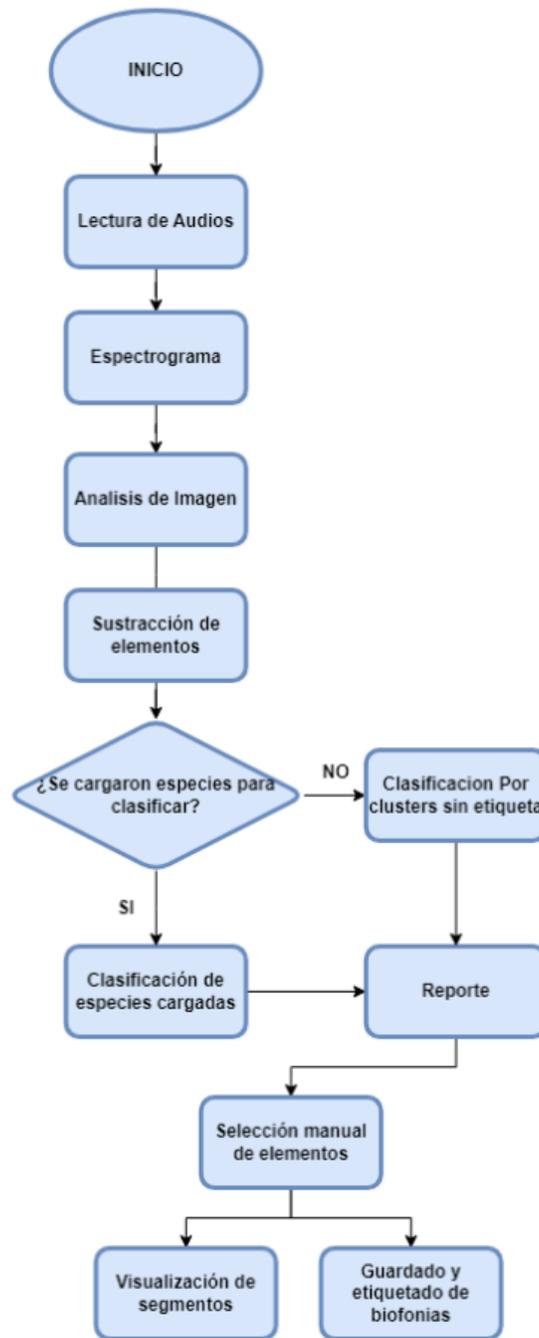


Fig. 4. Diagrama de flujo del software *Ecosonos* en Matlab

|   |  |         |            |
|---|--|---------|------------|
|  | <b>INFORME FINAL DE<br/>TRABAJO DE GRADO</b> | Código  | FDE 089    |
|   |  | Versión | 03         |
|   |  | Fecha   | 2015-01-22 |

En Matlab existen las toolbox, las cuales son herramientas descargables a través de la pestaña de Add-Ons, circunstancia que permite implementar muchas funciones para diversas tareas, en este caso el análisis de imágenes, sonidos y medidas estadísticas. Por su parte, en Python existe un equivalente llamado librerías, estas librerías son fáciles de instalar y crear; de hecho, existen librerías que desarrollan labores similares a muchos toolboxes de Matlab. Sin embargo, el problema que surge de esto es la estandarización de los métodos, ya que existen muchas librerías para Python para las mismas funciones, por ejemplo, diversas librerías permiten la lectura de audios y la generación de espectrogramas, pero es probable que lo hagan con pequeñas diferencias respecto a su equivalente en Matlab. Si bien, en principio, estas diferencias pueden parecer irrelevantes, pueden también causar perturbaciones que se van acumulando a lo largo del proceso, por lo cual fue necesario experimentar con diferentes librerías, en varios aspectos, para tratar de conseguir un resultado aceptable, en especial, en la parte de extracción de información.

El caso más importante y que permite ilustrar lo anterior es el relacionado con el modo de lectura de los audios: mientras en Matlab existe una función específica para esto (Audioread); en Python se encuentra una a librería de “Scipy”, con la extensión de “wavfile.read”, de la cual se hizo uso al comienzo para realizar las lecturas de audio. Si bien, al inicio, las pruebas daban resultados muy similares al los de su homólogo, con el transcurrir de los días, comenzaron a aparecer errores cada vez más notorios (audios que no se leían, espectrogramas con defectos, menor número de elementos detectados), entre otros. Por lo anterior, en una etapa posterior se decidió usar la librería “Soundfile”, una librería especializada en audio y con la cual mejoró notablemente la calidad de las lecturas.

|   |                                      |         |            |
|---|--------------------------------------|---------|------------|
|  | INFORME FINAL DE<br>TRABAJO DE GRADO | Código  | FDE 089    |
|   |                                      | Versión | 03         |
|   |                                      | Fecha   | 2015-01-22 |

Ahora bien, en la fase de realización de los espectrogramas ocurrió algo similar, ya que se empezó a usar la función “Matplotlib de spectrogram”. La cual, aunque generaba una imagen adecuada, era insuficiente para procesos posteriores, además de tomar mucho tiempo de ejecución. Por lo tanto se reemplazó por la librería “Scipy”, con la extensión “signal.spectrogram”, la cual permitió un mayor y mejor acercamiento a los resultados de Matlab.

Por su parte, en cuanto al procesamiento de imágenes, se utilizó la librería “Opencv (cv2)”, como contraparte del toolbox “Image Processing”. Aquella permitió realizar todos los procesos relevantes, como filtros adaptativos para binarizaciones, operaciones morfológicas, visualización de imágenes y otro tipo de filtros necesarios para la función más importante del software *Ecosonos*, esto es: la extracción de la información esencial de cada audio.

Adicionalmente, no solo se buscaron funciones homólogas de matlab en Python, también resulta importante anotar que, en esta etapa de la investigación, fue necesario crear algunas funciones adicionales, como “**findecentricity**”, la cual es una operación matemática, presente en las funciones de Matlab, que, para ese momento, no estaba disponible en las librerías de Python y, para continuar con la investigación, fue necesario tenerla con una funcionalidad similar.

Dadas algunas circunstancias, además de las ya señaladas, hay que decir que no todas las funciones llegaron a dar resultados idénticos a los obtenidos en Matlab.

Con el ánimo de ampliar esta idea, se puede hablar de dos casos relevantes: en primer lugar, el proceso ejecutado por “Gaussianblur”, por medio del cual se difumina un poco los segmentos antes de ser binarizados, pues, bien fuera que se realizará en Matlab o en

|   |  |         |            |
|---|--|---------|------------|
|  | <b>INFORME FINAL DE<br/>TRABAJO DE GRADO</b> | Código  | FDE 089    |
|   |  | Versión | 03         |
|   |  | Fecha   | 2015-01-22 |

Python, daba lugar a pequeñas diferencias que terminaban cambiando los tamaños de los segmentos. En segundo lugar y de forma similar, en Python existe la librería “Numpy”, en la cual existe una función que permite calcular la desviación estándar, sin embargo, aquella arroja en su resultado diferencias en cuanto a la misma función en Matlab, lo cual se debe al tamaño en memoria que ambos programas tienen para los números flotantes.

En cuanto a la selección de los segmentos de la tabla resultante del proceso, es posible afirmar que se trata de un paso necesario para poder guardar un fenotipo y luego darle una etiqueta adecuada. Sobre el particular, en Matlab se cuenta ya con una interfaz que permite navegar entre varias pantallas para realizar el guardado del segmento.

Teniendo en cuenta que en este producto de laboratorio el alcance era únicamente para la parte de backend, el elegir estos elementos resultaba una tarea compleja, más aún, teniendo en cuenta que el código para esta acción estaba dividido en varios segmentos, a lo largo de las tres figuras que lo componían; así las cosas, para simplificar esto, toda la operación de guardado de segmentos se unificó en el código presente en los diferentes botones de aquellas en una sola función, que toma y agrupa todos los elementos de un mismo clúster, permitiendo, a su vez, que el código al pasar a frontend sea sea posible agregar una modificación para seleccionar elementos independientes, imitando el funcionamiento en Matlab. Estos elementos se guardan, de forma conjunta, en una matriz que puede ser fácilmente almacenada en excel y se puede hacer uso de ella en instancias diferentes del sistema.

La última utilidad programada fue la encargada de la visualización de los elementos detectados en una imagen, remarcando la posición y el clúster al que pertenecen los segmentos encontrados en un determinado audio; siendo esta una herramienta muy útil, por cuanto facilita el encontrar patrones presentes en los espectrogramas y permite

|   |                                      |         |            |
|---|--------------------------------------|---------|------------|
|  | INFORME FINAL DE<br>TRABAJO DE GRADO | Código  | FDE 089    |
|   |                                      | Versión | 03         |
|   |                                      | Fecha   | 2015-01-22 |

identificar rápidamente las especies presentes en los audios, las más activas o sus patrones de movimiento a lo largo del día, a través de los audios recopilados en diferentes jornadas.

## 4. RESULTADOS Y DISCUSIÓN

---

En el marco de este proyecto de investigación se programó un total de dieciocho funciones y luego se juntaron en una librería llamada *Ecosonos*. De aquellas funciones se tiene que trece son traducciones directas de Matlab, con cambios menores de compatibilidad o ajustes internos. Éstas son:

- without\_subband\_mode\_intensities
- fcc5
- seg\_xie
- segmentación
- Zscore\_MV
- mov\_std
- lamda\_unsup
- compute\_icc
- compute\_cv
- seleccion\_features
- Metodología
- mlamda\_fuzzy\_3pi\_apract
- Metodologia\_Prueba

|   |  |         |            |
|---|--|---------|------------|
|  | <b>INFORME FINAL DE<br/>TRABAJO DE GRADO</b> | Código  | FDE 089    |
|   |  | Versión | 03         |
|   |  | Fecha   | 2015-01-22 |

Las otras cuatro funciones surgieron en respuesta a adaptaciones o agregados, creados para cumplir con requerimientos solicitados, o como modificación para facilitar la realización de tareas concretas:

- Findecentricity
- time\_and\_date
- smooth
- guardado\_cluster

Del total de las funciones anotadas, se destacan algunas en tanto que, por su relevancia en los procesos que se llevan a cabo y el valor de la información que reproducen, podría decirse que tienen un carácter de imprescindibles, por cuanto posibilitan la ejecución de procesos posteriores. Algunas de ellas son: Seg\_xie, metodología, guardado\_cluster, entre otras que, en lo sucesivo, tendrán un mayor desarrollo.

### **Análisis inicial del audio**

En este caso **Seg\_xie** es la función encargada de la extracción de la información inicial con la que trabaja el software. Por ello era importante tener una gran similitud entre los resultados de estas funciones para cada lenguaje. En las siguientes imágenes se observan las semejanzas de los audios analizados al usar la transformada rápida de Fourier.

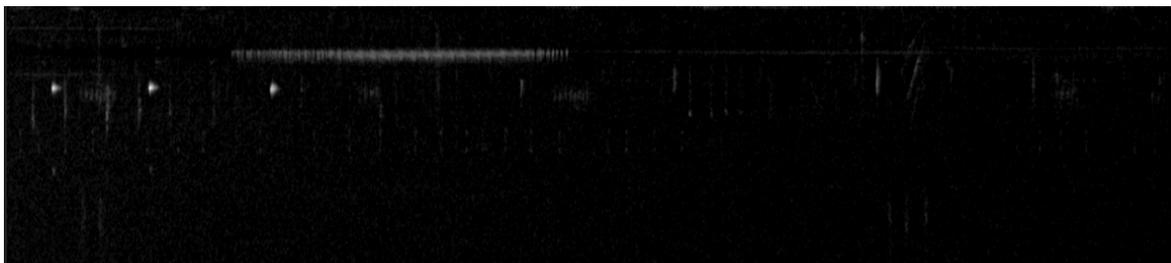


Fig. 5. Espectrograma del audio JAGUAS260\_20130131\_110352 en Python.

|   |  |         |            |
|---|--|---------|------------|
|  | <b>INFORME FINAL DE<br/>TRABAJO DE GRADO</b> | Código  | FDE 089    |
|   |  | Versión | 03         |
|   |  | Fecha   | 2015-01-22 |

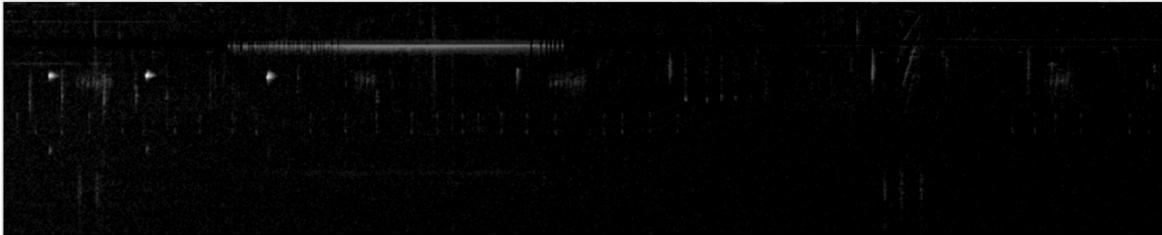


Fig.6. Espectrograma del audio JAGUAS260\_20130131\_110352 en Matlab.

Las pocas diferencias perceptibles son ocasionadas por lo que se comentó con anterioridad, esto es, diferencias esenciales en las librerías de los espectrogramas o algunos filtros de imagen. Sin embargo, estos errores se pueden ir reduciendo a medida que se pasa por los distintos filtros, hasta obtener resultados como el que se expone a continuación:

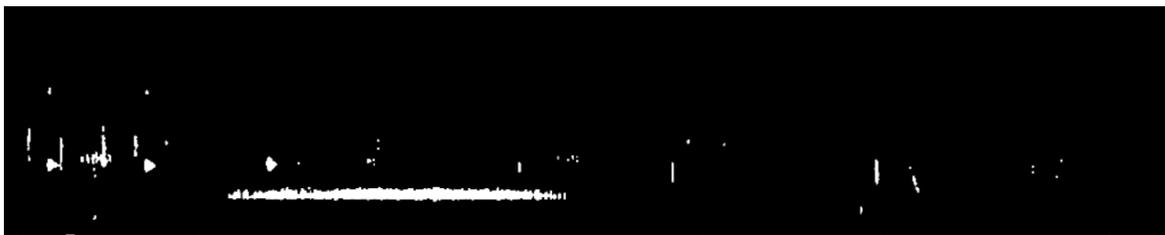


Fig. 7. Espectrograma con filtros morfológicos para el audio JAGUAS260\_20130131\_110352 en Python.

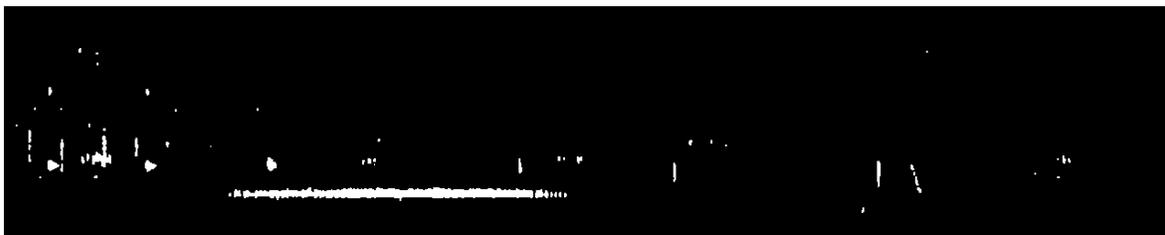


Fig. 8. Espectrograma con filtros morfológicos para el audio JAGUAS260\_20130131\_110352 en Matlab.

Tal como se muestra, esta imagen es luego utilizada para hacer la selección final de los elementos que pasarán a ser analizados y clasificados por el software. De un lado, en Matlab se eliminan de la imagen los elementos no aptos para clasificación, los cuales se

|   |                                      |         |            |
|---|--------------------------------------|---------|------------|
|  | INFORME FINAL DE<br>TRABAJO DE GRADO | Código  | FDE 089    |
|   |                                      | Versión | 03         |
|   |                                      | Fecha   | 2015-01-22 |

eligen mediante filtros de tamaño y forma; y, de otro, en Python ocurre el mismo proceso, pero los que no pasan son ignorados como se muestra a continuación.

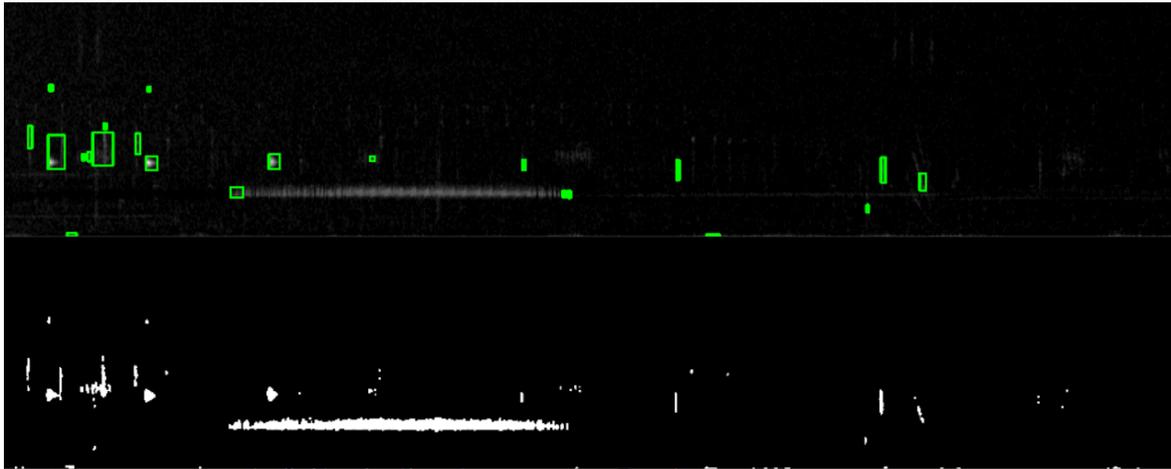


Fig. 9. Elementos finales detectados para el audio JAGUAS260\_20130131\_110352 en Python.

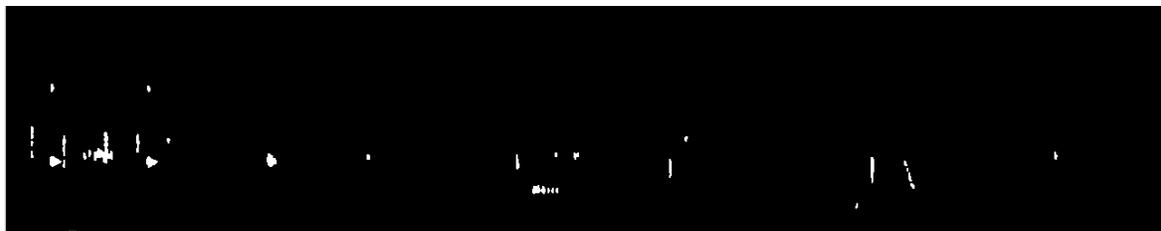


Fig. 10. Elementos finales detectados para el audio JAGUAS260\_20130131\_110352 en Matlab.

### Análisis y reporte de la información

La siguiente función más relevante es la que se denomina **metodología**, toda vez que en ésta se integran la mayoría de las funciones creadas con antelación. Dadas las circunstancias, ésta fue una de las funciones objetivo al comenzar la traducción del código en tanto aquella es la encargada de mostrar toda la información analizada de los audios elegidos.

|  |                                      |         |                |
|--|--------------------------------------|---------|----------------|
| <br>Institución Universitaria | INFORME FINAL DE<br>TRABAJO DE GRADO | Código  | FDE 089        |
|  |                                      | Versión | 03             |
|  |                                      | Fecha   | 2015-01-2<br>2 |

Continuando con la investigación, mediante la función metodología, se realizó la clasificación y extracción de características relevantes para los usuarios, así como los análisis estadísticos. Una vez obtenida, esta información es expuesta mediante una tabla que describe datos del segmento, como el nombre del archivo al que pertenece, la fecha y hora de la grabación, los tiempos inicial y final del canto en la grabación, las frecuencias máxima y mínima, la frecuencia dominante, el clúster al que pertenece, entre otros.

Posteriormente, toda esta información puede ser almacenada en Excel u otro tipo de archivo y puede ser visualizada, sin tener que realizar nuevamente el análisis; por lo demás, a partir de las otras variables de esta función, se abren las opciones para el guardado de clústers o la visualización en los espectrogramas de los elementos detectados; por lo que es importante leer una cantidad de elementos similares, a la vez que se tiene en cuenta la relevancia de estos. Los resultados al respecto son los siguientes:

|      | Sel.                     | File       | Month | Day | Hour | Minute | Start   | End     | Length | Fdom       | FminVoc    | FmaxVoc    | Fmin    | Fmax  | Cluster |
|------|--------------------------|------------|-------|-----|------|--------|---------|---------|--------|------------|------------|------------|---------|-------|---------|
| 1757 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 42.4112 | 43.0266 | 0.6153 | 2.9067e+03 | 2.6486e+03 | 6.2877e+03 | 21.5122 | 22050 | 18      |
| 1758 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 43.3284 | 43.5258 | 0.1974 | 2.9068e+03 | 2.7563e+03 | 3.0362e+03 | 21.5122 | 22050 | 5       |
| 1759 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 43.3400 | 43.5722 | 0.2322 | 6.6104e+03 | 6.3308e+03 | 6.8906e+03 | 21.5122 | 22050 | 9       |
| 1760 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 43.9205 | 44.2224 | 0.3019 | 2.8638e+03 | 2.7132e+03 | 3.0362e+03 | 21.5122 | 22050 | 2       |
| 1761 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 44.3733 | 44.5707 | 0.1974 | 2.8423e+03 | 2.7132e+03 | 3.0146e+03 | 21.5122 | 22050 | 4       |
| 1762 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 44.6984 | 45.2557 | 0.5573 | 3.1219e+03 | 2.6917e+03 | 6.4600e+03 | 21.5122 | 22050 | 18      |
| 1763 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 45.2789 | 45.4647 | 0.1858 | 2.8423e+03 | 2.7132e+03 | 3.0146e+03 | 21.5122 | 22050 | 4       |
| 1764 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 45.4182 | 45.5692 | 0.1509 | 5.3400e+03 | 5.1034e+03 | 5.6848e+03 | 21.5122 | 22050 | 4       |
| 1765 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 45.4298 | 45.6620 | 0.2322 | 4.3277e+03 | 3.8544e+03 | 4.6296e+03 | 21.5122 | 22050 | 12      |
| 1766 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 45.5343 | 45.6853 | 0.1509 | 9.8618e+03 | 9.4315e+03 | 9.9914e+03 | 21.5122 | 22050 | 9       |
| 1767 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 45.5924 | 45.8014 | 0.2090 | 2.8207e+03 | 2.7132e+03 | 2.9931e+03 | 21.5122 | 22050 | 4       |
| 1768 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 46.6605 | 46.9740 | 0.3135 | 2.9498e+03 | 2.7132e+03 | 6.5676e+03 | 21.5122 | 22050 | 11      |
| 1769 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 48.5878 | 48.8896 | 0.3019 | 3.0577e+03 | 3.0362e+03 | 5.9216e+03 | 21.5122 | 22050 | 11      |
| 1770 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 48.6226 | 48.8896 | 0.2670 | 6.3521e+03 | 6.1370e+03 | 6.5030e+03 | 21.5122 | 22050 | 16      |
| 1771 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 49.6094 | 49.8300 | 0.2206 | 4.3277e+03 | 3.8329e+03 | 4.6727e+03 | 21.5122 | 22050 | 12      |
| 1772 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 49.7371 | 49.8532 | 0.1161 | 5.0817e+03 | 4.9096e+03 | 5.4048e+03 | 21.5122 | 22050 | 4       |
| 1773 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 50.7820 | 51.1071 | 0.3251 | 3.1008e+03 | 3.0792e+03 | 5.9647e+03 | 21.5122 | 22050 | 11      |
| 1774 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 52.0127 | 52.3262 | 0.3135 | 3.2727e+03 | 2.9070e+03 | 6.3954e+03 | 21.5122 | 22050 | 11      |
| 1775 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 53.3478 | 53.4756 | 0.1277 | 5.5547e+03 | 4.6512e+03 | 5.7494e+03 | 21.5122 | 22050 | 4       |
| 1776 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 53.8122 | 53.9516 | 0.1393 | 8.7635e+03 | 8.2257e+03 | 9.2162e+03 | 21.5122 | 22050 | 9       |
| 1777 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 53.9864 | 54.3347 | 0.3483 | 3.2512e+03 | 2.9500e+03 | 6.3308e+03 | 21.5122 | 22050 | 11      |
| 1778 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 54.6249 | 54.9849 | 0.3599 | 5.3818e+03 | 3.8544e+03 | 5.6848e+03 | 21.5122 | 22050 | 3       |
| 1779 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 55.9601 | 56.2968 | 0.3367 | 3.3372e+03 | 2.9285e+03 | 6.4169e+03 | 21.5122 | 22050 | 11      |
| 1780 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 56.6451 | 56.8076 | 0.1625 | 5.5116e+03 | 4.5866e+03 | 5.7494e+03 | 21.5122 | 22050 | 4       |
| 1781 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 58.3285 | 58.7000 | 0.3715 | 3.3373e+03 | 2.9500e+03 | 6.4169e+03 | 21.5122 | 22050 | 11      |
| 1782 | <input type="checkbox"/> | JAGUAS2... | 1     | 17  | 10   | 45     | 58.8858 | 59.0948 | 0.2090 | 4.3708e+03 | 3.8760e+03 | 4.6512e+03 | 21.5122 | 22050 | 12      |

Fig. 11. Tabla de reporte de segmentos en Matlab con 1783 de éstos detectados, con 22 clústers, para 15 audios (visualización en interfaz).

|  |                                      |  |  |  |  |  |  |  |         |            |
|--|--------------------------------------|--|--|--|--|--|--|--|---------|------------|
| <br>Institución Universitaria | INFORME FINAL DE<br>TRABAJO DE GRADO |  |  |  |  |  |  |  | Código  | FDE 089    |
|  |                                      |  |  |  |  |  |  |  | Versión | 03         |
|  |                                      |  |  |  |  |  |  |  | Fecha   | 2015-01-22 |

|      |                      |    |    |   |   |             |             |             |             |             |             |             |       |    |
|------|----------------------|----|----|---|---|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------|----|
| 1656 | JAGUAS259_2013011... | 11 | 14 | 8 | 1 | 14.87012... | 15.16688... | 0.296757... | 3616.947... | 2971.582... | 6653.759... | 21.51219... | 22050 | 5  |
| 1657 | JAGUAS259_2013011... | 11 | 14 | 8 | 1 | 48.66175... | 48.89400... | 0.232244... | 6502.670... | 6136.962... | 6675.292... | 21.51219... | 22050 | 2  |
| 1658 | JAGUAS259_2013011... | 11 | 14 | 8 | 1 | 16.43132... | 16.62486... | 0.193537... | 6502.628... | 6093.896... | 6675.292... | 21.51219... | 22050 | 11 |
| 1659 | JAGUAS259_2013011... | 11 | 14 | 8 | 1 | 36.40438... | 36.64953... | 0.245147... | 6717.771... | 6115.429... | 6696.826... | 21.51219... | 22050 | 11 |
| 1660 | JAGUAS259_2013011... | 11 | 14 | 8 | 1 | 25.73402... | 26.55978... | 0.825759... | 3272.395... | 2605.517... | 6696.826... | 21.51219... | 22050 | 3  |
| 1661 | JAGUAS259_2013011... | 11 | 14 | 8 | 1 | 46.72638... | 46.95862... | 0.232244... | 6631.701... | 6093.896... | 6782.958... | 21.51219... | 22050 | 4  |
| 1662 | JAGUAS259_2013011... | 11 | 14 | 8 | 1 | 21.32137... | 21.57942... | 0.258049... | 6481.115... | 6093.896... | 6782.958... | 21.51219... | 22050 | 11 |
| 1663 | JAGUAS259_2013011... | 11 | 14 | 8 | 1 | 39.95257... | 40.15901... | 0.206439... | 6588.761... | 6180.029... | 6804.492... | 21.51219... | 22050 | 11 |
| 1664 | JAGUAS259_2013011... | 11 | 14 | 8 | 1 | 10.78003... | 11.08969... | 0.309659... | 3617.031... | 3057.714... | 6826.025... | 21.51219... | 22050 | 5  |
| 1665 | JAGUAS259_2013011... | 11 | 14 | 8 | 1 | 43.35883... | 43.56527... | 0.206439... | 6675.019... | 6395.361... | 6890.625    | 21.51219... | 22050 | 2  |
| 1666 | JAGUAS259_2013011... | 11 | 14 | 8 | 1 | 1.593458... | 1.890215... | 0.296757... | 6932.872... | 6093.896... | 6912.158... | 21.51219... | 22050 | 11 |
| 1667 | JAGUAS259_2013011... | 11 | 14 | 8 | 1 | 35.06252... | 35.19155... | 0.129024... | 6589.076... | 6503.027... | 7170.556... | 21.51219... | 22050 | 2  |
| 1668 | JAGUAS259_2013011... | 11 | 14 | 8 | 1 | 4.238469... | 4.406201... | 0.167732... | 7191.081... | 6158.496... | 7170.556... | 21.51219... | 22050 | 4  |
| 1669 | JAGUAS259_2013011... | 11 | 14 | 8 | 1 | 37.43658... | 37.65592... | 0.219342... | 8461.561... | 7450.488... | 8699.414... | 21.51219... | 22050 | 0  |
| 1670 | JAGUAS259_2013011... | 11 | 14 | 8 | 1 | 53.84856... | 53.95178... | 0.103219... | 8419.314... | 8247.216... | 8850.146... | 21.51219... | 22050 | 2  |
| 1671 | JAGUAS259_2013011... | 11 | 14 | 8 | 1 | 38.98488... | 39.11391... | 0.129024... | 9215.706... | 8699.414... | 9689.941... | 21.51219... | 22050 | 8  |
| 1672 | JAGUAS259_2013011... | 11 | 14 | 8 | 1 | 30.54665... | 30.66277... | 0.116122... | 9861.681... | 9323.876... | 9926.806... | 21.51219... | 22050 | 8  |

Fig. 12. Tabla de reporte con 1.672 segmentos y 14 clústers, para 15 audios, en Python (en la interfaz esta se guarda como csv).

Como se puede apreciar, en Matlab hay una cantidad de 1.783 segmentos, mientras que, en Python, para los mismos audios, se tiene un número de 1.678 segmentos, cada uno, con 22 y 15 clústers, respectivamente. Entonces, para los 15 audios JAGUAS, analizados, se tiene una diferencia del 5.8 % en la cantidad de segmentos, lo cual puede justificarse en los cambios realizados a los filtros, con el fin de aumentar la calidad, afectando igualmente el número de clústers.

### Guardado de elementos

En orden consecutivo, sigue la función **guardado\_cluster** la cual está encargada de la creación de los perfiles de especies que luego son reconocidos por el algoritmo; éstos constituyen una estructura compleja de celdas que se encarga de tomar y agrupar varios segmentos, pertenecientes a un mismo clúster, en una única fila, con una etiqueta que describe con que especie se está relacionando. Para elegir los segmentos que harán parte de este etiquetado de reconocimiento, Matlab cuenta con una interfaz gráfica que permite

|  |                                      |         |                |
|--|--------------------------------------|---------|----------------|
| <br>Institución Universitaria | INFORME FINAL DE<br>TRABAJO DE GRADO | Código  | FDE 089        |
|  |                                      | Versión | 03             |
|  |                                      | Fecha   | 2015-01-2<br>2 |

seleccionar, uno por uno, los elementos que se pretende utilizar; por su parte, en Python, ante la falta de esta función, se creó un sistema que unifica todos los segmentos pertenecientes a un mismo clúster, permitiendo, de esta manera, tener una estructura confiable de forma rápida.

Las estructuras creadas son bastante similares, en este caso, a las de Python y se les da la etiqueta de “ensayo”, con el número consecutivo correspondiente al clúster; en el caso de Matlab, éstas fueron estudiadas, en su momento, por biólogos y luego otorgadas a los investigadores como referencia.

Debido a las pocas diferencias en los espectrogramas y números de segmentos encontrados en los audios, la calidad de las etiquetas construidas en Matlab se pierde si se trata de usar en Python, por lo cual cada grupo de etiquetas restringe, de cierta forma, su uso únicamente al lenguaje del cual fueron extraídos los segmentos con los que se elaboraron.

|           |           |          |                |              |              |   |   |   |           |           |            |              |              |              |              |              |            |              |
|-----------|-----------|----------|----------------|--------------|--------------|---|---|---|-----------|-----------|------------|--------------|--------------|--------------|--------------|--------------|------------|--------------|
| [Ensayo3] | [[21.5... | [[220... | [[0, 1, 2, ... | [[236.655... | [[18323.3... | 0 | 0 | 0 | [[0.55... | [[2.74... | [[11071... | [[12321.7... | [[1.55573... | [[11218.7... | [[0.18575... | [[602.929... | [[0.0]]    | [[0.26330... |
| [Ensayo2] | [[21.5... | [[220... | [[0, 1, 2, ... | [[236.655... | [[18323.3... | 0 | 0 | 0 | [[29.2... | [[31.6... | [[2019...  | [[3471.58... | [[30.2091... | [[2519.38... | [[0.47600... | [[452.197... | [[0.255... | [[0.26107... |
| [Ensayo7] | [[21.5... | [[220... | [[0, 1, 2, ... | [[236.655... | [[18323.3... | 0 | 0 | 0 | [[52.6... | [[54.9... | [[3397...  | [[5194.23... | [[53.6032... | [[3897.50... | [[0.38312... | [[796.728... | [[2.182... | [[0.40238... |
| [Ensayo1] | [[21.5... | [[220... | [[0, 1, 2, ... | [[236.655... | [[18323.3... | 0 | 0 | 0 | [[43.1... | [[45.3... | [[3569...  | [[5237.30... | [[44.1179... | [[4069.77... | [[0.22058... | [[667.529... | [[0.139... | [[0.36608... |
| [Ensayo0] | [[21.5... | [[220... | [[0, 1, 2, ... | [[236.655... | [[18323.3... | 0 | 0 | 0 | [[31.7... | [[33.9... | [[3311...  | [[5323.43... | [[32.7169... | [[3811.37... | [[0.26702... | [[1012.06... | [[0.185... | [[0.22879... |
| [Ensayo6] | [[21.5... | [[220... | [[0, 1, 2, ... | [[236.655... | [[18323.3... | 0 | 0 | 0 | [[16.0... | [[18.7... | [[3203...  | [[5388.03... | [[17.0318... | [[3703.71... | [[0.74303... | [[1184.32... | [[7.534... | [[0.23224... |
| [Ensayo4] | [[21.5... | [[220... | [[0, 1, 2, ... | [[236.655... | [[18323.3... | 0 | 0 | 0 | [[18.0... | [[20.7... | [[5658...  | [[9307.08... | [[19.0287... | [[6158.49... | [[0.75464... | [[2648.58... | [[0.0]]    | [[0.62151... |
| [Ensayo5] | [[21.5... | [[220... | [[0, 1, 2, ... | [[236.655... | [[18323.3... | 0 | 0 | 0 | [[56.7... | [[59.0... | [[6304...  | [[9522.41... | [[57.7944... | [[6804.49... | [[0.25541... | [[2217.91... | [[0.127... | [[0.20450... |

Fig. 13. Segmentos extraídos en Python de forma automática.

|                            | 1       | 2     | 3           | 4           | 5           | 6 | 7 | 8 | 9       | 10      | 11         | 12         | 13      | 14         | 15     | 16         | 17      | 18 | 19          |
|----------------------------|---------|-------|-------------|-------------|-------------|---|---|---|---------|---------|------------|------------|---------|------------|--------|------------|---------|----|-------------|
| 'Leucosthetus_jota_6'      | 11.7073 | 12000 | 1x27 double | 1x27 double | 1x27 double | 0 | 0 | 0 | 21.9333 | 24.1680 | 3.6133e+03 | 5.0586e+03 | 22.9333 | 4.1133e+03 | 0.2347 | 445.3125   | 0       |    | 3x27 double |
| 'gularis_1'                | 11.7073 | 12000 | 1x27 double | 1x27 double | 1x27 double | 0 | 0 | 0 | 29.0160 | 31.3147 | 2.3477e+03 | 3.6641e+03 | 30.0160 | 2.8477e+03 | 0.2987 | 316.4063   | 1.9840  |    | 3x27 double |
| 'gularis_3'                | 11.7073 | 12000 | 1x27 double | 1x27 double | 1x27 double | 0 | 0 | 0 | 4.2053  | 6.6320  | 2.3477e+03 | 3.6758e+03 | 5.2053  | 2.8477e+03 | 0.4267 | 328.1250   | 0.4693  |    | 3x27 double |
| 'Espadarana_prosoblepon_6' | 11.7073 | 12000 | 1x27 double | 1x27 double | 1x27 double | 0 | 0 | 0 | 50.1573 | 52.5840 | 4.5273e+03 | 6.1953e+03 | 51.1573 | 5.0273e+03 | 0.4267 | 667.9688   | 22.5920 |    | 3x27 double |
| 'Espadarana_prosoblepon_3' | 11.7073 | 12000 | 1x27 double | 1x27 double | 1x27 double | 0 | 0 | 0 | 43.5653 | 45.9280 | 4.5977e+03 | 6.1367e+03 | 44.5653 | 5.0977e+03 | 0.3627 | 539.0625   | 0       |    | 3x27 double |
| 'boana_Boans_1'            | 11.7073 | 12000 | 1x27 double | 1x27 double | 1x27 double | 0 | 0 | 0 | 30.0400 | 32.2533 | 578.1250   | 1.8125e+03 | 31.0400 | 1.0781e+03 | 0.2133 | 234.3750   | 6.1547  |    | 3x27 double |
| 'boana_Boans_2'            | 11.7073 | 12000 | 1x27 double | 1x27 double | 1x27 double | 0 | 0 | 0 | 30.9147 | 33.1493 | 589.8438   | 1.8125e+03 | 31.9147 | 1.0898e+03 | 0.2347 | 222.6563   | 0.5760  |    | 3x27 double |
| 'Hyloxalus_4'              | 21.5122 | 22050 | 1x27 double | 1x27 double | 1x27 double | 0 | 0 | 0 | 39.0312 | 41.3215 | 3.6344e+03 | 5.3665e+03 | 40.0312 | 4.1344e+03 | 0.2902 | 732.1289   | 0.1509  |    | 3x27 double |
| 'Hyloxalus_2'              | 21.5122 | 22050 | 1x27 double | 1x27 double | 1x27 double | 0 | 0 | 0 | 41.5622 | 43.8640 | 3.8066e+03 | 5.4742e+03 | 42.5622 | 4.3066e+03 | 0.3019 | 667.5293   | 7.3607  |    | 3x27 double |
| 'aa'                       | 21.5122 | 22050 | 1x27 double | 1x27 double | 1x27 double | 0 | 0 | 0 | 58.5011 | 60.9888 | 5.1202e+03 | 1.0513e+04 | 59.5011 | 5.6202e+03 | 0.4876 | 4.3928e+03 | 0       |    | 3x27 double |

Fig. 14. Segmentos extraídos en Matlab de forma manual por biólogos.

|   |                                      |         |            |
|---|--------------------------------------|---------|------------|
|  | INFORME FINAL DE<br>TRABAJO DE GRADO | Código  | FDE 089    |
|   |                                      | Versión | 03         |
|   |                                      | Fecha   | 2015-01-22 |

Para poder usar estos segmentos, sólo es necesario hacerlo mediante la función **Metodologia\_prueba**, que conlleva un proceso idéntico al realizado por **Metodología**, con la diferencia que, en esta ocasión, en vez de clasificar los todos segmentos que son detectados, separándolos en grupos, lo que ésta hace es compararlos directamente con las etiquetas que se estén buscando y realiza un rastreo por el rango de frecuencias del clúster asignado, agrupando todo lo que coincida lo suficiente, según el estudio, y asignando a las demás una clase adicional.

### Visualización de segmentos

Frente a este asunto, parece oportuno indicar de donde fueron tomados algunos de estos segmentos, para poder tener una representación gráfica que permitiera determinar la exactitud de las selecciones, así como hacer cambios en ciertos aspectos del código. Para ello se construyó un código adicional que facilitara la visualización de un elemento representativo de cada clúster, siendo éste el que mayor porcentaje de pertenencia tenía en relación con el valor medio de todos los elementos en los que fue agrupado dicho clúster.

De manera complementaria, el código en mención permite visualizar todos los segmentos agrupados en clústers específicos, como se aprecia en las siguientes figuras.

|  |                                      |         |            |
|--|--------------------------------------|---------|------------|
| <br>Institución Universitaria | INFORME FINAL DE<br>TRABAJO DE GRADO | Código  | FDE 089    |
|  |                                      | Versión | 03         |
|  |                                      | Fecha   | 2015-01-22 |

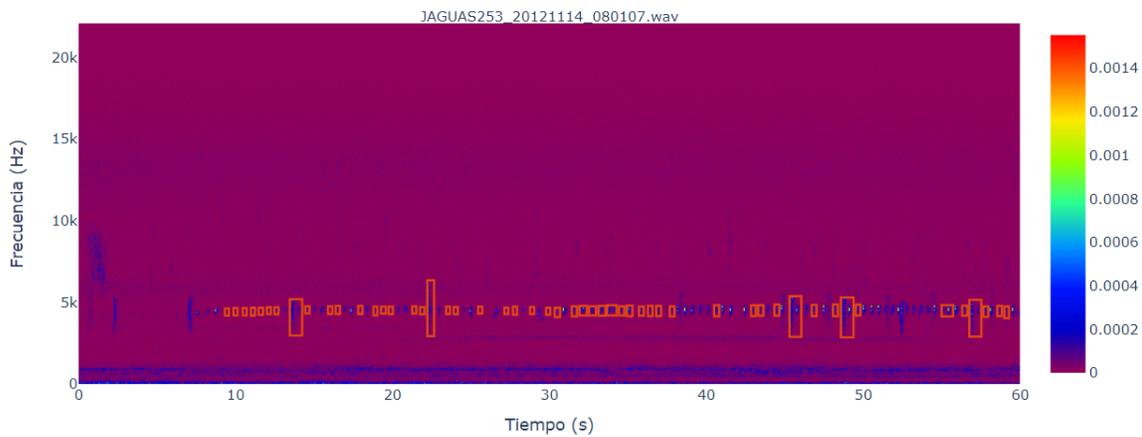


Fig. 15. Elementos marcados como clúster 3 para el análisis del audio indicado en Python.

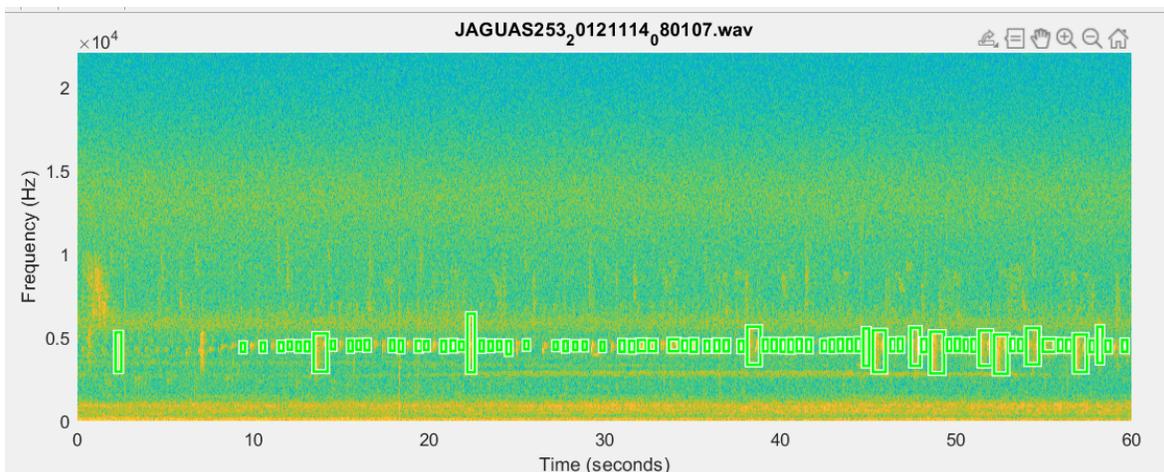


Fig. 16. Elementos marcados como cluster 3 para el análisis del audio indicado en Python.

|   |  |         |            |
|---|--|---------|------------|
|  | <b>INFORME FINAL DE<br/>TRABAJO DE GRADO</b> | Código  | FDE 089    |
|   |  | Versión | 03         |
|   |  | Fecha   | 2015-01-22 |

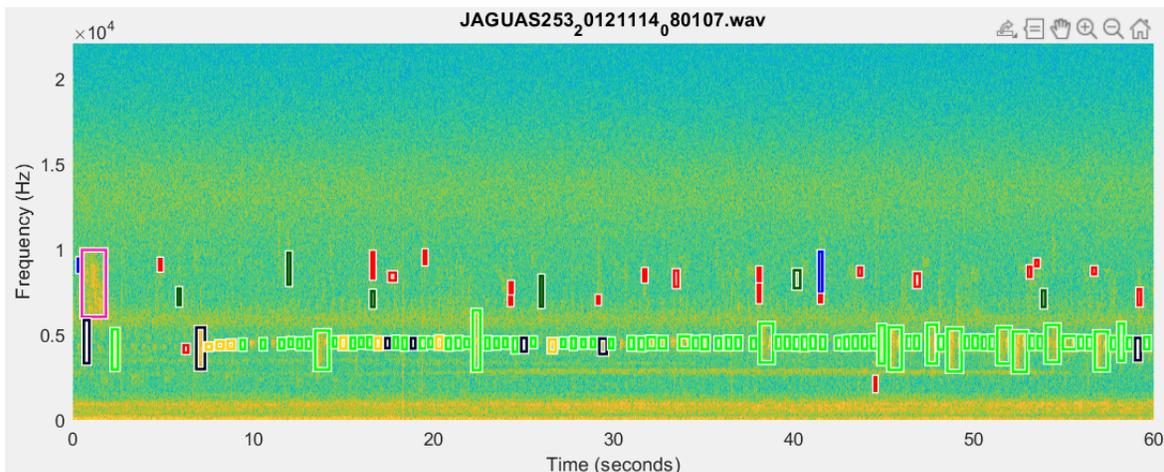


Fig. 17. Todos los elementos detectados en el audio indicado por el algoritmo de Matlab (5 clústers).

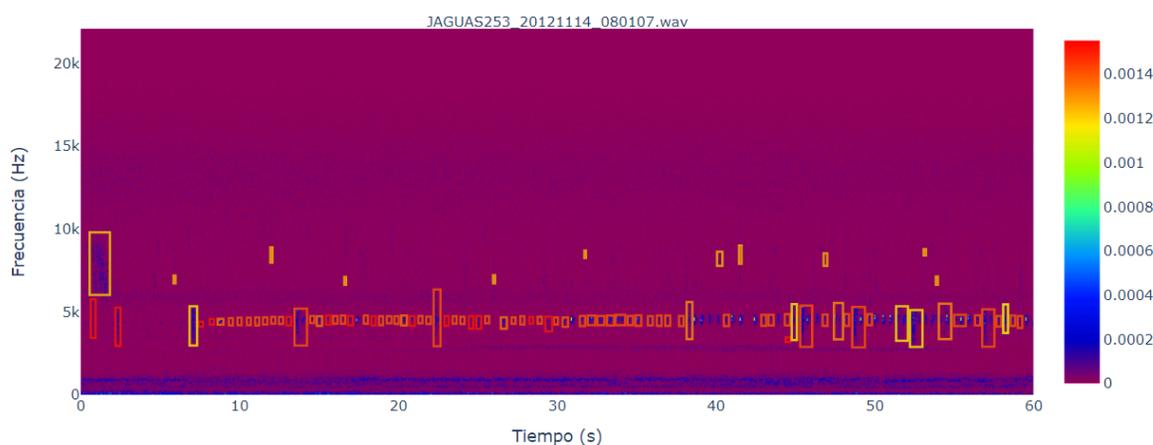


Fig. 18. Todos los elementos detectados en el audio indicado por el algoritmo de Python (5 clústers).

Como se evidencia en las imágenes, la cantidad de elementos detectados en los audios es similar, pero son levemente menores los detectados por Python, lo que puede causar algunas variaciones al momento de crear los clústers. No obstante, en general, la mayoría de las detecciones son idénticas, incluso la numeración de los clústers suele coincidir con los de Matlab, si se analizan los mismos audios. Esta funcionalidad es bastante importante, en tanto que, gracias a ella, se pueden filtrar los segmentos adecuados de cada clúster con el objetivo de crear un perfil de búsqueda etiquetado de mayor calidad.

|   |  |         |            |
|---|--|---------|------------|
|  | <b>INFORME FINAL DE<br/>TRABAJO DE GRADO</b> | Código  | FDE 089    |
|   |  | Versión | 03         |
|   |  | Fecha   | 2015-01-22 |

En Python, para seleccionar el elemento representativo de cada clúster, se debe abrir el espectrograma al que pertenezca el elemento representativo y marcar únicamente este segmento; en cambio, en Matlab, se hace un zoom en la zona marcada y se abre una nueva ventana con información adicional. En este estudio no se consideró importante implementar esta función, toda vez que la librería gráfica permite realizar un zoom manual, si el usuario así lo requiere.

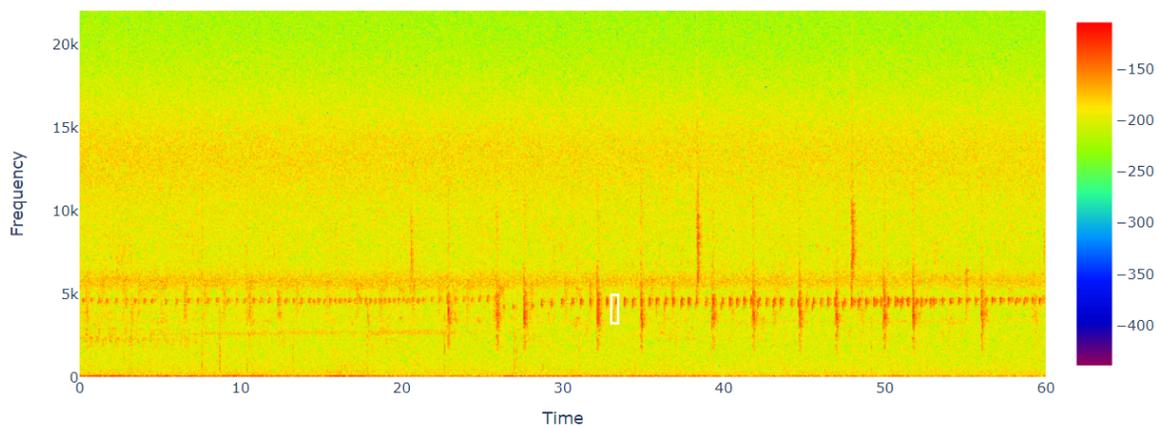


Fig. 19. Segmento representativo para el clúster número en Python 3.

|   |  |         |            |
|---|--|---------|------------|
|  | <b>INFORME FINAL DE<br/>TRABAJO DE GRADO</b> | Código  | FDE 089    |
|   |  | Versión | 03         |
|   |  | Fecha   | 2015-01-22 |

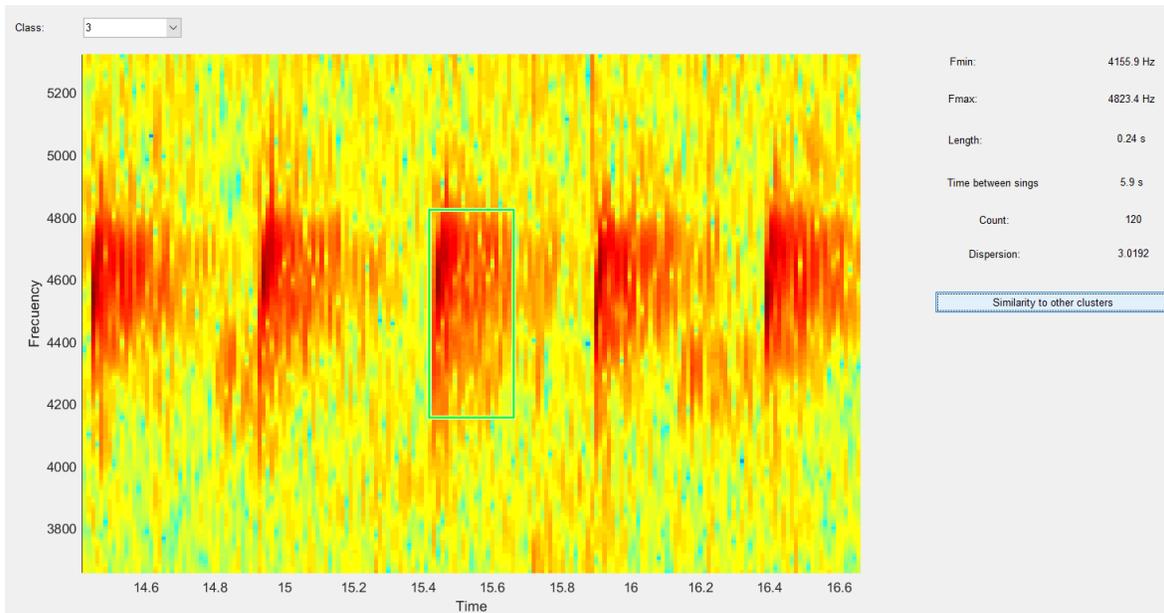


Fig. 20. Segmento representativo del clúster número 3 en Matlab.

### Patrones temporales del sonotipo

Finalmente, se implementó una gráfica que permitiera realizar el análisis temporal de los clústers encontrados, pues cada audio contiene información de la hora y el minuto en el que fue tomado. Como consecuencia de lo anterior, se realiza un conteo de cuántos segmentos de un clúster están asociados a un audio y, con esto, se puede construir un diagrama de reloj que permite dibujar las horas en las que fueron encontrados los sonotipos. En el siguiente ejemplo se presentan los diagramas de actividad obtenidos en una carpeta con 15 audios, los cuales sólo contienen audios tomados en el día. En ellos se puede ver que cada clúster tiene una distribución y que muchos de los sonotipos tienen información alrededor de las 8:00 a.m., con base en este resultado, un experto puede entrar a verificar si diferentes sonotipos son, en realidad, parte del mismo evento o si son especies que tienen un patrón de actividad similar.

|   |  |         |                |
|---|--|---------|----------------|
|  | <b>INFORME FINAL DE<br/>TRABAJO DE GRADO</b> | Código  | FDE 089        |
|   |  | Versión | 03             |
|   |  | Fecha   | 2015-01-2<br>2 |

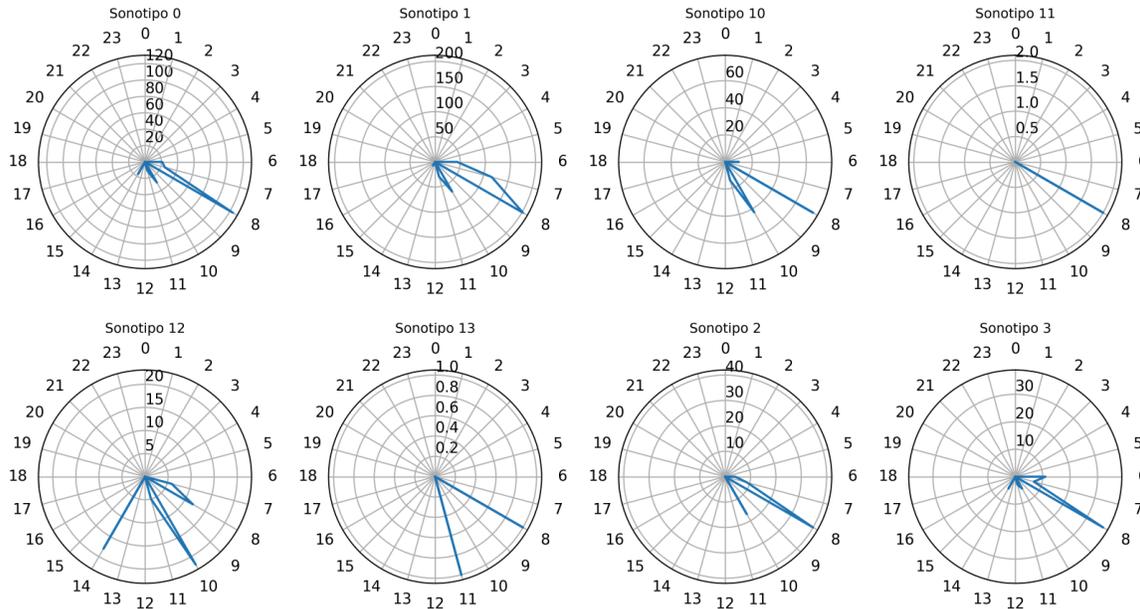


Fig. 21. Patrones de actividad temporal de los sonotipos. Cada sonotipo tiene una distribución temporal. En la gráfica se observa que algunos tienen más vocalizaciones a las 8:00 hrs, sin embargo, otros, como el sonotipo 12, tienen vocalizaciones a las 14:00 hrs.

En relación con lo expresado, cabe aclarar que, para la programación de este módulo, se realizó la extracción de las horas de cada audio, así como el conteo de los clústers en cada uno de ellos; más, la parte gráfica, fue realizada por el programador de frontend.

### Interfaz gráfica

Todo el código mencionado fue usado como “backend” para una interfaz gráfica de la funcionalidad de **SONOTIPOS**, realizada en HTML, con el fin de brindarle al usuario una mayor facilidad para utilizar el software y las funcionalidades hasta ahora implementadas.

|   |  |         |            |
|---|--|---------|------------|
|  | <b>INFORME FINAL DE<br/>TRABAJO DE GRADO</b> | Código  | FDE 089    |
|   |  | Versión | 03         |
|   |  | Fecha   | 2015-01-22 |

Este módulo sugiere automáticamente segmentos en los que posiblemente hay vocalizaciones de animales. Para llegar a esto, el software realiza un análisis no supervisado mediante “clustering” y propone agrupaciones de segmentos llamados sonotipos, de tal manera que el usuario asocia los sonotipos propuestos a vocalizaciones/llamados de una especie de interés. Así las cosas, se puede almacenar la información para, más adelante, reconocer las vocalizaciones/llamados de las mismas especies en nuevos audios.

Para su operación, este módulo se divide en tres procesos que, si bien son independientes, también son secuenciales. Entonces, para realizar el entrenamiento/reconocimiento con audios adquiridos en campo, los tres procesos que se pueden realizar son sonotipo, reconocer y patrones temporales del sonotipo.

#### Sonotipo

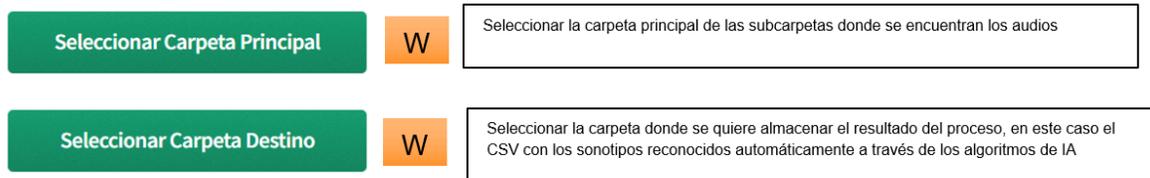


Fig. 22. Infografía de funcionamiento del software *Ecosonos* en su interfaz de Python para la funcionalidad sonotipos

Una vez seleccionadas la carpeta principal y de destino, se activan las casillas de Frecuencia Mínima (Hz) y Frecuencia Máxima (Hz), lo que permite hacer el análisis. Si no se indican estos valores, el programa tomará los mínimos y máximos que tienen los audios.

|   |  |         |            |
|---|--|---------|------------|
| <br><b>Institución Universitaria</b> | <b>INFORME FINAL DE<br/>TRABAJO DE GRADO</b> | Código  | FDE 089    |
|   |  | Versión | 03         |
|   |  | Fecha   | 2015-01-22 |

Sonotipo

**Seleccionar Carpeta Principal**

Frecuencia Mínima (Hz)

Ingrese frecuencia mínima

Frecuencia Máxima (Hz)

Ingrese frecuencia máxima

W

Espacio para ingresar la frecuencia mínima y máxima en las cuales se desea realizar el procesamiento.

**Procesar Carpetas**

Carpetas seleccionadas

- Audios
- Audio\_Prueba

Fig. 23. Infograma funcionamiento de interfaz.

Una vez seleccionadas las frecuencias deseadas, se debe seleccionar el botón “Procesar Carpetas” para iniciar el proceso. Al hacer esto, aparece una barra que indica el avance en el procesamiento.



Fig. 24. Infograma funcionamiento de interfaz.

Cuando finaliza la fase anterior, se habilitan los clústers que el sistema reconoció, además de una lista con los audios que fueron procesados.

|   |  |         |            |
|---|--|---------|------------|
| <br><b>Institución Universitaria</b> | <b>INFORME FINAL DE<br/>TRABAJO DE GRADO</b> | Código  | FDE 089    |
|   |  | Versión | 03         |
|   |  | Fecha   | 2015-01-22 |

Seleccione Clusters

- 0    1    2    3    4    5    6    7    8    9    10    11    12    13
- [JAGUAS253\\_20121114\\_080107.wav](#)
  - [JAGUAS253\\_20121114\\_081107.wav](#)
  - [JAGUAS253\\_20121114\\_082107.wav](#)
  - [JAGUAS253\\_20121114\\_083107.wav](#)
  - [JAGUAS253\\_20121114\\_084107.wav](#)
  - [JAGUAS253\\_20121122\\_061107.wav](#)
  - [JAGUAS253\\_20121122\\_070107.wav](#)
  - [JAGUAS253\\_20130117\\_075446.wav](#)
  - [JAGUAS253\\_20130117\\_081446.wav](#)
  - [JAGUAS253\\_20130119\\_100446.wav](#)
  - [JAGUAS259\\_20121114\\_144604.wav](#)
  - [JAGUAS259\\_20121115\\_104604.wav](#)
  - [JAGUAS259\\_20121115\\_110604.wav](#)
  - [JAGUAS259\\_20121116\\_081604.wav](#)
  - [JAGUAS259\\_20130117\\_104515.wav](#)
  - [164115\\_20170605\\_083200.wav](#)

Fig. 25. Infograma manejo de resultados en la interfaz.

Ahora, para verificar el procesamiento se pueden escoger los clústers deseados y un audio para analizar y visualizar los segmentos encontrados. Así también, el programa permite seleccionar los segmentos representativos de cada clúster, los cuales, al ser analizados, pueden dar indicios de las características de vocalización de una especie, así como se mostró con anterioridad, presentando los mismos diagramas y espectrogramas.

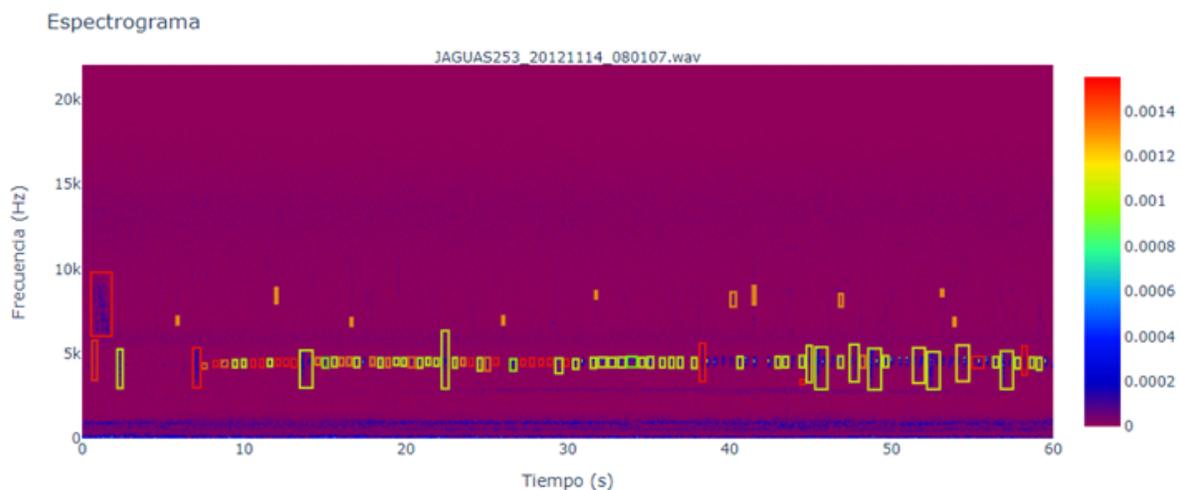


Fig. 26. Visualización de los elementos encontrados.

|   |  |         |            |
|---|--|---------|------------|
|  | <b>INFORME FINAL DE<br/>TRABAJO DE GRADO</b> | Código  | FDE 089    |
|   |  | Versión | 03         |
|   |  | Fecha   | 2015-01-22 |

El resultado de este módulo es un archivo, con extensión csv, que indica información procesada en **metodología** tal como: el nombre del archivo, el mes, día y año, el minuto, el inicio y final, la longitud, las frecuencias dominante, máxima y mínima del análisis, el clúster al que fue asignado y la membresía. La señalada información es dada en relación con cada segmento encontrado en el análisis.

Matlab cuenta con su propio sistema de creación de interfaces, el cual fue usado para construir el frontend del software de este lenguaje. Como comparación, a continuación se muestra la interfaz del software en matlab llamada “Áureas”, esta es la interfaz que permite la selección de la carpeta con los audios y así mismo habilita la posibilidad de elegir especies previamente seleccionadas para su entrenamiento.

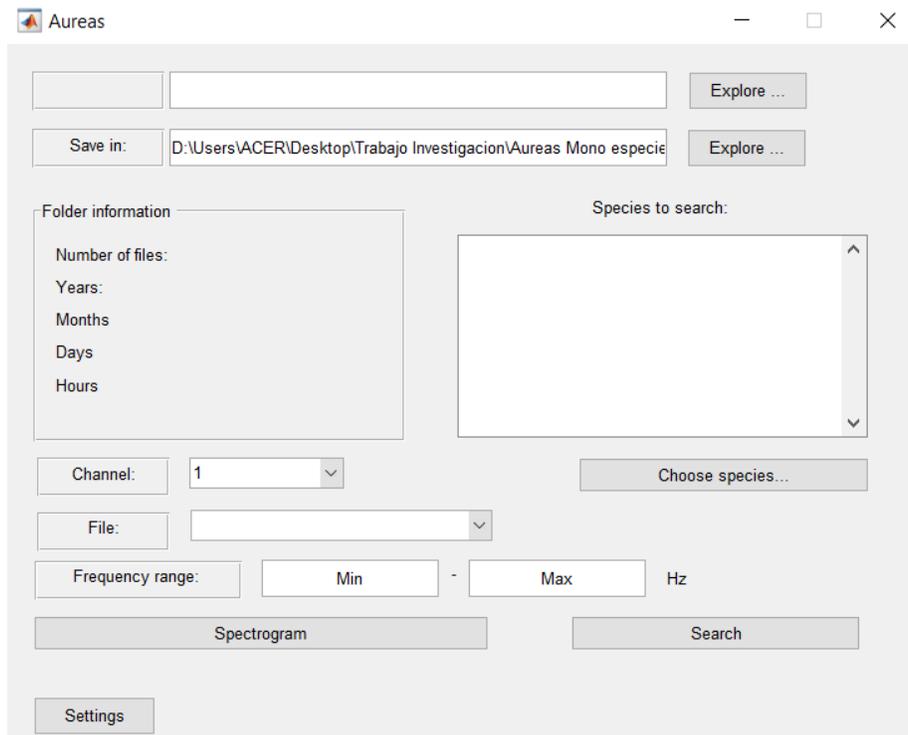


Fig 27. Interfaz de Áureas para el software ecozonos versión matlab

|   |  |         |            |
|---|--|---------|------------|
|  | <b>INFORME FINAL DE<br/>TRABAJO DE GRADO</b> | Código  | FDE 089    |
|   |  | Versión | 03         |
|   |  | Fecha   | 2015-01-22 |

Este software nuevo está pensado para la distribución a profesionales en biología e ingeniería, por ende tiene una interfaz más limpia e intuitiva claramente modularizada (fig 28), diseñada y programada por todo un equipo de desarrollo de software, tanto de backend y frontend. Este trabajo de grado permitió el desarrollo de backend del módulo de sonotipos y reconocimiento el cual se muestra en la figura 29 con un vistazo más general de lo mostrado en la sección de interfaz gráfica.

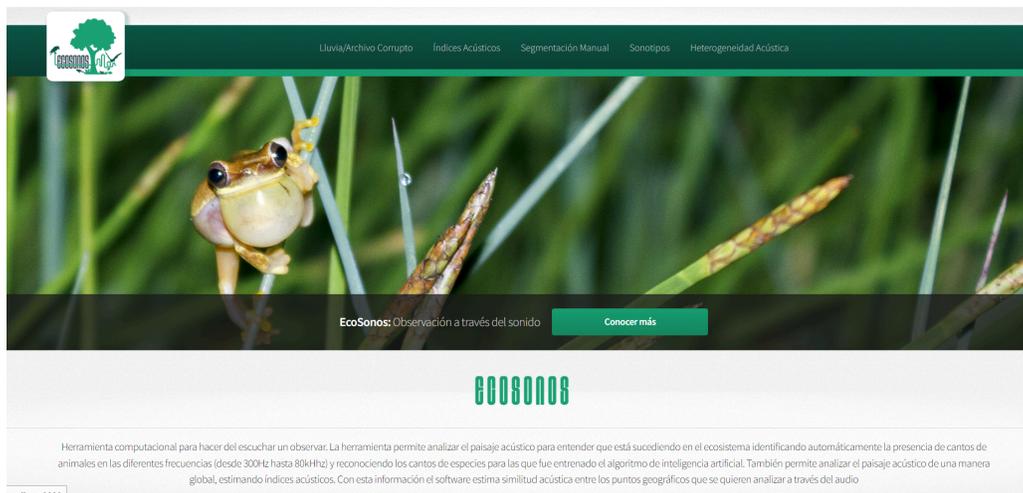


Fig 28. Interfaz de Ecosonos para el software ecozonos versión Python

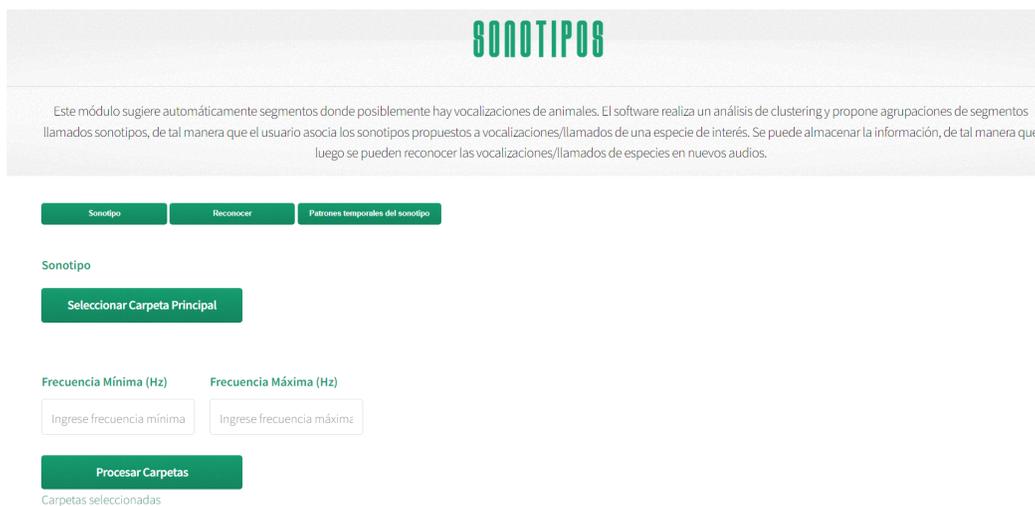


Fig 29. Interfaz de usuario para el módulo sonotipos para el software ecozonos versión Python

|   |                                      |         |            |
|---|--------------------------------------|---------|------------|
|  | INFORME FINAL DE<br>TRABAJO DE GRADO | Código  | FDE 089    |
|   |                                      | Versión | 03         |
|   |                                      | Fecha   | 2015-01-22 |

## 5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

---

### Conclusiones

- Como producto de laboratorio, para el grupo de Máquinas Inteligentes y Reconocimiento de Patrones (MIRP), se logró tener un software funcional, que cumple con las expectativas respecto a su homólogo, Matlab, habiendo completado las principales funciones de análisis y visualización de información, con la subsiguiente creación de un frontend en HTML con el que se facilita su interacción.
- Como resultado de la explotación del código otorgado y de las diferencias entre lenguajes y las herramientas disponibles, se modificó la forma en la cual fueron construidas algunas funciones. Así mismo, surgieron funciones adicionales o se modificó levemente el funcionamiento de otras.
- De otro lado, se determinó que un cambio en la estructura de la interfaz podría resultar más beneficioso, ya que, por ejemplo, se unificaron varios códigos de diferentes pantallas en uno solo con el fin de aportar simplicidad y facilitar, en la mayor medida posible, la creación del frontend.
- Se logró crear un sistema de reconocimiento que es capaz de tomar elementos, detectados con anterioridad, para luego permitir su búsqueda en nuevos audios.

|   |  |         |            |
|---|--|---------|------------|
|  | <b>INFORME FINAL DE<br/>TRABAJO DE GRADO</b> | Código  | FDE 089    |
|   |  | Versión | 03         |
|   |  | Fecha   | 2015-01-22 |

Aunque éstos presentan ciertos inconvenientes, a causa del método de agrupación de segmentos que se ha implementado lo cual puede llevar al software a tomar elementos poco relevantes lo que puede afectar la calidad de las detecciones. A pesar de eso esto nos abre la posibilidad a determinar las especies más activas y nos brinda la información necesaria para poder proponer diagramas que nos permitan describir el comportamiento de las especies de la zona o sus patrones temporales con lo cual se lograron crear diagramas que describen estos comportamientos.

- A pesar de las diferencias que se pueden presentar entre ambos softwares, versión Matlab y versión Python, la cantidad y calidad de los segmentos y clústers seleccionados es consistente con lo encontrado en Matlab; pudiendo identificar ya algunos fenotipos experimentales. Adicional a esto, las variables de las cuales depende la creación de clústers y selección de segmentos pueden ser fácilmente manipuladas, por lo que hay una gran posibilidad de optimización o un proceso de calibración que puede ser llevado a cabo por expertos, para mejorar la calidad de las selecciones.

## **Recomendaciones**

El código hecho en Matlab fue desarrollado a lo largo de varios años, por lo que hay que tener en cuenta su extensión y complejidad, ya que posee muchos módulos y funcionalidades aparte del de reconocimiento de especies en el cual se enfocó este proyecto, sin mencionar los errores presentes en algunas partes internas del código o una estructura obsoleta; es por esto que se considera valioso continuar con el proceso de traducción desde una perspectiva más proactiva, tendiente a modificar, actualizar, mejorar y corregir el código original y los procesos que se hayan podido pasar por alto, así como

|   |                                      |         |            |
|---|--------------------------------------|---------|------------|
|  | INFORME FINAL DE<br>TRABAJO DE GRADO | Código  | FDE 089    |
|   |                                      | Versión | 03         |
|   |                                      | Fecha   | 2015-01-22 |

abordar los demás módulos independientes del software, todo ello con el objetivo de optimizar el rendimiento del software *Ecosonos* y facilitar a otros la implementación de nuevas funciones.

### Trabajo futuro

A pesar de tener unos avances sobresalientes en el desarrollo del software *Ecosonos* en el lenguaje de Python y de haber construido un programa funcional, así como mejorado y ampliado algunos de sus procesos; podría afirmarse que esta versión enfocada en sonotipos y detección cumple con los requisitos, pero el software completo se ha desarrollado abarcando otras áreas y funcionalidades independientes de este módulo por lo que a futuro se puede tomar la decisión de continuar con la traducción de los demás módulos, además de eso teniendo en cuenta las actualizaciones que se le pudieron efectuar mientras se desarrollaba esta traducción y proyecto de grado. Por lo anterior, en la actualidad, existen procesos de adaptación y desarrollo pendientes por realizar fuera de la sección de detección de especies; circunstancia que, seguramente, garantizará un amplio campo de investigación. Además, porque el área de la bioacústica se encuentra aún en desarrollo en Colombia y en otras partes del mundo, descubriéndose constantemente nuevas técnicas y necesidades y, con ellas, probablemente, nuevas solicitudes y funcionalidades.

## REFERENCIAS

---

Martinez Medina, D. et al. (2021) 'Estado, Desarrollo y Tendencias de los Estudios en Acústica de la fauna en Colombia', *Biota Colombiana*, 22(1).  
 doi:10.21068/c2021.v22n01a01.

|   |                                      |         |                |
|---|--------------------------------------|---------|----------------|
|  | INFORME FINAL DE<br>TRABAJO DE GRADO | Código  | FDE 089        |
|   |                                      | Versión | 03             |
|   |                                      | Fecha   | 2015-01-2<br>2 |

Mcloughlin, M.P., Stewart, R. And Mcelligott, A.G. (2019) 'automated Bioacoustics: Methods In Ecology And Conservation And Their Potential For Animal Welfare Monitoring', *Journal Of The Royal Society Interface*, 16(155), P. 20190225. Doi:10.1098/Rsif.2019.0225.

Bradbury, J. W. & Vehrencamp, S. L. (1998). Principles Of Animal Communication. Sinauer, Sunderland, Ma.

Ariza, R.A. (2021) *Sinfonía Llanera: Diversidad Acústica En Un Gradiente De Intervención Humana En Corozal, Yopal*. Tesis. Universidad De Los Andes.

Sueur, J., & Farina, A. (2015). Ecoacoustics: The Ecological Investigation And Interpretation Of Environmental Sound. *Biosemiotics* 8, 493-502.

García , F.G. (2023) *La Bioacústica Y La Ecoacústica Al Servicio De La Valoración De Los Ecosistemas*. Available At:  
<https://www.inecol.mx/inecol/index.php/es/ct-menu-item-25/ct-menu-item-27/17-ciencia-hoy/1891-la-bioacustica-y-la-ecoacustica-al-servicio-de-la-valoracion-de-los-ecosistemas> (Accessed: 21 September 2023).

S. Na, L. Xumin and G. Yong, "Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm," 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, Jian, China, 2010, pp. 63-67, doi: 10.1109/IITSI.2010.74.

Y. R. Pandeya, B. Bhattarai and J. Lee, "Visual Object Detector for Cow Sound Event Detection," in *IEEE Access*, vol. 8, pp. 162625-162633, 2020, doi: 10.1109/ACCESS.2020.3022058.

## APÉNDICE

---

Apéndice A. Código de GitHub con el cual se puede iniciar la interfaz del software *Ecosonos*. Esta es una versión preliminar que se enfoca en la sección de sonotipos y reconocimiento ya que el software completo posee más funcionalidades.

Acceso al repositorio de Github: [https://github.com/JuanGiraldoB/biologos\\_uni](https://github.com/JuanGiraldoB/biologos_uni)

|   |  |         |            |
|---|--|---------|------------|
|  | <b>INFORME FINAL DE TRABAJO DE GRADO</b> | Código  | FDE 089    |
|   |  | Versión | 03         |
|   |  | Fecha   | 2015-01-22 |

FIRMA ESTUDIANTES Manuel Alejandro Ob. B.  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

FIRMA ASESOR Leonardo Duque H.  
 \_\_\_\_\_

FECHA ENTREGA: 20 de Noviembre del 2023

FIRMA COMITÉ TRABAJO DE GRADO DE LA FACULTAD \_\_\_\_\_

RECHAZADO\_\_\_\_      ACEPTADO\_\_\_\_      ACEPTADO CON MODIFICACIONES\_\_\_\_

ACTA NO. \_\_\_\_\_  
 FECHA ENTREGA: \_\_\_\_\_

FIRMA CONSEJO DE FACULTAD \_\_\_\_\_

ACTA NO. \_\_\_\_\_  
 FECHA ENTREGA: \_\_\_\_\_