



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

— **TELECOM** ESCUELA  
TÉCNICA **VLC** SUPERIOR  
DE INGENIERÍA DE  
TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de  
Telecomunicación

Herramienta de validación y etiquetado de cantos de ave

Trabajo Fin de Máster

Máster Universitario en Ingeniería de Telecomunicación

AUTOR/A: Sanz Montrull, Daniel

Tutor/a: Piñero Sipán, María Gemma

CURSO ACADÉMICO: 2022/2023



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

— **TELECOM** ESCUELA  
TÉCNICA **VLC** SUPERIOR  
DE INGENIERÍA DE  
TELECOMUNICACIÓN

Escuela Técnica Superior de Ingeniería de Telecomunicación  
Universitat Politècnica de València  
Edificio 4D. Camino de Vera, s/n, 46022 Valencia  
Tel. +34 96 387 71 90, ext. 77190  
[www.etsit.upv.es](http://www.etsit.upv.es)

**VLC/**  
**CAMPUS**  
VALENCIA, INTERNATIONAL  
CAMPUS OF EXCELLENCE



## Resumen

Este TFM se enmarca en el proyecto de investigación DAPHNE (*Deep-learning Analysis and cyber-PHysical systems applied to biodiversity in urban and Natural Environments*) cuyo objetivo es determinar el grado de biodiversidad en entornos naturales y urbanos. En el marco del proyecto DAPHNE se ha llevado a cabo el despliegue de diez nodos acústicos en el *Parc Natural de l'Albufera* que ha permitido adquirir una gran cantidad de señales sonoras.

El objetivo del presente TFM es desarrollar una herramienta de validación y etiquetado de cantos de ave que pueda ser usada por expertos en ornitología. Esta herramienta será una aplicación web que permitirá al experto escuchar el canto del ave y visualizar el espectrograma Mel. A partir de dicha información, la persona experta etiquetará el sonido con la especie y el tipo de canto. Además, se automatizará la obtención de un primer etiquetado de los audios a partir de la aplicación *BirdNET*, para posteriormente segmentar dichos audios y presentarlos en la herramienta web.

## Resum

Aquest TFM va emmarcat en el projecte d'investigació DAPHNE (*Deep-learning Analysis and cyber-PHysical systems applied to biodiversity in urban and Natural Environments*) que té com objectiu determinar el grau de biodiversitat en entorns naturals i urbans. En el marc del projecte DAPHNE s'ha portat a terme el desplegament de deu nodes acústics en el Parc Natural de l'Albufera que ha permès adquirir una gran quantitat de senyals sonores.

L'objectiu d'aquest TFM és desenvolupar una ferramenta de validació i etiquetat de cants d'aus que serà emprada per experts en ornitologia. Aquesta ferramenta serà una aplicació web que permetrà a l'expert escoltar el cant de l'au i visualitzar l'espectrograma Mel. A partir d'aquesta informació, la persona etiquetarà el so amb l'espècie i el tipus de cant. A més a més, s'automatitzarà l'obtenció d'un primer etiquetat dels àudios a partir de l'aplicació *BirdNET*, per a posteriorment segmentar aquests àudios y presentar-los en la ferramenta web.

## Abstract

This Master's thesis is part of the research project DAPHNE (*Deep-learning Analysis and cyber-PHysical systems applied to biodiversity in urban and Natural Environments*), aimed at determining the level of biodiversity in natural and urban environments. Within the framework of the DAPHNE project, the deployment of ten acoustic nodes has been carried out in *Parc Natural de l'Albufera*, enabling the acquisition of a considerable amount of sound signals.

The objective of this Master's thesis is to develop a validation and labeling tool for bird songs that can be utilized by ornithology experts. This tool will be a web application allowing the expert to listen to the bird song and visualize the Mel spectrogram. Based on this information, the expert will label the sound with the species and type of song. Furthermore, the initial labeling of the audio will be automated using the *BirdNET* application, followed by the segmentation of these audio segments for presentation within the web tool.



## Índice

1.	Introducción .....	3
1.1	Parque natural de la Albufera.....	3
1.2	Motivación .....	3
1.3	Objetivos .....	4
2.	Marco teórico .....	5
2.1	Aprendizaje Automático ( <i>Machine Learning</i> ) y sus ramas.....	5
2.1.1	Aprendizaje profundo (Deep Learning) .....	5
2.1.2	Redes Neuronales Artificiales (Artificial Neural Networks) .....	6
2.1.3	Redes Neuronales Profundas (Deep Neural Networks) .....	6
2.2	Tratamiento del audio.....	7
2.2.1	Segmentación .....	7
2.2.2	Espectrograma.....	8
2.2.3	Escala Mel.....	9
2.3	Base de datos relacional .....	10
3.	Entorno de trabajo .....	11
3.1	<i>eBird</i> .....	11
3.2	<i>BirdNET</i> .....	11
3.3	PHP .....	11
3.4	Conjunto de datos.....	12
3.5	Nodos de grabación.....	13
3.6	Librosa.....	14
3.7	Subsistema de Windows para Linux (WSL) .....	15
4.	Metodología .....	16
4.1	Especificaciones .....	16
4.2	Base de datos.....	17
4.2.1	Elección del modelo de base de datos .....	17
4.2.2	Diseño de la base de datos.....	18
4.2.3	Resultado final.....	20
4.3	Segmentación de las grabaciones .....	21
4.3.1	Segmentado .....	21
4.3.2	Duración del segmento.....	21
4.3.3	Script para segmentar las grabaciones.....	22



4.4	Definición de la herramienta .....	22
4.5	Estudio y aplicación de <i>BirdNET</i> .....	24
4.5.1	Instalación de BirdNET y primera ejecución .....	24
4.5.2	Umbral de confianza de BirdNET .....	24
4.5.3	Validación de los resultados de BirdNET .....	25
4.5.4	Lista de especies de BirdNET .....	27
5.	Desarrollo .....	29
5.1	Procesado de las grabaciones .....	29
5.2	Desarrollo de la web en local .....	32
5.2.1	Inicio de sesión .....	33
5.2.2	Página principal .....	34
5.2.3	Historial .....	37
5.2.4	Edición de registro .....	38
5.2.5	Matriz de confusión .....	38
5.2.6	Filtrado de segmentos .....	39
5.2.7	Segmentos por especie .....	41
5.2.8	Registro de nuevos usuarios (Administrador) .....	42
5.2.9	Top confianza BirdNET (Administrador) .....	42
5.3	Despliegue de la web .....	43
6.	Conclusiones .....	44
7.	Líneas futuras .....	45
8.	Objetivos de Desarrollo Sostenible .....	47
	Bibliografía .....	48
	Anexo .....	51
A.	Lista de especies consideradas por <i>BirdNET</i> .....	51

## Índice de figuras

Figura 2.1 Representación simplificada de una neurona artificial. [9].....	6
Figura 2.2 A la izquierda se observa una ANN y a la derecha se observa una DNN. Fuente: Elaboración propia. ....	7
Figura 2.3 Espectrograma del canto de un ejemplar de carricero tordal. Fuente: Elaboración propia.....	9
Figura 2.4 Representación de la relación entre Hz y la escala Mel elaborada con la fórmula que aparece en [16]. ....	10
Figura 2.5 Espectrograma en Escala Mel del canto de un ejemplar de carricero tordal. Fuente: Elaboración propia. ....	10
Figura 3.1 Nodos de grabación distribuidos en el <i>Parc Natural de l'Albufera</i> [22].....	13
Figura 3.2 Distribución de los nodos de grabación en el <i>Parc Natural l'Albufera</i> . ....	14
Figura 4.1 Primera propuesta de diseño de la herramienta. ....	17
Figura 4.2 Descripción del nombre de archivo de un segmento. ....	19
Figura 4.3 Diseño final de la base de datos. ....	20
Figura 4.4 Aspecto del entorno Ubuntu sobre WSL. ....	24
Figura 4.5 Ubicación de las coordenadas 39°20'06.2"N 0°21'09.0"W ..... 28	28
Figura 5.1 Espectrograma en escala Mel del segmento "SMA07394_20230520_110002_0_3.wav" .....	30
Figura 5.2 Descripción del procesado que se realiza en local .....	31
Figura 5.3 Captura del panel de control de XAMPP con los módulos Apache y MySQL en marcha .....	32
Figura 5.4 Aspecto de <i>login.php</i> .....	34
Figura 5.5 Aspecto de <i>home.php</i> .....	34
Figura 5.6 Aspecto tras pulsar el botón “Mostrar resultados de BirdNET”.....	36
Figura 5.7 En la parte izquierda: Menú usuario normal. En la parte derecha: Menú usuario administrador.....	36
Figura 5.8 Aspecto de <i>historial.php</i> .....	37
Figura 5.9 Aspecto de <i>editar_registro.php</i> .....	38
Figura 5.10 Aspecto de <i>confusion_matrix.php</i> .....	39
Figura 5.11 Aspecto de <i>filtro_segmentos.php</i> .....	40
Figura 5.12 Aspecto de <i>filtro_segmentos2.php</i> .....	41
Figura 5.13 Aspecto de <i>segmentos_por_especie.php</i> .....	41
Figura 5.14 Aspecto de <i>registration.php</i> .....	42
Figura 5.15 Aspecto de <i>register.php</i> .....	42
Figura 5.16 Aspecto de <i>top_confidence.php</i> .....	43



## Índice de tablas

Tabla 3.1 Características de los nodos de grabación [22]. .....	13
Tabla 3.2 Ubicación y configuración de los nodos de grabación durante los periodos de primavera y verano. ....	14
Tabla 4.1 Resumen de las características de los segmentos. ....	22
Tabla 4.2 Resumen de las características de la herramienta. ....	23
Tabla 4.3 Ejemplo de etiquetas utilizando umbrales de confianza distintos, .....	25
Tabla 4.4 Extracto de las etiquetas generadas por los ornitólogos utilizando Excel.....	26

## 1. Introducción

### 1.1 Parque natural de la Albufera

El *Parc Natural de l'Albufera* se encuentra ubicado al sur de la ciudad de Valencia, a unos 10 km de distancia. Este parque natural abarca una superficie de 21.120 hectáreas y es uno de los humedales más emblemáticos y representativos de la Comunidad Valenciana y de la región mediterránea. Este humedal es el hábitat de una gran variedad de especies, siendo las aves uno de los principales atractivos de este parque [1]. En *l'Albufera* se pueden encontrar unas 300 especies de aves en lo que dura un año, siendo algunas de estas especies protegidas, por ejemplo, la cerceta pardilla (*Marmaronetta angustirostris*) o el tarro blanco (*Tardona tardona*) [2]. Dentro de las especies de aves que se pueden encontrar en *l'Albufera*, hay especies que habitan en el humedal todo el año y hay otras especies que sólo pueden observarse en invierno, en verano o en períodos de migración [3]. Algunas de las especies más comunes son el pato colorado (*Netta rufina*), el avetorillo común (*Ixobrychus minutus*), el calamón común (*Porphyrio porphyrio*), la focha común (*Fulica atra*) y la garza real (*Ardea cinerea*). Entre las especies más raras de avistar se pueden encontrar el avetoro común (*Botaurus stellaris*), la cerceta pardilla (*Marmaronetta angustirostris*) o el aguilucho lagunero (*Circus aeruginosus*).

### 1.2 Motivación

Para tener un conocimiento del estado en el que se encuentra un ecosistema, es una buena práctica monitorizar la población y diversidad de especies que habitan en este. Las aves son un buen indicador que se utiliza para controlar el estado del ecosistema, ya que, viven en muchos entornos diferentes. Normalmente se monitoriza la biodiversidad de las aves a través del conteo de ejemplares. En este método, los expertos visualizan, identifican y cuentan ejemplares de ave que aparecen en una zona durante un periodo de tiempo. Este método es sensible a diversos factores que afectan a la detección, como pueden ser, por ejemplo, factores climáticos. Asimismo, este método resulta costoso si se quieren cubrir áreas grandes. Por otro lado, se tiene que el uso de un sistema de monitorización acústica pasiva, donde se utilizan nodos de grabación distribuidos en una zona, permite monitorizar las especies que habitan en un área durante largos periodos de tiempo con un coste notablemente inferior [4].

Debido a la gran biodiversidad de especies de aves que existe dentro del *Parc Natural de l'Albufera*, es interesante el uso e implantación de diferentes tecnologías que permitan optimizar el estudio y el control de las especies que habitan en el humedal, pudiendo llevar una monitorización de las especies que habitan dentro del parque en diferentes momentos del año. Para realizar esta tarea, se han distribuido 10 nodos de grabación en diferentes puntos estratégicos del parque. Finalmente, dentro de las grabaciones que realicen estos nodos, será importante identificar correctamente las especies que aparezcan. Para ello será necesario etiquetar en momentos concretos de la grabación qué especies se escuchan.

Finalmente, la creación de un conjunto de datos (en inglés, *dataset*) etiquetado de audios es un proceso que ocupa mucho tiempo y puede resultar tedioso tanto para un usuario que no haya tenido una experiencia previa con esta tarea como para usuarios que ya tengan experiencia. Además, las etiquetas que se deben utilizar en un proceso de etiquetado deben de ser etiquetas que ya vengan con nombres preestablecidos para evitar confusiones y para tener los datos etiquetados ordenados. La creación de una herramienta que sea amigable y sencilla de utilizar para el usuario permitirá optimizar en tiempo el proceso de etiquetado de segmentos de audios [5]. El tiempo se optimiza porque se reduce la carga de trabajo que recaería sobre el usuario a la hora de identificar segmentos de audio y anotarlos manualmente en un archivo aparte. A su vez, estas etiquetas tendrán nombres ya preestablecidos independientemente del usuario que realice la clasificación, dando un orden al conjunto de datos obtenido.



### 1.3 Objetivos

Teniendo en cuenta lo expuesto en la sección 1.2, el objetivo principal de este trabajo es el diseño e implementación de una herramienta que facilite el proceso de determinar a qué especies de ave corresponden los cantos que se escuchen en una grabación en diferentes instantes de tiempo. Este proceso se define como **etiquetado**. Como los archivos de audio tienen una duración de 1 hora cada uno, será necesario dividir este archivo en diferentes segmentos para facilitar así su tratamiento y procesado. Este proceso se define como **segmentación**. De esta manera se tiene que la herramienta o sistema a implementar deberá de ser capaz de segmentar una grabación de 1 hora pudiendo omitir segmentos de audio que no aporten información al conjunto de datos y que después estos segmentos puedan ser etiquetados por un usuario final, que en este caso será uno o más ornitólogos.

Esta herramienta deberá de poder ser utilizada desde cualquier ordenador, no debe de requerir de un gran esfuerzo por parte del usuario para poder hacer uso de ella, y debe ser escalable, lo que quiere decir, que pueda ser funcional independientemente del tamaño que adquiera el conjunto de datos. Para ello, habrá que ver qué implementación será la más adecuada para lograr estos requerimientos.

Tal y como se verá en el desarrollo de este trabajo, esta herramienta se apoyará en el uso de un algoritmo de aprendizaje profundo (en inglés, *deep learning*) ya implementado por el *Cornell Lab of Ornithology* llamado *BirdNET* [4]. Este modelo entrenado es capaz de distinguir qué especies de ave se escuchan dentro de una grabación. Pudiendo estudiar la precisión que tiene este modelo clasificando muestras en el ámbito del *Parc Natural de l'Albufera* y poder así validar este algoritmo.

## 2. Marco teórico

### 2.1 Aprendizaje Automático (*Machine Learning*) y sus ramas

El aprendizaje automático se define como una rama de la Inteligencia Artificial donde los sistemas que son capaces de aprender de los datos son estudiados y construidos [6]. Arthur L. Samuel lo definió como el “campo de estudio que les da a los ordenadores la habilidad de aprender sin ser explícitamente programados”. El aprendizaje automático se utiliza en situaciones donde la implementación manual de un algoritmo supone un coste elevado y resulta en una tarea tediosa. En el ejemplo que ocupa este trabajo, el reconocimiento de cantos de ave no es algo que, a priori, se pueda programar manualmente o se le pueda especificar a una máquina en tan bajo nivel. Por lo tanto, en esta situación, es la máquina la que, siguiendo unas pautas específicas, será capaz de encontrar la manera de distinguir el canto de diferentes especies de pájaro.

El aprendizaje automático puede ser dividido en 3 áreas distintas:

- Aprendizaje supervisado: Se diseña un modelo que sea capaz de relacionar una entrada con una salida. Para ello se enseña a la máquina una serie de entradas con sus correspondientes salidas (conjunto de entrenamiento) y después se evalúa cómo se comporta el sistema ante entradas que no ha “visto” anteriormente. La variedad en los datos de entrenamiento del modelo es importante, ya que, éste debe de ser capaz de generalizar adecuadamente las muestras que no se han utilizado en este entrenamiento.
- Aprendizaje no supervisado: En este caso, al algoritmo no se le especifican las salidas deseadas dados unos datos de entrada. El objetivo del algoritmo es el de encontrar patrones dentro de los datos o descubrir cómo están estructurados. Se suele utilizar para agrupar objetos que tengan características similares.
- Aprendizaje por refuerzo: En este caso, el sistema se trata de un agente que, dentro de un entorno, tratará de cumplir un objetivo. Para ello se define una función de recompensa que deberá de ser maximizada por el agente. Un ejemplo para este caso sería el de un robot que tiene que salir de un laberinto

La herramienta que se ha implementado en este trabajo servirá para crear conjuntos de datos etiquetados, que en relación con lo que se comenta anteriormente, podrá ser utilizado por un modelo de aprendizaje supervisado. Este modelo interpretará los archivos de audio como entrada y las etiquetas de la especie de ave como salida.

Por otro lado, el caso de *BirdNET* entraría dentro de un sistema de aprendizaje supervisado donde se le han especificado unas entradas (grabaciones de audio) y unas salidas (especie o especies que se escuchan en esas grabaciones). De esta manera, el conjunto de datos que crearemos con la herramienta servirá para validar cómo de bueno es *BirdNET* distinguiendo especies de ave del *Parc Natural de l'Albufera* comparando la salida que dará *BirdNET* con la etiqueta o etiquetas que dieron los ornitólogos a ese archivo.

#### 2.1.1 Aprendizaje profundo (*Deep Learning*)

El aprendizaje profundo (en inglés, *Deep Learning* y comúnmente abreviado como DL) se define como una parte o rama del aprendizaje automático basado en redes neuronales artificiales (en inglés, *Artificial Neural Networks* y abreviado como ANN) que trata de encontrar una representación de los datos de entrada (sin procesar) que lleven como resultado una salida deseada. Un sistema de aprendizaje profundo está compuesto por varias capas de neuronas que realizan transformaciones no lineales en los datos de entrada. Dicho de otra manera, se pasa de una representación de más alto nivel a una representación más abstracta. En tareas de clasificación, que es lo que corresponde a este trabajo, las capas se encargan de amplificar aspectos de los datos de entrada que son importantes para discriminar entre las diferentes salidas posibles y eliminar las variaciones que son más irrelevantes. La idea principal del aprendizaje

automático es que estas capas no son diseñadas por los desarrolladores, sino que estas capas se van auto-diseñando a través del proceso de aprendizaje [7].

### 2.1.2 Redes Neuronales Artificiales (Artificial Neural Networks)

Las redes neuronales artificiales (ANN) son algoritmos englobados dentro de la Inteligencia Artificial. Estos algoritmos suelen ser utilizados en el reconocimiento de patrones y la realización de toma de decisiones [8]. Las ANN se caracterizan por dos principios:

- Están formadas por un conjunto de neuronas artificiales que están interconectadas. Estas conexiones entre neuronas artificiales se llaman enlaces. Esta característica hace que se comporte de manera similar a un cerebro humano.
- Normalmente no hace ninguna presunción sobre la distribución de los datos antes del proceso de entrenamiento del modelo.

Las neuronas artificiales reciben varias señales de entrada a las que le aplican una función de activación (como podrían ser la función sigmoide o la tangente hiperbólica) a la suma de todas las entradas que recibe. Esta señal la envía a la salida y pasa al enlace con la siguiente neurona. En el enlace entre neuronas se puede aumentar o disminuir la intensidad con la que se transmite la señal. De esta manera, las conexiones entre éstas tienen unos valores, conocidos como pesos, que se ajustan en el proceso de aprendizaje que realiza la red neuronal [9]. A continuación, en la Figura 2.1 se muestra una representación de una neurona artificial, donde la salida de la neurona viene dada por  $a_j$ , que es resultado de aplicar la función de activación,  $g(x)$ , a la suma de cada una de las entradas,  $a_i$ , multiplicadas por su peso correspondiente,  $w_{i,j}$ :

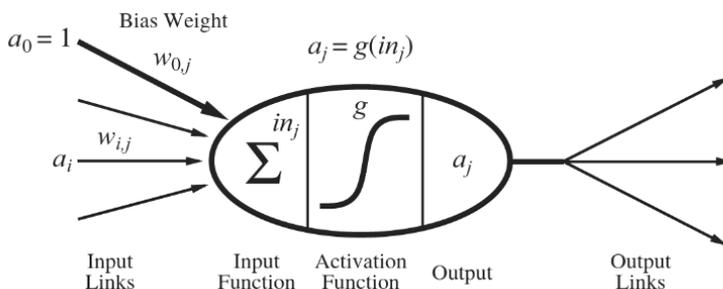


Figura 2.1 Representación simplificada de una neurona artificial. [9]

De esta manera, teniendo en cuenta representación de la Figura 2.1, la salida de una neurona artificial viene determinada por (2.1) [9].

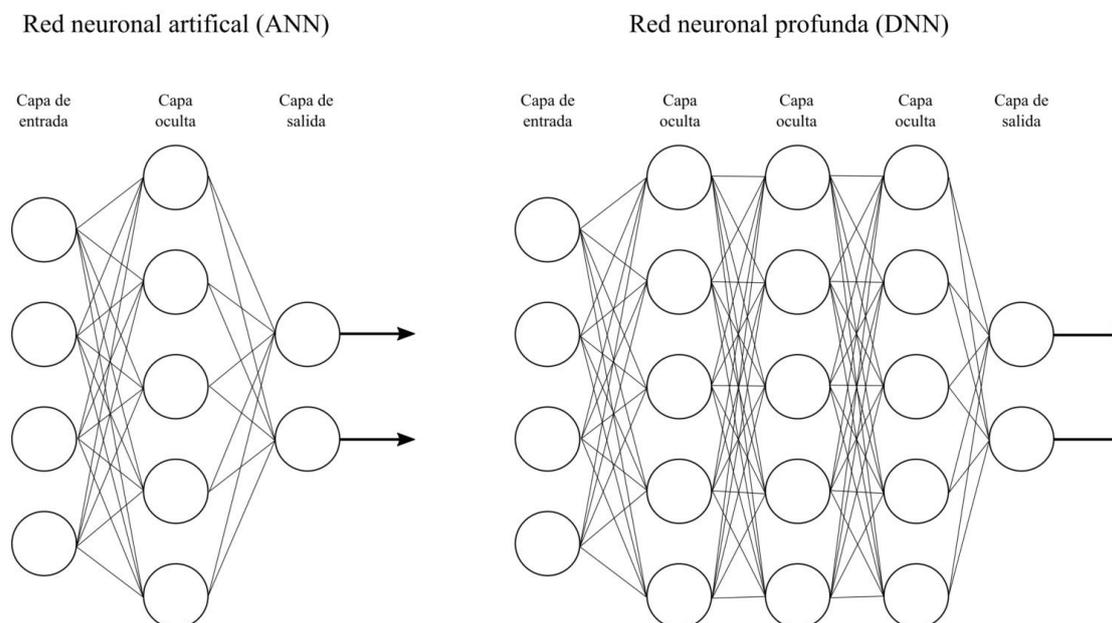
$$a_j = g_j \left( \sum_i w_{i,j} a_i \right) \quad 2.1$$

Las neuronas van agrupadas dentro de capas donde, normalmente, en cada capa se realizan diferentes transformaciones a los datos que reciben de la capa anterior. La primera capa de la red neuronal es la capa de entrada, donde se introduce la señal de entrada. Luego, la salida de la capa de entrada puede atravesar una o múltiples capas, donde se efectúan las transformaciones. Estas capas intermedias se conocen como capas ocultas. La última capa de la red neuronal, que es la que proporciona la salida, se conoce como capa de salida [9].

### 2.1.3 Redes Neuronales Profundas (Deep Neural Networks)

Las redes neuronales profundas (en inglés *Deep Neural Networks*, DNN) se definen como redes neuronales artificiales que cuentan con múltiples capas entre la capa de entrada y la capa de salida. Una DNN sería lo mismo que una ANN multicapa. La red neuronal profunda trata de materializar, en forma de algoritmos, el comportamiento del cerebro humano cuando trata conjuntos de datos muy grandes durante el proceso de aprendizaje por parte de la red. Dentro de la arquitectura de las DNN podemos encontrar un número grande de capas que pueden llegar a ser muy profundas.

En algunos casos se podría hablar de miles de capas y millones de neuronas. Cada capa está formada por neuronas que, igual que ocurre con las ANN, su función viene modelada por la función de activación [10]. En la Figura 2.2, se muestran un ejemplo de una ANN con una capa oculta y un ejemplo de una DNN con 3 capas ocultas:



**Figura 2.2** A la izquierda se observa una ANN y a la derecha se observa una DNN. Fuente: Elaboración propia.

## 2.2 Tratamiento del audio

### 2.2.1 Segmentación

Un archivo de audio puede ser dividido en otras unidades semánticas más pequeñas y a su vez, estas unidades pueden volver a ser divididas en unidades más pequeñas, pudiendo establecer una jerarquía entre los diferentes niveles de división. Por ejemplo, de un archivo de 1 hora se pueden obtener 2 archivos de media hora, dentro de cada uno de esos archivos de media hora se pueden obtener otros archivos de 5 minutos y así hasta llegar hasta unidades cada vez más pequeñas. Estas unidades más diminutas pueden tener duraciones del orden de milisegundos. Según se explica en [11], se puede establecer una jerarquía de 4 niveles que se presentará a continuación, ordenando de la unidad más grande a la más pequeña:

1. **Grabación:** Es la unidad de más alto nivel donde está contenido el mayor contenido semántico. Dentro de este nivel se tendrían las grabaciones originales donde no se les ha aplicado ningún tipo de procesado.
2. **Toma:** Es una unidad dentro de la grabación que contiene una misma fuente de sonido. En el caso de este trabajo, sería un trozo de duración variable en la que está contenido el canto de una misma especie.
3. **Segmento:** Es un fragmento de audio que tiene una duración máxima fijada. Esta duración suele ser del orden de segundos. El proceso de clasificación que realizan los modelos de aprendizaje automático se hace sobre los segmentos de audio [11]. En este proceso, el modelo trata de predecir a qué clase de las ya definidas dentro de éste pertenece el segmento.
4. **Trama:** Es la unidad de audio más pequeña que se establece en esta jerarquía cuya duración suele estar comprendida entre los 20 ms y 30ms.

Tal y como se observa, por definición, tanto la grabación como la toma tienen duraciones indeterminadas, mientras que, por el contrario, el segmento y la trama tienen una duración fija. Aplicando esta jerarquía a este proyecto, se hará uso de las grabaciones y de los segmentos, siendo los niveles 1 y 3 de la jerarquía. Los segmentos van relacionados a una grabación original y estos mismos llevan una indicación en la que se especifica a qué momento exacto de la grabación corresponden. A su vez, las grabaciones llevarán, en el nombre del archivo, un indicador de a qué día y hora se hizo, y en qué nodo se tomó la grabación. Esta información irá implícita tanto en el nombre de las grabaciones como de los segmentos. Esto puede resultar útil para visitar la grabación original en caso de encontrar algún evento que sea lo suficientemente relevante.

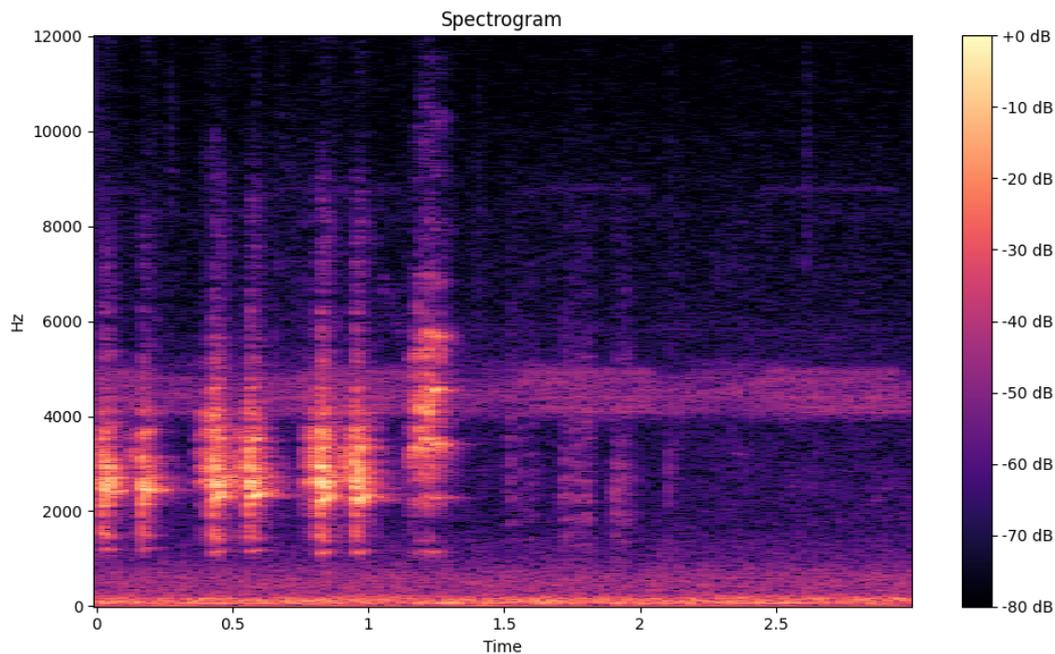
### 2.2.2 Espectrograma

El espectrograma es una representación de la intensidad de una señal no estacionaria en el dominio de la frecuencia y del tiempo. Esta representación en frecuencia de la señal se suele hacer en dos dimensiones en forma de mapa de calor. El eje horizontal suele representar el tiempo, el eje vertical suele representar la frecuencia y el color suele representar la amplitud, normalmente en escala logarítmica (en decibelios). Para realizar esta representación se suele utilizar la Transformada de Fourier de Tiempo Reducido (en inglés *short-time Fourier transform*, STFT) [12].

La representación en frecuencia de una señal se obtiene a partir de la transformada discreta de Fourier (en inglés *Discrete Fourier Transform*, DFT), también conocida como *Fast Fourier Transform* (FFT). De esta manera, se tiene que la STFT es una secuencia de FFTs de segmentos de datos (en este caso de audio) en ventanas, donde normalmente estas ventanas tienen solapes en tiempo del 25% o del 50% [12]. La STFT descompone la señal en fragmentos que se solapan utilizando una ventana que calcula la FFT en cada uno de estos fragmentos, también llamados tramas [13].

La STFT puede ser representada como una matriz de coeficientes de la FFT en cada una de las tramas. En esta matriz las columnas se asocian al tiempo y las filas a la frecuencia. La obtención de la magnitud de cada uno de los coeficientes resulta en una matriz que puede ser tratada como una imagen. Esta imagen resultante se conoce como el espectrograma de la señal [13].

A continuación, en la Figura 2.3, se muestra un espectrograma del canto de un espécimen de carricero tordal (*Acrocephalus arundinaceus*) grabado el 27 de mayo de 2022 en torno a las 8:42 am. El espectrograma se ha generado con Python, utilizando la librería de audio *librosa* [14]. Hay que tener en cuenta que sólo aparece hasta 12 kHz porque la frecuencia de muestreo de los micrófonos que han grabado este segmento está a 24 kHz. La intensidad se ha representado en escala logarítmica tomando como referencia el máximo nivel de señal.

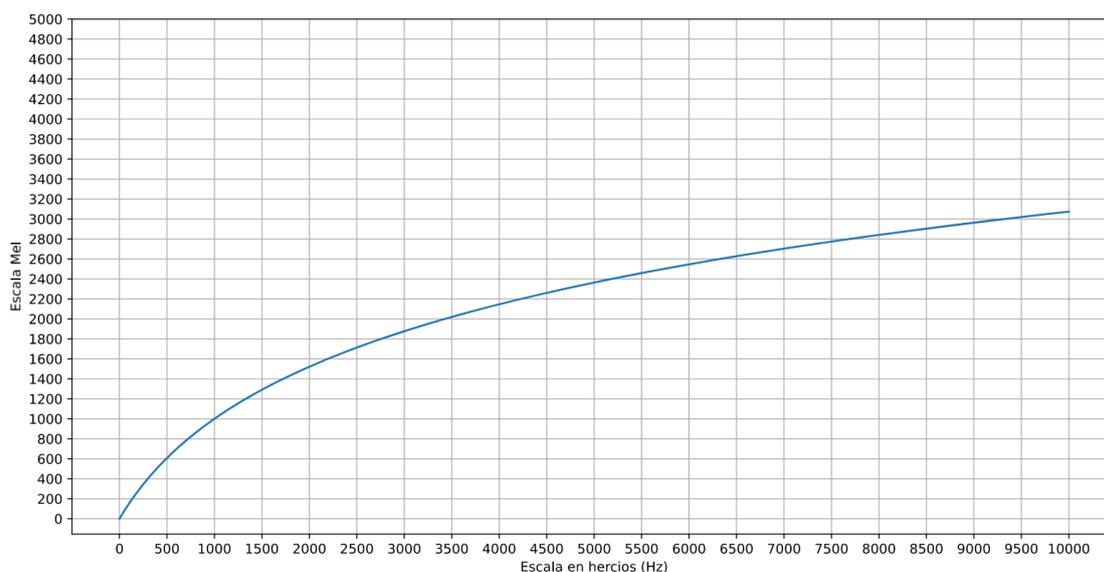


**Figura 2.3** Espectrograma del canto de un ejemplar de carricero tordal. Fuente: Elaboración propia.

### 2.2.3 Escala Mel

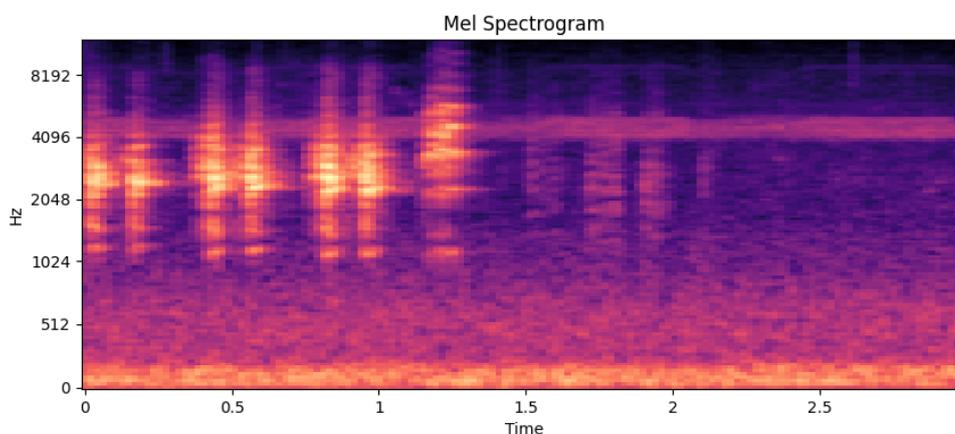
La Escala Mel es una escala de frecuencias donde, desde el punto de vista humano, los tonos dentro de la escala son percibidos como equidistantes. Desde la percepción del oído del ser humano, las frecuencias bajas son más fáciles de distinguir entre ellas que las frecuencias altas. Cuanta más alta sea la frecuencia, mayor tiene que ser la distancia en hercios entre 2 tonos para que sean distinguidos como equidistantes en comparación a una pareja de 2 tonos ubicados en una frecuencia menor [15]. En la Figura 2.4 se observa una gráfica donde se muestra la relación entre la escala en hercios (Hz) y la Escala Mel. En [16] se puede encontrar la fórmula que relaciona la frecuencia en hercios con valores de la Escala Mel (mels), siendo  $f$  la frecuencia en hercios y  $m$  el valor en mels. Esta fórmula es la que se muestra a continuación:

$$m = 2595 \log_{10} \left( 1 + \frac{f}{700} \right) \quad 2.2$$



**Figura 2.4** Representación de la relación entre Hz y la escala Mel elaborada con la fórmula que aparece en [16].

En la Figura 2.5 se muestra cómo quedaría el espectrograma que aparece en la Figura 2.3 aplicando la Escala Mel en el rango de frecuencias. Se aprecia que la señal queda más comprimida en los rangos de frecuencias más agudas mientras que se ve más expandida en el rango de frecuencias más grave. Dicho de otra manera, correspondería a la visualización de cómo percibe el oído humano la distancia entre frecuencias.



**Figura 2.5** Espectrograma en Escala Mel del canto de un ejemplar de carricero tordal. Fuente: Elaboración propia.

### 2.3 Base de datos relacional

La base de datos relacional es una categoría de base de datos que satisface las características correspondientes al modelo relacional de los datos. Este modelo se basa principalmente en que se puedan establecer interconexiones, también llamadas relaciones, entre los datos, que van guardados en tablas. Una tabla dentro de una base de datos relacional está compuesta por registros, también llamados filas, donde se almacenan los datos [17].

Aplicándolo a este trabajo, se puede tener una tabla para almacenar los datos de los usuarios, otra tabla para almacenar los datos de las especies, otra para almacenar los datos relacionados a las grabaciones y otra para almacenar tanto los resultados de las clasificaciones de *BirdNET* como de los ornitólogos. También será posible obtener, mediante relaciones, la información que se desee de la base de datos, por ejemplo, cuántas especies encontró un ornitólogo, en qué archivos de audio las encontró y el nombre de estas especies.

### 3. Entorno de trabajo

En este capítulo de la memoria se van a describir y detallar las herramientas tecnológicas que se han utilizado en este proyecto. Esto servirá como punto de partida para empezar a desarrollar el proyecto. Se explicará la DNN que se ha utilizado para identificar cantos de ave dentro de las grabaciones, el conjunto de datos que se utilizará en este proyecto y cómo se ha generado, el lenguaje web del que se basa la herramienta implementada, la librería de Python utilizada para procesar las grabaciones y el entorno que se utiliza para ejecutar algoritmos de procesado.

#### 3.1 *eBird*

El objetivo principal de la plataforma *eBird* consiste en recopilar datos de aves a nivel global. Se recopilan datos de localización a nivel diario de especies de aves. Estos datos pueden ser analizados a nivel de continente, país o región de un país. Se pueden aportar observaciones, subir fotos y grabaciones. El *Cornell Lab of Ornithology* es el que se encarga de esta plataforma [18].

Esta plataforma permite que sus usuarios puedan realizar un análisis de sus observaciones, monitorizar la cantidad de especies que se observan en distintos puntos geográficos en distintos instantes del año, subir ficheros de imágenes y grabaciones, y la exportación de los datos a un fichero.

#### 3.2 *BirdNET*

*BirdNET* es una herramienta multiplataforma basada en Aprendizaje Automático que se centra en clasificar y detectar sonidos de aves determinando su especie. Dicho de otra manera, se centra en reconocer especies de aves a través del sonido. Esta herramienta puede ser utilizada en teléfonos móviles inteligentes, en ordenadores (en navegador), y en otros dispositivos como Arduino o Raspberry Pi. Es una herramienta que se encuentra en mejora continua. Hoy en día *BirdNET* es capaz de detectar unas 3000 especies de ave del mundo y tienen planeado añadir más especies en el futuro [19].

Entrando más en detalle, *BirdNET* es una DNN capaz de identificar diferentes especies de pájaros a través del sonido de sus cantos. Concretamente, el modelo de red que utiliza se basa en las redes residuales (también conocidas como *ResNets*). El primer modelo que propusieron utilizaba 157 capas con más de 27 millones de parámetros. El entrenamiento de estos modelos lo han realizado usando *Data augmentation* y *mixup*, siendo ambos procesos muy utilizados para el aumentado de datos de entrenamiento. Ambos procesos son muy útiles para después construir modelos que sean robustos a la presencia de fuertes fuentes de ruido en el ambiente. El modelo ha sido probado con distintos conjuntos de datos provenientes de distintas fuentes. La precisión media que obtuvo *BirdNET* en su primer modelo fue del 79,1% en grabaciones donde hay una sola especie [4].

El uso de *BirdNET* en este trabajo sirve para el procesado de las grabaciones que se han realizado en el *Parc Natural de l'Albufera*, obteniéndose un primer etiquetado de la mayoría de los segmentos sonoros de una grabación. Esta herramienta puede ser utilizada en un equipo local, para ello hay que descargar el repositorio que se encuentra en *GitHub* [20]. Dentro de este repositorio se puede encontrar el código que conforma la herramienta, teniendo, además, la posibilidad de ejecutarlo en distintas plataformas. Como se ha mencionado en el párrafo anterior, es una herramienta que se va actualizando continuamente, por esto, hay que mantener el modelo actualizado a la última versión. A lo largo de la realización de este TFM, la herramienta se tuvo que actualizar de la versión 2.3 a la versión 2.4 que fue publicada recientemente.

#### 3.3 PHP

También conocido como *Hypertext Preprocessor*, el lenguaje de programación PHP es un lenguaje de código abierto que se diseñó para desarrollar páginas web. Su uso está extendido para la creación de páginas web con funciones dinámicas e interactivas. El contenido de la web puede

depender del contenido de una base de datos o según como interactúe el usuario con ésta. PHP se ejecuta en el lado del servidor, esto significa que las instrucciones que se hayan escrito se ejecutan en el lado del servidor antes de que se envíe la página web al usuario.

PHP se considera una excelente herramienta para manipular, almacenar u obtener información de una base de datos. Las aplicaciones más comunes para webs que utilicen bases de datos con PHP son las siguientes: registrar y autenticar usuarios, gestionar el contenido, elaborar formularios interactivos, entre otras [21].

En el proyecto se han utilizado las funcionalidades que permiten acceder a la base de datos, registrar usuarios, autenticación de usuarios, elaborar formularios interactivos y diseñar la web de forma dinámica.

### 3.4 Conjunto de datos

El conjunto de datos que se encuentra en este proyecto consiste en una batería de grabaciones realizadas por una serie de nodos distribuidos a lo largo del *Parc Natural de l'Albufera*. Estas grabaciones tienen 1 hora de duración, se utiliza una frecuencia de muestreo de 24000 Hz y se almacenan en un formato sin compresión, como es en este caso, el formato WAV. El rango de frecuencias de la mayoría de los cantos de pájaros va comprendido de los 250 Hz a los 8300 Hz [4]. Así pues, con una frecuencia de muestreo de 24 kHz es suficiente para cubrir estos rangos.

Los rangos de horas que se eligieron para realizar las grabaciones que conformarán el conjunto de datos comprenden el amanecer y el atardecer. Se eligen estos rangos de horas por el hecho de que son los rangos donde se pueden grabar más cantos de pájaros con poca interferencia de los ruidos de la actividad humana (voces, motores, etc.). Las grabaciones extraídas de los nodos hasta la fecha llegan a superar los 4TB en tamaño. Estos registros datan del mes de mayo del año 2022 hasta el mes de junio del año 2023. Para recopilar las grabaciones de los nodos, se realiza un desplazamiento a las localizaciones donde éstos están instalados cuando se quedan sin pilas, se reemplazan las pilas y las tarjetas de memoria SD. Luego se extraen los archivos que ha grabado desde la última vez que se pusieron en marcha.

La información que se pretende extraer de este conjunto de datos son segmentos de audio en los cuales se tendrá una etiqueta o más de una etiqueta que corresponderá a la especie o especies de ave que se escuchan durante ese instante de tiempo. La división de los audios en segmentos es necesaria dado que se escuchan diferentes especies en diferentes rangos de tiempo, así como, también dentro de las grabaciones de 1 hora, hay muchos instantes en los que no se escucha ningún ave. Almacenar esos trozos de audio donde no se escucha ningún canto pueden resultar improductivos, ya que, no aportan información relacionada con el objetivo de este conjunto de datos. Por tanto, se podría adelantar que los segmentos que se almacenarán para el posterior etiquetado por parte de los ornitólogos serán aquellos en los que *BirdNET* ha encontrado una especie de ave y descartar los segmentos en los que no ha encontrado ninguna especie. De esta manera, se optimiza la tarea de los ornitólogos a la hora de clasificar los segmentos, ya que, no tienen que invertir tiempo escuchando muestras en las que no se escucha ningún canto de pájaro.

La segmentación también ayuda a optimizar la memoria que ocupa el conjunto de datos, ya que, se está eliminando los segmentos de audio que no contienen información relevante a este proyecto, en este caso, los segmentos que no contienen cantos de aves. Para evaluar el impacto que podría tener esto de manera aproximada, se ha tomado una muestra de 9 archivos WAV de 1 hora de grabación, se han segmentado conservando sólo los segmentos en los que se ha detectado alguna especie de ave. Estos segmentos ocupan el 43,74% de lo que ocupaban los archivos originales, pasando de ocupar 1,44 GB a ocupar 645 MB. Por ello, se tiene que la segmentación también ayudará a reducir el tamaño del conjunto de datos.

La herramienta que se implementa en este proyecto facilitará mucho la conformación de este conjunto de datos etiquetado, evitando así el uso de métodos tediosos y/o etiquetas ambiguas que

pueden inducir a error. La observación de estas etiquetas almacenadas permitirá realizar un análisis de la biodiversidad de especies de ave en cada uno de los puntos de *l'Albufera*.

### 3.5 Nodos de grabación

Los nodos que se han distribuido para conformar el conjunto de datos son los *Song Meter Mini* fabricados por la empresa *Wildlife Acoustics* [22]. Estos nodos están diseñados para grabar audio en la naturaleza. Estas grabadoras pueden llevar 1 o 2 micrófonos, se pueden programar para que graben en rangos de horas determinados, permiten la conexión por Bluetooth y pueden funcionar tanto en batería como en pilas. En la Figura 3.1, se muestra una imagen del nodo utilizado y en la Tabla 3.1 se detallan sus características.



Figura 3.1 Nodos de grabación distribuidos en el *Parc Natural de l'Albufera* [22].

Tabla 3.1 Características de los nodos de grabación [22].

Formato de grabación	Archivos WAV 16 bits PCM
Canales	1 canal por micrófono
Frecuencia de muestreo en 1 o 2 canales en Hz	8000, 12000, 16000, 22050, 24000, 32000, 44100, 48000 y 96000
Micrófono	Directividad: Omnidireccional Sensibilidad: 6 dB $\pm$ 4dB (0 dB = 1Pa@1kHz) Relación Señal/Ruido: 78 dB Typ. a 1 kHz Máximo nivel de entrada de sonido: 101 dB SPL Typ.
Autonomía	4 pilas AA: 210 horas (24 kHz mono)
Dimensiones	123 mm x 168 mm x 36 mm
Peso	290 g
Temperaturas a las que puede operar	De -20°C a 85°C

Como se ha mencionado en la sección 1.2, a lo largo del parque hay 10 nodos distribuidos de los cuales 6 cuentan con 1 micrófono y 4 cuentan con 2 micrófonos. En la Figura 3.2, se muestra en un mapa, en qué puntos están ubicados los nodos de grabación en el *Parc Natural de l'Albufera* y, en la Tabla 3.2, se detalla más acerca de la configuración que tienen estos nodos y el nombre que se les ha dado. El nombre del nodo aparecerá en el nombre del archivo WAV que éste almacene después de realizar una grabación.

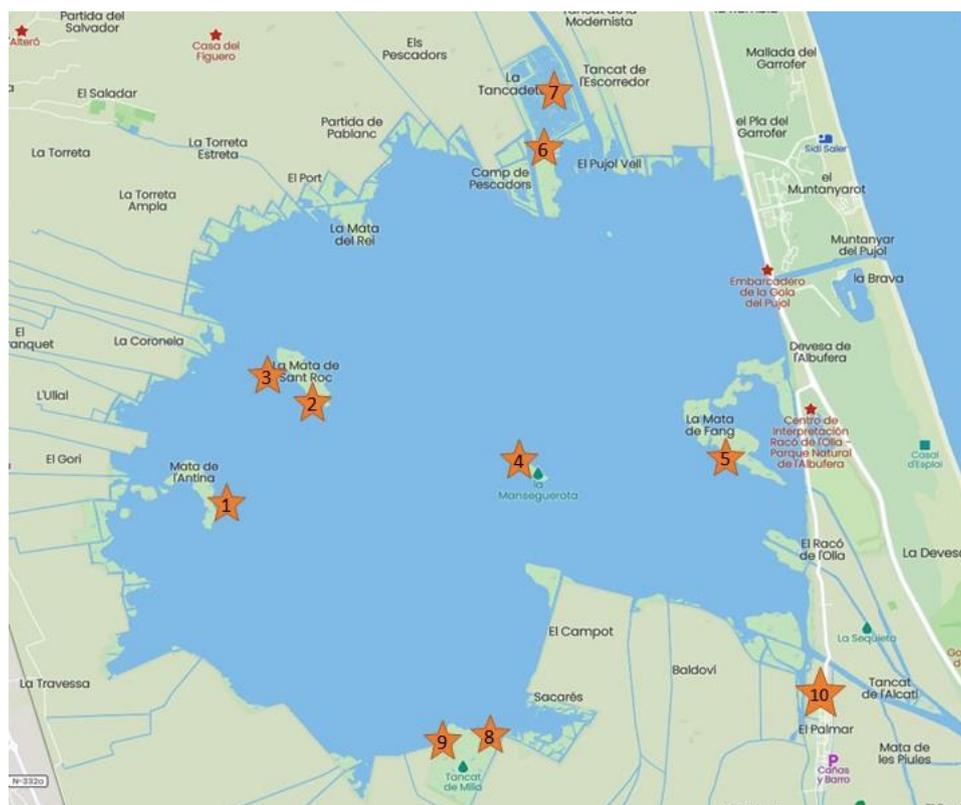


Figura 3.2 Distribución de los nodos de grabación en el Parc Natural l'Albufera.

Tabla 3.2 Ubicación y configuración de los nodos de grabación durante los periodos de primavera y verano.

#	Nodo	Nombre	Ubicación	Horario Amanecer	Horario Ocaso	Micrófonos activos	Frecuencia de muestreo	Instalación
1	SMA07415	Mata de l'Antina	Oeste	4:20 a 10:45	20:00 a 23:59	1	24000	26/05/2022
2	SMA07370	Mata de Sant Roc	Oeste	4:20 a 10:45	20:00 a 23:59	1	24000	26/05/2022
3	SMA07371			4:20 a 10:45	20:00 a 23:59	1	24000	26/05/2022
4	SMA07394	Maseguerota	Centro	4:20 a 13:00	20:00 a 23:59	1	24000	26/05/2022
5	SMA07356	Mata del Fang	Este Centro	4:20 a 13:00	20:00 a 23:59	2	24000	26/05/2022
6	SMA07361	La Tancadeta	Norte	4:20 a 10:45	20:00 a 23:59	1	24000	02/06/2022
7	SMA07400			4:20 a 10:45	20:00 a 23:59	2	24000	02/06/2022
8	SMA07355	Tancat de Milia	Sur	4:20 a 10:45	20:00 a 23:59	2	24000	02/06/2022
9	SMA07340			4:20 a 10:45	20:00 a 23:59	1	24000	02/06/2022
10	SMA07397	Oficina Técnica el Palmar	Sur Este	4:20 a 13:00	20:00 a 23:59	2	24000	02/06/2022

### 3.6 Librosa

La librería de Python *librosa*, es una librería que se utiliza para análisis y procesado de audio. Una de las principales motivaciones para desarrollar esta librería fue la de proveer de una librería de Python que pueda ser utilizada en el campo de la obtención de información musical (en inglés *Music Information Retrieval* [MIR]) [14].

En este trabajo se utiliza *librosa* para generar los espectrogramas que visualizarán los ornitólogos a la vez que escuchan el segmento que tienen que clasificar y para segmentar las grabaciones originales de los nodos de grabación. Asimismo, *BirdNET* también utiliza *librosa* para el



procesado del audio de entrada. La versión de *librosa* utilizada en este trabajo es la versión 0.10.0.post2 [23].

### 3.7 Subsistema de Windows para Linux (WSL)

El Subsistema de Windows para Linux (en inglés, *Windows Subsystem for Linux* [WSL]) permite a los desarrolladores que se pueda ejecutar un entorno Linux directamente en Windows, incluyendo utilidades, líneas de comandos y aplicaciones. Este entorno se ejecuta directamente en Windows sin necesidad de arranque dual o máquina virtual. Entre otras cosas, permite la instalación de software compatible en Linux [24].

En este trabajo se ha utilizado Ubuntu sobre WSL versión 2 al que posteriormente se le han instalado las aplicaciones y librerías necesarias para este proyecto. En este caso, se hace servir Ubuntu sobre WSL versión 2 para ejecutar la herramienta *BirdNET*, y ejecutar los códigos de Python que generan los segmentos, los espectrogramas y las consultas en MySQL que servirán para actualizar la base de datos.

## 4. Metodología

En este capítulo de la memoria se van a definir los procedimientos que se han seguido para alcanzar el objetivo final del trabajo. Es importante definir de qué manera se va a llevar a cabo la implementación de esta herramienta y qué pasos deben seguirse durante este proceso.

Durante el diseño de la herramienta será necesario proponer y estudiar las distintas maneras que existen de implementar la solución, y junto con el usuario final, validar y elegir la propuesta de diseño que sea más adecuada. La retroalimentación del usuario final será muy importante en la toma de decisiones cuando haya que elegir un diseño.

### 4.1 Especificaciones

Lo primero que se hizo fue definir las especificaciones que tendrá que cumplir la herramienta que utilizará el usuario final y cómo se va a implementar.

De esta manera se tiene que el cometido final es disponer de un sistema que sea capaz de convertir varios audios de 1 hora en segmentos de una duración aún no determinada. Además, estos segmentos deberán de ser escuchados uno a uno por un equipo de ornitólogos y estos deben ser capaces de determinar qué especies de ave se escuchan en cada uno de estos archivos. Este proceso generará como salida un segmento de audio con una o varias etiquetas correspondientes a las especies de ave presentes en el audio. Asimismo, en cada segmento se podrá añadir un comentario que corresponderá a las observaciones que el ornitólogo desee destacar de cada uno de los archivos de audio. Este segmento será almacenado con sus etiquetas y comentarios correspondientes. Un segmento podrá ser etiquetado por más de un ornitólogo.

Con todas estas especificaciones en mente, se ha creado un esquema para tener una aproximación o una idea de qué forma podría tener la herramienta final e ir trabajando sobre ella. La creación de un esquema permite estructurar y dar forma a las ideas del proyecto, ayuda a detectar posibles errores o inconsistencias dentro de las especificaciones previas, ayuda a comunicar la propuesta que se desea implementar con los ornitólogos, y permite tener una guía a seguir a la hora de implementar.

En la Figura 4.1, se muestra el esquema que se ha realizado teniendo en cuenta las especificaciones que se han detallado en este apartado, planteando una propuesta de diseño. En esta propuesta de diseño, se tiene una carpeta de entradas donde se pondrán las grabaciones tomadas por los nodos del *Parc Natural de l'Albufera*. Luego, cada uno de estos audios de 1 hora son segmentados en varios segmentos, en este caso se supondrá que los segmentos serán de una duración de 3 segundos, de la misma manera que lo hace *BirdNET* [4]. Teniendo ya los segmentos, la tarea del usuario será etiquetar segmento a segmento qué especie o especies se escuchan a través de la aplicación a diseñar. Dentro de la aplicación, el ornitólogo podrá visualizar el espectrograma del segmento donde el eje de frecuencias sigue la Escala Mel, una representación del segmento en el tiempo y un reproductor de audio que le permitirá escuchar el segmento. También se incluyen dos botones con flechas en la parte superior donde se puede pasar al segmento anterior y al siguiente segmento, en caso de querer repasar algún segmento en específico. Asimismo, habrá tres botones donde el usuario deberá decidir si lo que escuchó fue una especie de ave (botón “Especie”), si escuchó otro tipo de sonido que le pueda resultar de interés para tener en cuenta (botón “Otro”) o, no escuchó directamente nada relevante (botón “Nada”). Así pues, la aplicación actuará de una manera u otra dependiendo de la opción que se elija. A continuación, se describe cómo actúa internamente la aplicación en cada una de las opciones:

- Botón “Especie”: Cuando el usuario ha escuchado una especie de ave, la aplicación cambia la interfaz, manteniendo todos los elementos y solamente cambiando los 3 botones mencionados anteriormente por 3 desplegables donde pueden seleccionar hasta 3 especies que hayan escuchado en el segmento. Se puede seleccionar 1 o más especies e introducir el

porcentaje de confianza en que esa especie está presente en el audio. Luego hay una casilla de comentarios en la que el usuario puede añadir observaciones que desee incluir acerca del segmento. Una vez se ha introducido todo, hay un botón de enviar que sirve para guardar el resultado de etiquetado del segmento en una fila de un archivo CSV.

- Botón “Otro”: Similar a la opción anterior, aunque solamente estará habilitada la caja de comentarios para indicar qué sonido se ha escuchado en ese segmento. Esta opción también genera una fila en el CSV solamente que en nombre de especie el valor será de “Otro”.
- Botón “Nada”: Al pulsar este botón simplemente se descarta el segmento y se pasa al siguiente. Como se puede suponer, esta opción no genera ninguna fila en el CSV.

A medida que se van obteniendo segmentos etiquetados, se van introduciendo los resultados de clasificación de éstos dentro de un archivo CSV asociado a una grabación de una hora. Una vez se han completado todos los segmentos de un archivo de una hora, se genera un fichero CSV que se asocia a esa grabación con todos los resultados de la clasificación.

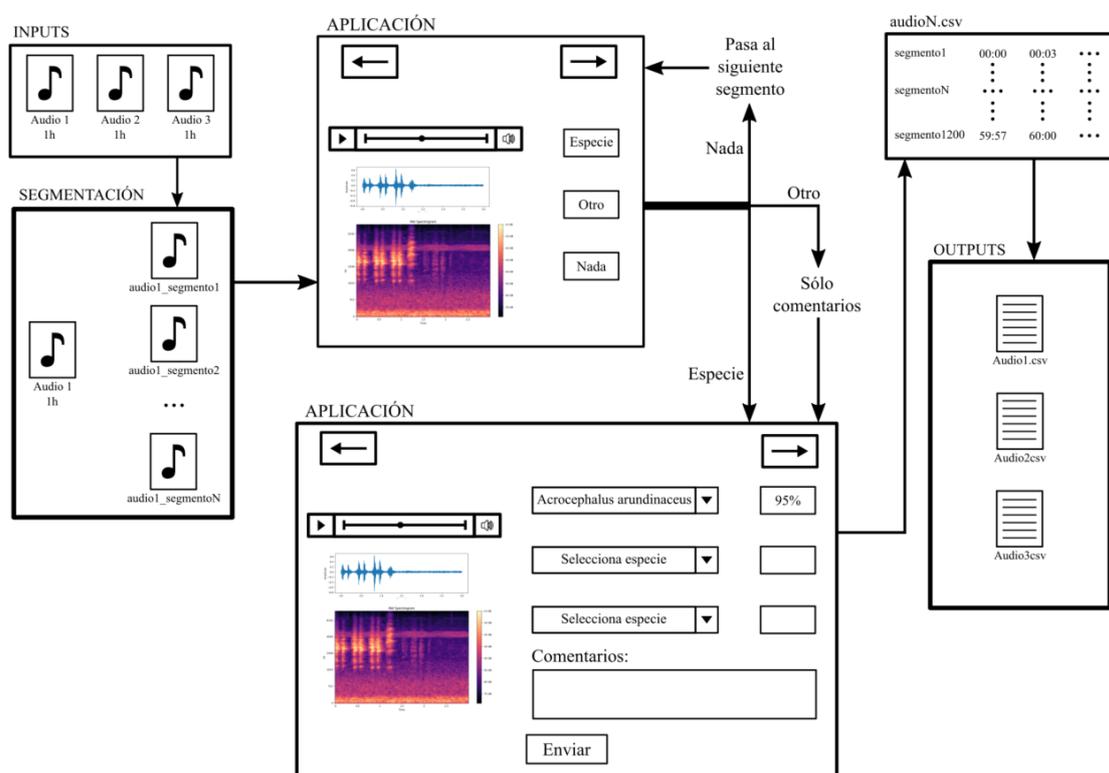


Figura 4.1 Primera propuesta de diseño de la herramienta.

Es importante destacar que esta propuesta de diseño es una propuesta inicial y está sujeta a variaciones. Aun así, sirve como base para empezar a implementar la herramienta y aplicar las variaciones que sean necesarias a medida que se va desarrollando.

## 4.2 Base de datos

### 4.2.1 Elección del modelo de base de datos

Teniendo en cuenta las especificaciones de la herramienta definidas en la sección 4.1, será necesaria la recopilación de la información relacionada con los segmentos que etiqueten los usuarios. A priori, y como se pudo observar en las especificaciones, la idea previa era tener un archivo CSV por cada audio y usuario con las etiquetas proporcionadas. Esta implementación queda descartada, ya que, consultar y filtrar información a través de varios ficheros CSV puede resultar un proceso tedioso y lento. Además, si la cantidad de datos aumenta en tamaño, los ficheros pueden ser más complejos de administrar.

Por otro lado, las bases de datos, en comparación a los CSV, disponen de una estructura para guardar los datos y tenerlos organizados de una forma eficaz. Asimismo, las bases de datos están pensadas para administrar cantidades voluminosas de datos y ofrecen una excelente escalabilidad acorde al crecimiento del volumen de los datos. La obtención de información de una base de datos se realiza a través de comandos llamados consultas que se ejecutan en la base de datos. Con estas consultas se pueden añadir, obtener o modificar datos.

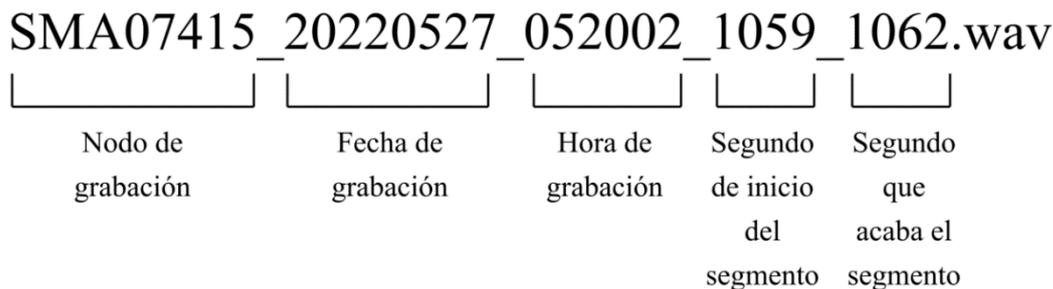
Con lo expuesto anteriormente, se elige la base de datos como sistema para almacenar la información relacionada a los segmentos y a sus etiquetas. Esto se decide porque el volumen de datos que se tendrá que gestionar será elevado, teniendo en cuenta que también se almacenarán dentro de la base de datos las etiquetas que genere *BirdNET*, y porque ofrece una mayor escalabilidad que la gestión de ficheros CSV.

Para este proyecto se eligió el modelo de base de datos relacional. En una base de datos relacional, los datos se almacenan en tablas que están compuestas por filas y columnas. Cada una de las columnas corresponde a cada uno de los campos de la tabla, y las filas corresponden a cada uno de los registros que se almacenan dentro de la tabla. De esta manera, por cada dato que se introduce en una tabla se genera un nuevo registro, y todos los datos que estén en una misma columna serán del mismo tipo (campo numérico, texto, etc.). En el modelo de base de datos relacional, los datos de diferentes tablas se relacionan entre sí mediante relaciones. Esta relación se puede establecer, por ejemplo, haciendo que dos tablas tengan una columna común. Este valor común servirá para relacionar un registro de una tabla con un registro de la otra. La clave primaria (en inglés, *primary key*) es un campo que es único para cada uno de los registros de una misma tabla, sirve para reconocer cada registro de la tabla de manera única. Dicho de otra manera, la clave primaria es un campo que permite identificar cada fila de una tabla de forma individual. Por otro lado, la clave foránea (en inglés, *foreign key*) es un campo que se relaciona con la clave primaria correspondiente a otra tabla distinta, creando así lo que se conoce como una relación. La relación que se puede establecer entre los datos permite que se puedan gestionar de manera eficaz dado a que se pueden relacionar entre sí [25].

#### 4.2.2 *Diseño de la base de datos*

Una vez tomada la decisión de que se va a implementar una base de datos, el primer paso que se ha seguido para diseñarla ha sido definir qué datos se quieren almacenar en esta base de datos. En este proyecto tenemos datos asociados a los usuarios, a los segmentos de audio y a las especies de ave. Con esto se pueden intuir 3 tablas dentro de la base de datos: una tabla asociada a los usuarios, otra asociada a los segmentos de audio y otra asociada a las especies. A partir de estas 3 tablas se puede llegar a una cuarta tabla que servirá para relacionar a todas entre sí. Otra forma de verlo es que, la cuarta tabla se encargará de especificar qué etiqueta (especie de ave) dio un usuario a un determinado segmento.

Los segmentos de las grabaciones originales serán almacenados dentro de un directorio. El nombre de archivo de un segmento está formado por el nombre de la grabación original, la que dura una hora, y dos sufijos: el primer sufijo corresponde al segundo en el que empieza el segmento y el segundo sufijo corresponde al segundo en el que termina el segmento. En la Figura 4.2, se muestra un ejemplo de nombre de archivo de un segmento, donde se indica qué nodo grabó el segmento, la fecha y hora en que se grabó y, los segundos de inicio y final del segmento dentro de la grabación original.



**Figura 4.2** Descripción del nombre de archivo de un segmento.

En primer lugar, con lo que respecta a la tabla relacionada a los segmentos, será una tabla que almacenará el nombre de la grabación original de la cual procede este segmento, el segundo en el que empieza este segmento dentro de la grabación original y el segundo en el que termina este segmento con respecto a la grabación original. Para acceder a un segmento relacionado a un registro, simplemente hay que formar el nombre del segmento utilizando el campo del nombre del archivo original junto con los campos del segundo de inicio y del segundo que finaliza el segmento. A modo de ejemplo, el segmento de la Figura 4.2 quedaría guardado dentro de la base de datos como SMA07415\_20220527\_052002.wav en el campo del nombre del archivo, 1059 en el campo de segundo de inicio y 1062 en el segundo que finaliza el segmento. Cabe destacar, que en el nombre de las grabaciones originales proporcionadas por los nodos de grabación ya va implícito el nombre del nodo que generó la grabación, la fecha en la que se hizo la grabación y la hora exacta en la que se inició la grabación. Por esta razón, no será necesario almacenar dentro de la base de datos la información que ya va implícita en el nombre del archivo, para optimizar espacio. Por último, hay que añadir que esta tabla tendrá una clave primaria asociada a cada uno de los segmentos de audio que se vayan introduciendo a la base de datos.

En segundo lugar, con lo que respecta a la tabla relacionada a las especies de ave, será una tabla que almacenará los nombres científicos de las aves y su nombre común. A la hora de etiquetar los segmentos, los ornitólogos visualizarán los nombres científicos y no los nombres comunes, para evitar confusiones. Además, aunque no se traten de especies, se han creado 3 registros genéricos que servirán para indicar si la especie que se escucha corresponde a un ave que no se encuentra en la lista de especies, si el sonido corresponde a otra especie de animal y si corresponde a otro tipo de ruido. En estos casos, como se verá más adelante, el ornitólogo podrá incluir un comentario a la hora de etiquetar el sonido para especificar qué otro tipo de sonido es. También se ha creado un registro genérico que se utilizará en caso de que el ornitólogo desee saltar a otro segmento en caso de no tener claro qué especie está presente en el segmento. Posteriormente, si el usuario lo desea, puede retomar la muestra que se dejó por etiquetar. La ventaja de tener una tabla de especies es que facilita la tarea de obtener segmentos de alguna especie en específico. De la misma manera que en la tabla anterior, cada especie tendrá una clave primaria.

En tercer lugar, con lo que respecta a los usuarios de la herramienta, se ha creado una tabla que recopilará sus datos. Los datos de los usuarios serán su nombre de usuario, el nombre, el apellido, su contraseña para acceder a la herramienta y un correo electrónico. El nombre de usuario deberá ser único para cada uno. Así pues, la contraseña de acceso a la herramienta nunca será almacenada en texto plano en la base de datos y se utilizará un algoritmo de encriptación para almacenarla. Como en todas las tablas, cada usuario tendrá una clave primaria. En este caso, se ha contado la herramienta *BirdNET* como un usuario más, ya que, así es más sencillo de cara a la gestión de la base de datos. Así pues, las etiquetas generadas a través de *BirdNET* serán almacenadas dentro de la base de datos como si se tratara de un ornitólogo más. La clave primaria del usuario *BirdNET* o su identificador es el número 1.

Por último, a partir de estas 3 tablas anteriores se ha creado una cuarta tabla que servirá para relacionar los registros de éstas, añadiendo también la información correspondiente a la etiqueta que se le dé a cada uno de los segmentos. En esta tabla hay 3 campos que se corresponden a cada una de las claves foráneas que están relacionadas a las claves primarias de las otras tablas. Pudiendo así relacionar a los usuarios, los segmentos y las especies. Asimismo, cada registro que se incluya en esta nueva tabla tendrá su propia clave primaria. Cada fila de esta tabla corresponde a un segmento etiquetado. A todos los campos anteriores, se añade un campo numérico que sirve para que el ornitólogo pueda indicar qué tanto seguro está de que una determinada especie está presente en el segmento. Este campo se llamará confianza. La motivación de que exista un campo de confianza es que *BirdNET* también proporciona valores de confianza sobre la clasificación, además de que, permite dar una importancia distinta a cada una de las etiquetas. Concluyendo con esta tabla, se ha añadido un campo de texto para que el ornitólogo pueda incluir comentarios con relación a lo que se ha escuchado en un segmento. Este campo sirve tanto para especificar especies o fuentes sonoras que no aparecen listadas, como para añadir comentarios relevantes relacionados con el segmento.

### 4.2.3 Resultado final

Teniendo en cuenta lo descrito en la sección 4.2.2, el diseño final de la base de datos quedaría tal y como se muestra en la Figura 4.3. En esa imagen se pueden observar las tablas *usuarios*, *audios* y *species* que hacen referencia a los usuarios, a los segmentos de las grabaciones originales y a las especies de ave respectivamente. La tabla *classification* es la tabla que relaciona las 3 tablas anteriores. Las tablas *usuarios*, *audios* y *species* cuentan con los campos *UserID*, *AudioID* y *SpecieID* que corresponden a las claves primarias de cada una de las tablas. Estos 3 campos también aparecen en la tabla *classification* como claves foráneas que sirven para relacionar cada una de las tablas anteriores con ésta. Estas relaciones se pueden observar gráficamente en las conexiones que hay dibujadas entre las distintas tablas. Como detalle, cabe destacar que la tabla *audios* se ha implementado de tal manera que sólo pueda haber un registro por segmento, dicho de otro modo, para que no existan duplicidades en los segmentos que hay almacenados en la base de datos. De la misma manera, tampoco podrán existir etiquetas duplicadas.

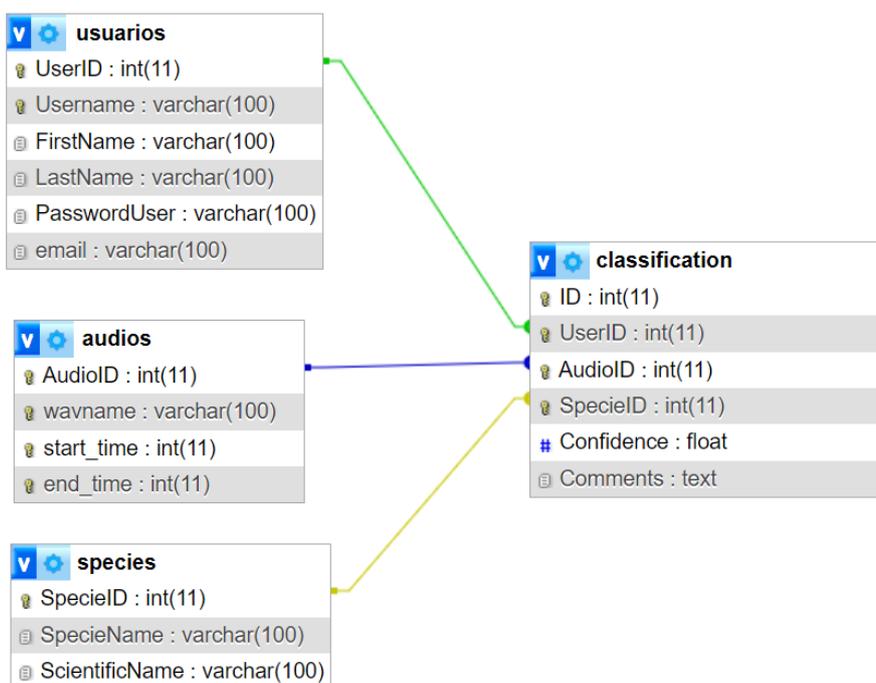


Figura 4.3 Diseño final de la base de datos.

### 4.3 Segmentación de las grabaciones

#### 4.3.1 Segmentado

Con lo que respecta al segmentado de las grabaciones originales, es importante establecer cómo se generarán los segmentos que etiquetarán los usuarios a través de la herramienta.

En primer lugar, hay que considerar si los segmentos tendrán una duración fija o, en su lugar, una duración variable según el ornitólogo genere el segmento. Que los segmentos puedan generarse en el proceso de etiquetado de forma manual, causará que éstos tengan una duración variable según el ornitólogo que lo genere. Este proceso añade un grado de dificultad considerable en la implementación de la herramienta, ya que, implica implementar un editor de audio que esté integrado dentro de la herramienta que funcione perfectamente y que sea robusto. Este editor debe tener un funcionamiento que sea intuitivo, teniendo así que implementar una interfaz gráfica que sea amigable con el usuario. Así pues, con esta opción, el usuario visualizaría un espectrograma de 1 hora y sobre este espectrograma iría generando recortes (segmentos) y etiquetándolos. Esta solución causa varios problemas. El primero de estos problemas es que se debe generar un espectrograma de una grabación de 1 hora que tenga la suficiente resolución como para que el usuario sea capaz de detectar todos los cantos de pájaros que pueda haber presentes. El segundo problema es que este proceso de generar manualmente los segmentos añade más carga de trabajo al usuario final, teniendo que ser responsable de recordar qué partes de la grabación ya etiquetó y que partes no. Y, por último, otro problema que puede añadir este proceso es que cada ornitólogo generará sus propios segmentos, teniendo así que cada usuario generará segmentos distintos o ligeramente distintos entre ellos. De esta manera, un segmento de un usuario corresponderá a unos segundos específicos de la grabación original y otro segmento generado por otro usuario corresponderá a otros segundos específicos de la grabación original, teniendo que sendos segmentos estén en instantes parecidos con respecto a la grabación original. Esto causa que comparar etiquetas dadas por diferentes ornitólogos sea un proceso más complejo. Sobre todo, también dificulta la comparación de las etiquetas proporcionadas por los ornitólogos con las que proporciona *BirdNET*. Por todo lo anterior, se ha elegido la opción de que los segmentos tengan una duración fija y que sean generados de forma automática.

En segundo lugar, hay que considerar si los segmentos van a estar almacenados físicamente como ficheros WAV o si podrán ser almacenados simplemente como referencias temporales a las grabaciones originales almacenando el segundo de inicio y el segundo que finaliza el segmento. En la sección 3.4 se ha comentado que dentro de las grabaciones originales existen instantes en los que no hay información relevante, en este caso serían momentos en los que no se percibe ningún canto de ninguna especie de ave. Dentro del conjunto de datos se puede prescindir de estos instantes en los que no hay ningún canto de ave y conservar los segmentos en los que sí que hay cantos. Esto reduciría significativamente el espacio de almacenamiento necesario para el conjunto de datos. Por este motivo, la decisión que se toma es que el audio esté ya segmentado dentro del conjunto de datos.

#### 4.3.2 Duración del segmento

La duración del segmento es uno de los parámetros más determinantes a la hora de procesar el espectrograma. Teniendo en cuenta que el espectrograma es una imagen, esta imagen tiene que ser pequeña si se quiere conseguir un bajo coste computacional. Una de las maneras de reducir el espectrograma en el eje temporal es limitando la longitud del segmento. En base a datos empíricos extraídos de sonidos etiquetados, se tiene que la media de los cantos de cientos de especies es de 1,94 segundos. De esta manera, una duración de segmento de 3 segundos será suficiente para representar la gran mayoría de cantos [26].

Por otro lado, tal y como se comenta en la sección 3.4, se tiene que *BirdNET* será utilizado para conservar los segmentos que contienen algún canto de ave y descartar los segmentos que no contengan información relevante con relación a los cantos de pájaro, evitando que los ornitólogos

tengan que invertir tiempo escuchando segmentos que no aportan ninguna información. Los segmentos que proporciona *BirdNET* tienen una duración de 3 segundos.

Con todo lo anterior, se decide que la duración de los segmentos del conjunto de datos sea de 3 segundos. De esta manera, se aprovecha el segmentado que hace *BirdNET* de las grabaciones, pudiendo, además, realizar una comparación y validación directa de las etiquetas que proporciona el modelo a cada uno de los segmentos. Así pues, la segmentación de las grabaciones resultará en un proceso más sencillo, donde se toman los segundos de inicio y final de cada uno de los segmentos que ya proporciona *BirdNET*, para posteriormente generar los segmentos a través de un algoritmo que se implementa en Python.

En la Tabla 4.1 se muestran las características que tienen los segmentos en este proyecto de acuerdo con las decisiones que se han tomado a lo largo de esta sección.

**Tabla 4.1** Resumen de las características de los segmentos.

Proceso de segmentación	Automático
Longitud del segmento	Fija
Duración del segmento	3 segundos

#### 4.3.3 Script para segmentar las grabaciones

Con todo detallado, ya se puede empezar a implementar un *script* en Python que divida las grabaciones de 1 hora tomadas por los nodos en segmentos de 3 segundos a partir de las etiquetas proporcionadas por *BirdNET*. Antes de ejecutar el algoritmo, se colocan las grabaciones originales que se deseen segmentar dentro de un directorio y los CSV originados por el modelo de clasificación en otro directorio. El *script* hará una iteración por cada pareja de archivos WAV y CSV que existan en los directorios de entrada. A continuación, se describe el proceso que sigue el *script* en una iteración:

1. Se toma una pareja de fichero de audio y de CSV, siendo el CSV una salida generada del fichero WAV, y se cargan para leer su información. Ambos archivos se relacionan porque tienen el mismo prefijo que los distingue.
2. Se toman los valores de segundos de inicio y de segundos de fin que aparecen en los ficheros CSV y se almacenan en 2 variables distintas
3. Con cada una de las parejas de segundo de inicio y de segundo de fin que aparecen en cada fila del CSV, se crea un segmento hasta haber recorrido todas las filas del CSV.
4. Se almacenan los segmentos en un directorio de salida nombrándolos según el formato de la Figura 4.2. Cabe recordar que estos segmentos de 3 segundos son los segmentos en los que *BirdNET* encontró alguna especie de ave.
5. Si hay más grabaciones por segmentar, se vuelve al paso 1.

#### 4.4 Definición de la herramienta

Teniendo en cuenta las especificaciones que se detallan en la sección 4.1 y la decisión de diseño de almacenar las etiquetas en una base de datos, detallada en la sección 4.2.1, se procede a determinar cómo será implementada la interfaz de usuario.

La primera idea que se tuvo fue la de implementar una interfaz de usuario que funcionara sin necesidad de conexión a Internet y completamente en el equipo local. Esta idea se puede visualizar en la propuesta de diseño que aparece en la Figura 4.1. Dado que en este proyecto se gestiona un conjunto de datos grande (alrededor de 4 TB), no es viable compartir el conjunto de datos en una o varias unidades físicas y recopilar las etiquetas que generen los ornitólogos en diferido. Además, es necesario que la herramienta pueda utilizarse desde cualquier equipo. Así pues, tareas como la segmentación de las grabaciones, la generación de espectrogramas o que el uso de *BirdNET* de

forma local podrían resultar lentas en algunos equipos lo que afectaría en el rendimiento de la herramienta. Una de las ventajas que presenta este diseño es que es más robusto en cuanto a seguridad, ya que, no queda ningún servidor expuesto a usuarios malintencionados. Además, la implementación, al menos a priori, puede resultar sencilla de realizar, pudiendo desarrollar la herramienta completamente en Python. Aun así, teniendo en cuenta las desventajas que presenta este diseño, queda descartada la implementación en local y sin conexión a Internet.

Una solución con conexión a Internet presenta la desventaja de que será necesaria la implementación de diferentes mecanismos de seguridad para evitar que usuarios malintencionados puedan interactuar con la herramienta. Otra de las desventajas que presenta es que requiere de una complejidad mayor, dado que será necesario que el equipo del usuario esté conectado a un servidor. Por otra parte, una de las ventajas que presenta es que el conjunto de datos se encontraría almacenado en el servidor sin necesidad de que esté en el equipo del usuario. Asimismo, con la herramienta conectada a Internet, las etiquetas se obtienen en el mismo instante que las genera el ornitólogo, pudiendo insertarlas directamente en la base de datos. De esta manera, el usuario no debe preocuparse de entregar cualquier archivo o de tener conocimientos sobre qué archivos debe de enviar de vuelta. El único esfuerzo que recae encima del usuario es el de abrir la herramienta, escuchar segmentos y etiquetar los segmentos. Además, la carga computacional no recae en el cliente sino en el servidor, pudiendo tener las grabaciones ya segmentadas y los espectrogramas ya procesados. Teniendo en cuenta todas las ventajas que aporta esta solución, se toma la decisión de que la herramienta esté conectada a Internet.

Teniendo en cuenta lo anterior, implementar la herramienta como una aplicación web que podrá ser accesible desde un navegador es una opción interesante. Una de las principales ventajas de esta opción es que el único requisito que debe tener un equipo para utilizar una aplicación web es que tenga un navegador web y conexión a Internet, haciendo también que la herramienta pueda funcionar en cualquier sistema operativo. Al estar alojada en un servidor web, la herramienta siempre estará actualizada de forma centralizada. Además, el usuario sólo necesita tener el enlace y unas credenciales para poder utilizar la herramienta. El mantenimiento de la web es sencillo y permite añadir nuevas funcionalidades sin que genere problemas en su funcionamiento. Con todo esto, se toma la decisión de que la herramienta esté implementada como una aplicación web a la que el usuario accederá a través de unas credenciales.

El servidor sobre el que irá implementado la herramienta alojará la página web, la base de datos, almacenará los segmentos de las grabaciones originales y almacenará los espectrogramas de los segmentos que se hayan generado. De esta manera, se tiene que la herramienta consistirá en un servidor web que alojará las aplicaciones web, los segmentos y los espectrogramas, y un servidor *MySQL* que alojará la base de datos. Por otro lado, de manera local, se procesarán las grabaciones de audio extrayendo las etiquetas que genere *BirdNET*, y con un script se generarán los segmentos, los espectrogramas de estos segmentos, las consultas que servirán para insertar la información de los segmentos en la base de datos y las consultas que servirán para insertar las etiquetas generadas por *BirdNET* en la base de datos.

En la Tabla 4.2 se muestran las características que tendrá la herramienta en cuanto a cómo será desplegada de acuerdo con las decisiones que se han tomado en esta sección.

**Tabla 4.2** Resumen de las características de la herramienta.

¿Conexión a Internet?	Sí
Plataforma	Aplicación web
Servidor web	Remoto
Base de datos	Remota

## 4.5 Estudio y aplicación de *BirdNET*

### 4.5.1 Instalación de *BirdNET* y primera ejecución

En primera instancia, se ha instalado un entorno Ubuntu sobre WSL en el ordenador local. La interfaz de este entorno consiste en una línea de comando. En la Figura 4.4 se observa el aspecto que tiene el entorno Ubuntu que se ha instalado.

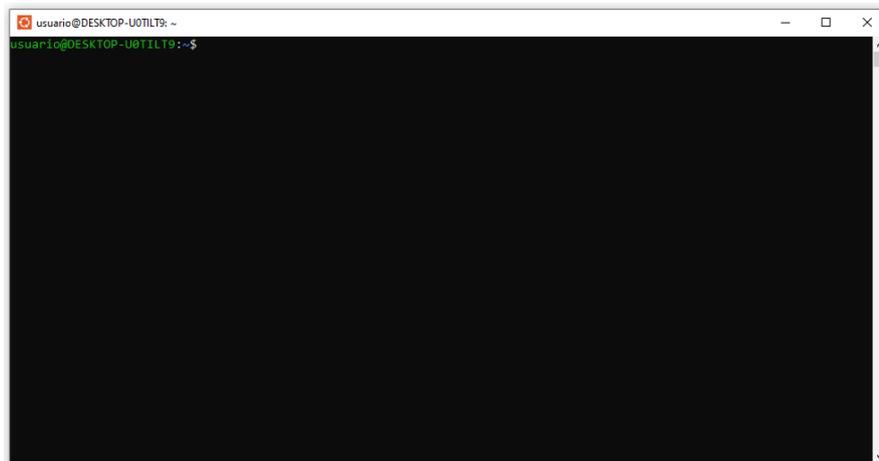


Figura 4.4 Aspecto del entorno Ubuntu sobre WSL.

Tras esto, se ha estudiado el funcionamiento de *BirdNET*. En su repositorio [20] se explica cómo se instala *BirdNET* y cómo se ejecuta. En primer lugar, se ha clonado el repositorio de GitHub en local, en segundo lugar, se han instalado las librerías y dependencias necesarias, y, en tercer lugar, se ha ejecutado *BirdNET* para que analice una grabación de un canto de pájaro tomada con el teléfono móvil a modo de prueba. A la hora de ejecutar el comando que inicia un análisis en *BirdNET* es posible incluir una serie de argumentos, entre los posibles argumentos que se pueden incluir se hará uso de los siguientes: el que especifica el directorio de entrada, el que especifica el directorio de salida, el que especifica el formato de los resultados del análisis, la latitud de la grabación, la longitud de la grabación, el que especifica el grado de confianza mínima sobre la etiqueta que da al segmento y el número de semana en el que se realiza la grabación. En cuanto a estos argumentos, se desea que el formato de salida sea CSV para que sea más sencillo de procesar en Python, se establecen las coordenadas de *l'Albufera* en los argumentos de latitud y longitud, se establece la confianza mínima en el valor de 0,1 para conservar todas las etiquetas que *BirdNET* pueda proporcionar y el número de semana se pone en -1 para que no tenga en cuenta este parámetro en el etiquetado. Aunque se pueda especificar que los nombres comunes de las especies aparezcan en español en la salida, esta opción queda descartada, ya que, se usarán los nombres científicos de las especies para distinguir entre unas aves y otras, y porque las tildes no se guardan adecuadamente en un fichero CSV.

### 4.5.2 Umbral de confianza de *BirdNET*

Tras poner en marcha *BirdNET*, se procede a estudiar qué efectos tienen sobre la salida los argumentos que se pueden especificar en los comandos cuando se ejecuta *BirdNET*.

El primer parámetro que se va a estudiar es el umbral de confianza de *BirdNET*. Cuando *BirdNET* proporciona las etiquetas de una grabación, especifica, al mismo tiempo, qué grado de confianza tiene el algoritmo sobre la etiqueta que proporciona al segmento. Dicho de otra manera, cómo de seguro está el algoritmo de que la especie que proporciona en la etiqueta esté presente en el segmento que clasifica. Este umbral de confianza puede ser configurado con valores comprendidos desde 0,01 hasta 0,99, siendo el primer valor un 1% de confianza y el segundo valor un 99% de confianza. El valor predeterminado de este argumento si no se especifica es de

0,1. Lo que se va a estudiar es qué variación va a haber en los resultados probando diferentes valores de umbral de confianza.

Para estudiar este argumento, se han escogido 9 grabaciones de 1 hora cada una del *Parc Natural de l'Albufera* y se han generado las etiquetas utilizando umbrales de confianza del 0,1 al 0,9 con saltos de 0,1. Se ejecuta el algoritmo con la configuración predeterminado solamente especificando diferentes umbrales de confianza. Al analizar las salidas, se observa que las etiquetas proporcionadas tienen valores de confianza que se encuentran por encima del especificado. Tras este análisis, se ha comprobado que una salida a la que se la especificado un umbral inferior, contiene a otra salida a la que se la especificado un umbral superior, con las mismas etiquetas. En otras palabras, el modelo que utiliza *BirdNET* para etiquetar los segmentos va a ser el mismo independientemente del umbral de confianza que se especifique. Por lo tanto, se puede concluir que el umbral de confianza actúa como un filtro que se aplica a la salida una vez se han generado las etiquetas. En la

Tabla 4.3 se comparan las etiquetas generadas con *BirdNET* utilizando dos umbrales de confianza distintos. Se puede observar cómo las etiquetas generadas con un umbral del 70% están contenidas dentro de las etiquetas obtenidas estableciendo un umbral del 50%. El nombre del fichero utilizado para obtener esta tabla es *SMA07415\_20220526\_200000.wav*.

**Tabla 4.3** Ejemplo de etiquetas utilizando umbrales de confianza distintos,

Umbral de confianza: 50%				Umbral de confianza: 70%			
Start (s)	End (s)	Scientific name	Confidence	Start (s)	End (s)	Scientific name	Confidence
1170	1173	<i>Acrocephalus arundinaceus</i>	0,7903	1170	1173	<i>Acrocephalus arundinaceus</i>	0,7903
1194	1197	<i>Acrocephalus arundinaceus</i>	0,7609	1194	1197	<i>Acrocephalus arundinaceus</i>	0,7609
1200	1203	<i>Acrocephalus arundinaceus</i>	0,5820	1227	1230	<i>Acrocephalus arundinaceus</i>	0,7754
1218	1221	<i>Acrocephalus arundinaceus</i>	0,5230				
1227	1230	<i>Acrocephalus arundinaceus</i>	0,7754				
1278	1281	<i>Aegolius acadicus</i>	0,6915				
1284	1287	<i>Aegolius acadicus</i>	0,5556				
2880	2883	<i>Chroicocephalus ridibundus</i>	0,5667				

Además, el tiempo que tarda en generar las etiquetas es prácticamente el mismo independientemente del umbral de confianza que se establece. Con esto y con que el modelo que se utiliza para etiquetar va a ser siempre el mismo independientemente del umbral de confianza que se utilice, se decide dejar el umbral de confianza en el valor predeterminado que es del 10% (0,1). Con esto, se almacenarán todas las etiquetas que genere *BirdNET* conservando la mayor cantidad de información y pudiéndola filtrar posteriormente con cualquier *script* en caso de que sea necesario.

#### 4.5.3 Validación de los resultados de *BirdNET*

Para confiar en el funcionamiento de *BirdNET* es necesario conocer cómo de preciso es el algoritmo etiquetando muestras provenientes de las grabaciones tomadas en *l'Albufera*. Para ello, es necesario disponer de audios etiquetados por los ornitólogos.

Para realizar una validación de los resultados de *BirdNET*, los ornitólogos etiquetaron 9 ficheros de audio generados por el nodo de grabación SMA07415. Estas grabaciones son las que están comprendidas entre el día 26 de mayo del 2022 y el 27 de mayo del 2022, y las horas en las que se empezaron a tomar fueron a las 20:00, 21:00, 22:00, 23:00, 4:20, 5:20, 6:20, 7:20 y 8:20. Para esta tarea los ornitólogos siguieron una metodología en la que se especificaban qué especies de ave habían escuchado y entre que minutos se habían encontrado. Realizaron esta tarea utilizando el editor de audio *Audacity* donde se generaron el espectrograma de toda la grabación para visualizar en qué instantes había presencia de cantos de ave. Tras escuchar diferentes tomas e identificar qué ave se escuchaba en cada una de ellas, se han insertado los resultados en un fichero Excel. A continuación, en la Tabla 4.4, se visualiza cómo los ornitólogos anotaron las diferentes especies que detectaban en las grabaciones utilizando la metodología que se acaba de detallar.

**Tabla 4.4** Extracto de las etiquetas generadas por los ornitólogos utilizando Excel

Fichero wav	Hora inicio	Hora fin	Columna1	Especie Acústica	Confianza	Comentarios
SMA07415_20220526_200000.wav	0:19:27	0:20:31	0:01:04	Acrocephalus arundinaceus	100%	Canto - hay mucho ruido. Eliminar del audio completo
SMA07415_20220526_200000.wav	0:26:23	0:26:24	0:00:01	Acrocephalus arundinaceus	100%	Canto
SMA07415_20220526_200000.wav	0:30:38	0:30:41	0:00:03	Acrocephalus arundinaceus	100%	Canto
SMA07415_20220526_200000.wav	0:30:56	0:30:58	0:00:02	Acrocephalus arundinaceus	100%	Canto
SMA07415_20220526_200000.wav	0:43:06	0:43:08	0:00:02	Acrocephalus arundinaceus	100%	Canto
SMA07415_20220526_200000.wav	0:43:15	0:43:17	0:00:02	Acrocephalus arundinaceus	100%	Canto
SMA07415_20220526_200000.wav	0:43:34	0:43:35	0:00:01	Acrocephalus arundinaceus	100%	Canto
SMA07415_20220526_200000.wav	0:43:43	0:43:46	0:00:03	Acrocephalus arundinaceus	100%	Canto
SMA07415_20220526_200000.wav	0:43:48	0:43:48	0:00:00	Acrocephalus arundinaceus	100%	Reclamo
SMA07415_20220526_200000.wav	0:45:18	0:45:23	0:00:05	Chroicocephalus ridibundus	100%	Voces de adulto, lejanas
SMA07415_20220526_200000.wav	0:47:11	0:47:17	0:00:06	Ardea cinerea	100%	
SMA07415_20220526_200000.wav	0:48:00	0:48:05	0:00:05	Chroicocephalus ridibundus	100%	Voces de adulto
SMA07415_20220526_200000.wav	0:48:06	0:48:06	0:00:00	Chroicocephalus ridibundus	100%	Voz de agresión
SMA07415_20220526_200000.wav	0:48:20	0:48:21	0:00:01	Chroicocephalus ridibundus	100%	Voz de agresión
SMA07415_20220526_200000.wav	0:48:22	0:48:24	0:00:02	Acrocephalus arundinaceus	100%	Canto
SMA07415_20220526_200000.wav	0:48:40	0:48:41	0:00:01	Porphyrio porphyrio	100%	
SMA07415_20220526_200000.wav	0:48:41	0:48:56	0:00:15	Porphyrio porphyrio	100%	
SMA07415_20220526_200000.wav	0:57:10	0:57:11	0:00:01	Acrocephalus arundinaceus	100%	Canto
SMA07415_20220526_200000.wav	0:57:46	0:57:48	0:00:02	Acrocephalus arundinaceus	100%	Canto
SMA07415_20220526_200000.wav	0:58:05	0:58:07	0:00:02	Acrocephalus arundinaceus	100%	Canto
SMA07415_20220526_200000.wav	0:58:13	0:58:15	0:00:02	Acrocephalus arundinaceus	100%	Canto
SMA07415_20220526_200000.wav	0:58:40	0:58:43	0:00:03	Acrocephalus arundinaceus	100%	Canto
SMA07415_20220526_210002.wav	0:02:09	0:02:17	0:00:08	Himantopus himantopus	100%	
SMA07415_20220526_210002.wav	0:02:20	0:02:30	0:00:10	Himantopus himantopus	100%	

Así pues, como estas etiquetas tienen un formato ligeramente distinto a las etiquetas que proporciona *BirdNET*, hay que adaptarlas para que ambos formatos sean comparables. Para ello, se ha realizado un análisis y una serie de procesados sobre el fichero Excel proporcionado por los ornitólogos. En el análisis se ha encontrado que los usuarios han utilizado la columna de comentarios para incluir más especies de ave que han escuchado en un fragmento de tiempo además de la que han especificado en la columna correspondiente a la especie de pájaro. Además de adaptar las etiquetas proporcionadas por los ornitólogos al formato de *BirdNET*, se han considerado las especies que colocaron los ornitólogos en el apartado de comentarios como etiquetas, teniendo así más de una etiqueta para un mismo fragmento de audio, al igual que hace *BirdNET*. Asimismo, también se asume que las etiquetas proporcionadas por los ornitólogos corresponden a la especie que realmente se escucha en el fragmento etiquetado.

Finalmente, se pueden validar los resultados de *BirdNET* teniendo que, en estas 9 grabaciones, acertó con 357 segmentos y falló con 245 segmentos. También se tiene que hay 130 segmentos en los que no encontró ninguna especie y los ornitólogos sí. Dentro de los segmentos en los que tanto *BirdNET* como los especialistas detectaron una especie, se tiene que el modelo acertó en el 59,30% de los casos y falló en el 40,70% de los segmentos. Este porcentaje de aciertos dista del 79,1% que se menciona en la sección 3.2. Debe tenerse en cuenta la principal limitación que tiene esta validación y es que, por el momento, el conjunto de segmentos etiquetados por los ornitólogos es bastante pequeño para realizar una validación del modelo de *BirdNET*. Todos estos segmentos duran 30 minutos y 9 segundos (603 etiquetas en segmentos de 3 segundos de duración). Esto es muy pequeño en comparación a la cantidad de datos que se utilizó para validar el modelo en [4]. Luego, también existe la limitación de que los segmentos etiquetados corresponden a dos días consecutivos del año, generando así un sesgo en la precisión obtenida. La última limitación del resultado obtenido de la validación es que los segmentos que se tienen corresponden al *Parc Natural de l'Albufera*, por el contrario, la precisión del 79,1% del algoritmo que se menciona en

[4] se obtiene con segmentos correspondientes a grabaciones realizadas en lugares muy diversos de todo el mundo.

Como conclusión a este apartado se tiene que, a pesar de que una precisión del 59,30% no está mal teniendo en cuenta las limitaciones, no es un resultado concluyente debido a la escasez de segmentos etiquetados por parte de los ornitólogos y al sesgo temporal que hay en estas etiquetas. Por tanto, será necesario obtener un número mayor de muestras etiquetadas correspondientes a diferentes épocas del año para poder obtener resultados más concluyentes.

#### 4.5.4 Lista de especies de *BirdNET*

Otro de los argumentos que se puede especificar a *BirdNET* es una lista de especies personalizada a la que el modelo debe ceñirse al realizar el etiquetado. Esto significa que las especies que etiquetará el modelo en los segmentos serán las que aparezcan en esta lista que consiste en un fichero de texto.

Teniendo en cuenta las especies que querían detectar los ornitólogos, se ha creado un archivo de lista de especies con 190 especies de ave. El objetivo es ver cómo se comporta el modelo cuando se especifican las especies a cuando no se especifican, de forma análoga a como se ha hecho en el apartado 4.5.2. Para hacer esta comprobación, se han etiquetado 9 ficheros de audio utilizando la configuración por defecto, es decir, la que utiliza todas las especies que puede detectar *BirdNET* y se ha hecho otra clasificación especificando la lista de especies.

Tras hacer esta verificación se ha comprobado que, de forma análoga a cuando se cambian los umbrales de confianza, que el modelo que se utiliza para detectar especies es el mismo independientemente del clasificador que se esté utilizando. Las etiquetas obtenidas en la configuración por defecto contienen a las etiquetas que se obtienen utilizando la lista de especies. Por tanto, esta funcionalidad filtra las etiquetas que proporciona el modelo de clasificación eliminando las especies que no están incluidas en la lista de especies.

Cuando se ejecuta el algoritmo, el argumento de la lista de especies se ignora si en la instrucción que ejecuta el modelo se especifican las coordenadas del lugar de grabación. Analizando el código de *BirdNET* se ha encontrado que cuando se especifican las coordenadas, se obtiene una lista de especies que corresponden a las que se pueden avistar en las coordenadas especificadas. Así pues, especificar las coordenadas tiene un funcionamiento análogo al de especificar una lista de especies, el modelo de *BirdNET* seguirá siendo el mismo.

Cuando no se especifica una lista de especies, lo mismo que la configuración por defecto, el modelo tiene en cuenta 6522 especies de ave y cuando se especifican las coordenadas de *l'Albufera* el modelo considera 277 especies. Como la configuración por defecto introduce muchas etiquetas espurias, que corresponden a especies que no están en el *Parc Natural de l'Albufera* en ninguna época del año, se opta por la opción de especificar las coordenadas en el comando de ejecución. En este caso las coordenadas que se han especificado son 39°20'06.2"N 0°21'09.0"W, en la Figura 4.5 se muestra el punto exacto éstas marcado en rojo.



**Figura 4.5** Ubicación de las coordenadas 39°20'06.2"N 0°21'09.0"W

Las 277 especies que considera el modelo tras introducir las coordenadas de *l'Albufera* serán insertadas dentro de la tabla de especies de la base de datos. Estas 277 especies aparecen en la Tabla A.1 del Anexo A.

## 5. Desarrollo

### 5.1 Procesado de las grabaciones

Tras detallar cómo va a estar estructurada la información en la base de datos (sección 4.2), cómo van a ser los segmentos (sección 4.3) y estudiar el funcionamiento de *BirdNET* (sección 4.5) se procede a implementar el procesado que seguirán las grabaciones desde que se descargan hasta que los datos generados se insertan dentro de la aplicación web y la base de datos.

En líneas generales, por cada fichero de 1 hora se obtendrán las etiquetas de *BirdNET*. Para ello se ejecuta en el entorno de Ubuntu la siguiente instrucción en la que se incluyen los argumentos necesarios para configurar *BirdNET*:

```
python3 analyze.py --i INPUTS/ --o CSV/ --rtype 'csv' --min_conf  
0.1 --lat 39.335048 --lon -0.3525 --week -1
```

Con estas etiquetas, se generarán los segmentos de estas grabaciones. A partir de este momento, los procesos se realizan en los segmentos. Luego, se procesarán los segmentos para obtener las instrucciones SQL que permitan rellenar los campos para cada uno de los registros de la tabla *audios* de la base de datos. Después, se procesan los ficheros CSV que contienen las etiquetas de *BirdNET* para obtener las instrucciones SQL que permitan insertar estas etiquetas dentro de la tabla *classification* de la base de datos. Finalmente, se genera una imagen que corresponde al espectrograma en escala Mel para cada uno de los segmentos. Todo el procesado que se especifica en este párrafo se ha implementado en un *script* que se ha guardado como *csv\_processing.py*.

Los segmentos que se obtienen en este algoritmo contienen en el nombre del fichero los campos necesarios para ser insertados en la tabla *audios* de la base de datos. Estos campos son los ya mostrados en la Figura 4.2:

- El nombre de la grabación de origen
- El tiempo de inicio con respecto a la grabación original
- El tiempo de fin del segmento

Con estos campos se forman las consultas SQL que servirán para crear nuevos registros en la base de datos dentro de la tabla *audios*. Estas consultas SQL generadas se almacenan dentro de un fichero llamado *queries.sql*.

A continuación, se muestra un ejemplo de cómo quedaría una instrucción generada con este *script* en la que se insertan 4 nuevas filas en la tabla de *audios* con los datos de 4 segmentos:

```
INSERT INTO audios(wavname, start_time, end_time) VALUES  
("SMA07394_20230520_110002.wav",0,3),  
("SMA07394_20230520_110002.wav", 3, 6),  
("SMA07394_20230520_120002.wav", 51, 54),  
("SMA07394_20230520_120002.wav", 276, 279);
```

Después de insertar los datos asociados a los segmentos en la base de datos, se procede con la inserción de los datos relacionados con las etiquetas que ha obtenido *BirdNET* en el proceso de clasificación. En el fichero CSV que genera el modelo se encuentran los datos necesarios para crear nuevos registros en la tabla *classification*. Cabe recordar que el campo *UserID* correspondiente a *BirdNET* es 1. Con todo esto, se forman las consultas SQL que servirán para insertar las etiquetas en la base de datos y se almacenan dentro del fichero *queries.sql*.

A continuación, se muestra un ejemplo de una instrucción SQL generada con este *script* en la que se insertan 2 etiquetas:

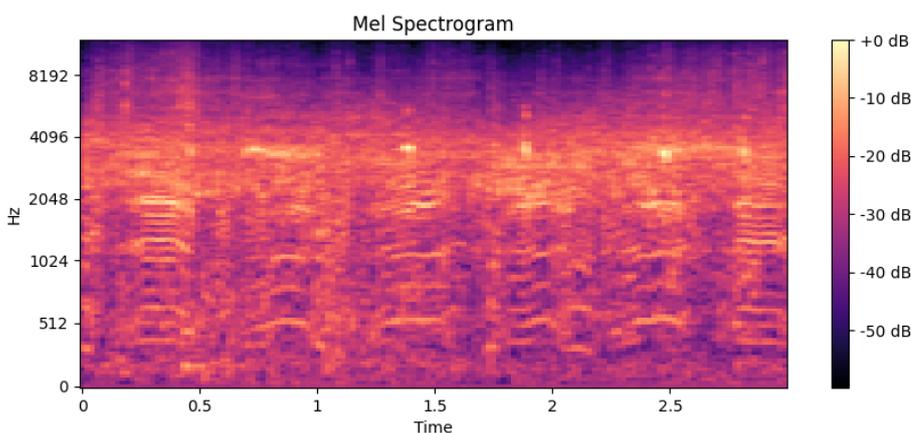
```
INSERT INTO classification(AudioID, SpecieID, UserID, Confidence) VALUES
((SELECT AudioID FROM audios WHERE wavname =
"SMA07394_20230520_110002.wav" AND start_time = 0 AND end_time = 3
LIMIT 1), (SELECT SpecieID FROM species WHERE ScientificName =
"Plegadis falcinellus"), 1, 0.2323),

((SELECT AudioID FROM audios WHERE wavname =
"SMA07394_20230520_110002.wav" AND start_time = 1200 AND end_time =
1203 LIMIT 1), (SELECT SpecieID FROM species WHERE ScientificName =
"Plegadis falcinellus"), 1, 0.9678);
```

Por último, el algoritmo implementado obtendrá una imagen del espectrograma en escala Mel por cada uno de los segmentos de audio que se procese. Para generar estos espectrogramas se ha utilizado la función `librosa.feature.melspectrogram()` de la librería *librosa*. A continuación, se muestra el código utilizado para cargar un segmento y generar un espectrograma de éste:

```
audio_data, sampling_rate = librosa.load(audio_file_path, sr=None)
audio_file_name = audio_file.split('.')[0]
mel_spectrogram = librosa.feature.melspectrogram(y=audio_data, sr=sampling_rate)
mel_spectrogram_db = librosa.power_to_db(mel_spectrogram, ref=np.max)
plt.figure(figsize=(10, 4))
librosa.display.specshow(mel_spectrogram_db, sr=sampling_rate, x_axis='time', y_axis='mel')
plt.colorbar(format='%+2.0f dB')
plt.title('Mel Spectrogram')
plt.savefig(plot_directory + audio_file_name + '_mel_spectrogram.png')
plt.close()
```

De esta manera se obtiene un espectrograma en escala Mel como se puede observar en la Figura 5.1.



**Figura 5.1** Espectrograma en escala Mel del segmento "SMA07394\_20230520\_110002\_0\_3.wav"

Aunque se consideró que la propia aplicación web se encargara de obtener el espectrograma en directo a medida que se iban etiquetando cada uno de los segmentos, esta implementación quedó descartada. La razón principal es que no se pudo ejecutar la librería *librosa* a través de PHP, incluso utilizando los entornos virtuales de Python. La generación de espectrogramas a medida que se iban etiquetando segmentos habría ayudado a economizar espacio en la aplicación web, pero dada la imposibilidad de generarlos en vivo, se procede a generarlos en local y almacenarlos en el servidor web a la vez que se guardan los segmentos.

Una vez se lanza el procesado de *BirdNET* y luego el *script csv\_processing.py*, el último paso a realizar es subir los archivos al servidor web. En este caso, habría que subir los segmentos y los espectrogramas generados. Asimismo, las instrucciones SQL que se almacenaron en el fichero *queries.sql* se ejecutan en la base de datos para insertar nuevos registros en las tablas correspondientes.

En la Figura 5.2 se muestra un diagrama que describe procesado de las grabaciones en local que se ha estado definiendo en esta sección de la memoria.

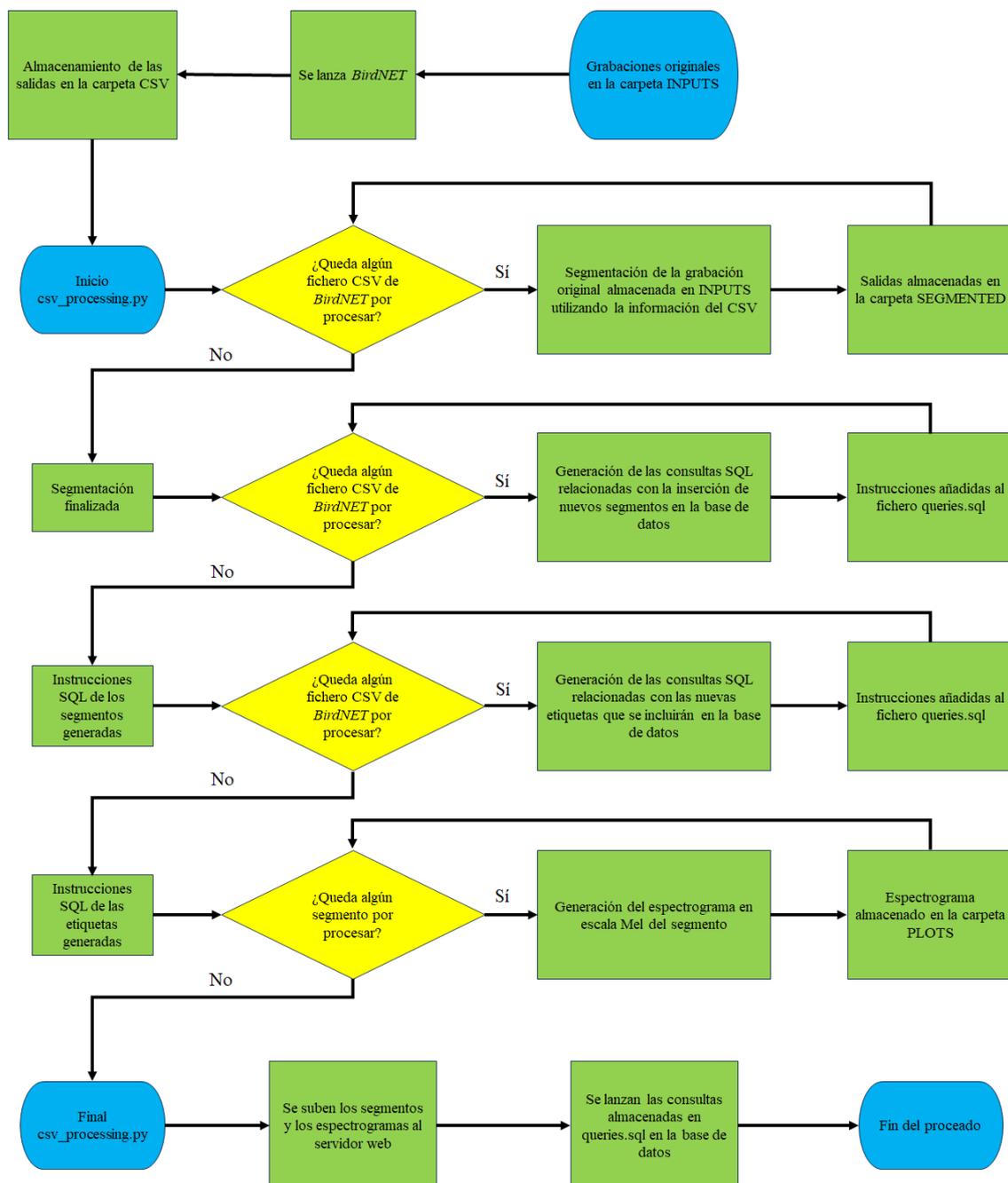


Figura 5.2 Descripción del procesado que se realiza en local

## 5.2 Desarrollo de la web en local

Después de haber implementado el procesado de las grabaciones que se realizará en local (sección 5.1) y se decidido que la herramienta consistirá en una aplicación web (sección 4.4), es momento de desarrollar la aplicación web.

Durante la fase de desarrollo, tanto la página web y la base de datos se implementarán en un entorno local antes de su despliegue en un servidor web con conexión a Internet. Como primera opción, se optó por utilizar *Docker* [27] para alojar tanto el servidor web Apache como para alojar la base de datos *MySQL*. *Docker* es un sistema que permite desarrollar aplicaciones dentro de lo que se conoce como contenedores que se ejecutan en entornos de Linux, evitando que se tenga que crear una máquina virtual para cada aplicación que se pretenda desarrollar. Su principal punto fuerte es que cualquier aplicación que esté alojada en un contenedor podrá funcionar en cualquier entorno compatible con *Docker*. Asimismo, *Docker* permite que los contenedores puedan conservar las versiones de los programas o aplicaciones que se han utilizado, pudiendo dar facilidades en la compatibilidad y evitando problemas de dependencias [28]. Aun así, las desventajas con las que cuenta *Docker* son que requiere de un esfuerzo inicial alto para aprender a utilizarlo y la necesidad de realizar modificaciones en el contenedor a medida que se avance con la implementación.

Finalmente, la opción de utilizar contenedores de *Docker* para implementar la herramienta quedó descartada después de una toma de contacto con *Docker* y después de realizar diferentes pruebas debido a la complejidad añadida de implementar los contenedores que alojen tanto el servidor Apache como el servidor *MySQL*. Además, no es necesaria esta dificultad añadida teniendo en cuenta que existen soluciones más sencillas de utilizar.

Como alternativa se ha utilizado XAMPP. XAMPP es un entorno que lleva integrado un servidor web Apache, PHP, MariaDB y Perl [29]. El uso de este programa ha facilitado mucho el desarrollo de la herramienta, ya que, permite tener tanto el servidor web (servidor Apache) como la base de datos en un mismo entorno, pudiendo interconectarlos de una manera sencilla. Además, XAMPP también incluye el programa *phpMyAdmin* que permite gestionar la base de datos a través de una interfaz web [30]. En la Figura 5.3 se muestra una captura del panel de control de XAMPP. Para acceder a la web que se está implementando en el servidor Apache se debe de introducir, en cualquier navegador, la dirección <http://localhost/>. Para acceder al gestor de bases de datos *phpMyAdmin* se hace introduciendo la dirección <http://localhost/phpmyadmin/> en el navegador o pulsando el botón *Admin* del panel de control de XAMPP en la parte del módulo *MySQL*.

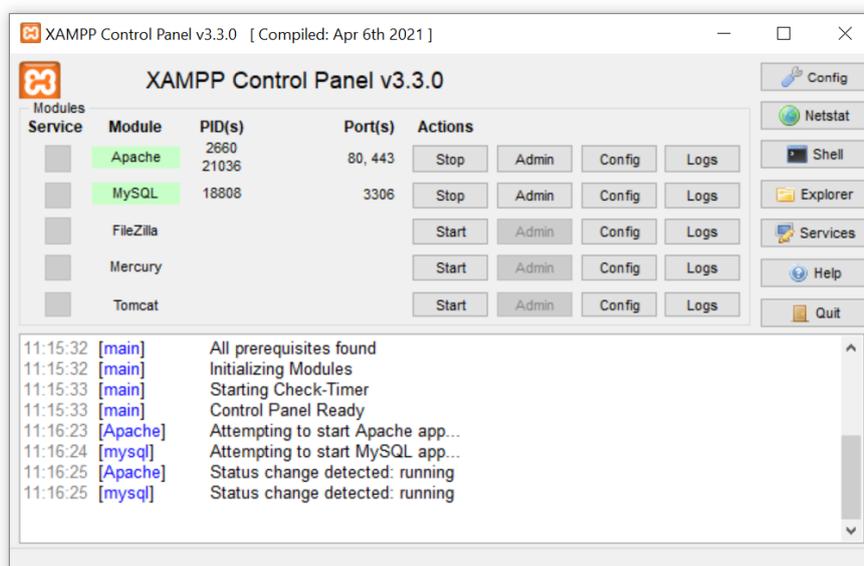


Figura 5.3 Captura del panel de control de XAMPP con los módulos Apache y MySQL en marcha

### 5.2.1 Inicio de sesión

Con el entorno local que permitirá visualizar cómo va quedando la web a medida que se va implementando, se procede a implementar la página de inicio de sesión que tendrá la herramienta. Esta página será la primera a la que accederá el usuario al abrir la herramienta.

Como no se desea que cualquier persona pueda tener acceso a la herramienta, la página principal consistirá en una página de inicio de sesión que servirá para autenticar a un usuario. Es fundamental tener en cuenta la seguridad al implementar esta funcionalidad. Para ello, se ha programado en PHP un mecanismo que evita la inyección de código SQL en el formulario de inicio de sesión, lo que garantiza la integridad de la información que se almacena en la base de datos e imposibilita el acceso de usuarios malintencionados que no posean las credenciales necesarias. Esto se ha hecho mediante el uso de sentencias preparadas en las consultas SQL.

Por otro lado, las contraseñas de los usuarios no se almacenarán en la base de datos en texto plano ni serán enviadas en este formato. En este caso, las contraseñas se han encriptado utilizando una función *hash* diseñada para contraseñas llamada *bcrypt* que puede ser utilizada en PHP.

Asimismo, cada página de la aplicación web detecta si el usuario ha iniciado sesión previamente, si no lo ha hecho, lo redirecciona automáticamente a la página de inicio de sesión. Esta restricción impide que el usuario pueda tener acceso a cualquier otra parte de la web si no ha iniciado sesión previamente.

En caso de que se produzca un error de *MySQL*, éste se maneja mostrando un mensaje de error genérico al usuario. Este mensaje genérico sirve para que un usuario malintencionado no pueda conocer qué causó el error y aprovecharlo en su beneficio.

El último mecanismo de seguridad que se ha implementado es un mecanismo que limita los intentos de inicio de sesión para evitar ataques de fuerza bruta. Para ello, se registra la dirección IP del visitante de la web y se registra el número de intentos que ha hecho esa dirección en 24 horas. Si en 24 horas se registran más 5 intentos fallidos se introduce un retraso de 1 minuto por cada intento nuevo que se realice. Todos estos parámetros son configurables.

En este proceso de autenticación, se interactúa con la base de datos para obtener la información relacionada al usuario. Para interactuar con la base de datos se utiliza la extensión *MySQLi* de PHP, lo que asegura una gestión eficiente y segura de la información. También se consideró utilizar el controlador *PDO\_MYSQL*, pero esta opción quedó descartada debido a que ha sido más sencillo hacerlo con *MySQLi*.

Con lo que respecta el aspecto de la página, está compuesta de un formulario de inicio de sesión sencillo donde el usuario tiene que especificar su nombre de usuario y su contraseña. Tras completar los campos y pulsar el botón “Acceder”, los datos de inicio de sesión procesados se envían al servidor a través del método POST. Internamente, el servidor verifica que la contraseña proporcionada en el formulario sea correcta teniendo como referencia la contraseña almacenada en la base de datos. Si la contraseña es correcta, se crea una sesión para el usuario. Por último, el usuario es redireccionado a la página principal, llamada *home.php*.

Esta página de inicio de sesión se ha implementado en el fichero *login.php*. En la Figura 5.4, se muestra el aspecto que tiene.

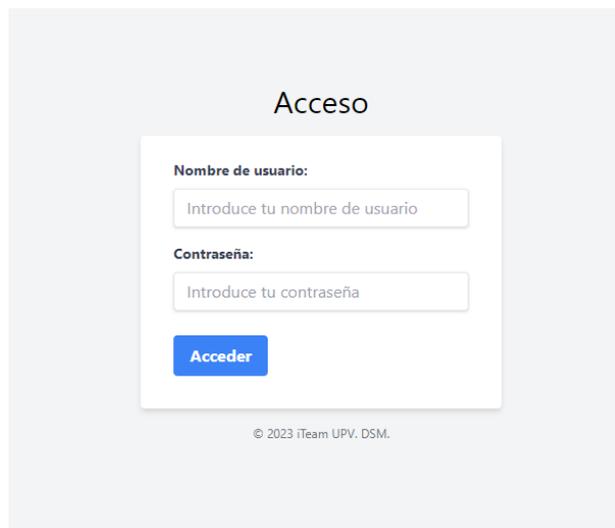


Figura 5.4 Aspecto de *login.php*

### 5.2.2 *Página principal*

La página principal se compone principalmente de 3 partes: La herramienta de etiquetado, un menú que permite acceder a páginas secundarias y un botón para cerrar sesión. La parte de la herramienta de etiquetado será la que tendrá más relevancia. En la Figura 5.5 se muestra el aspecto de la página principal.

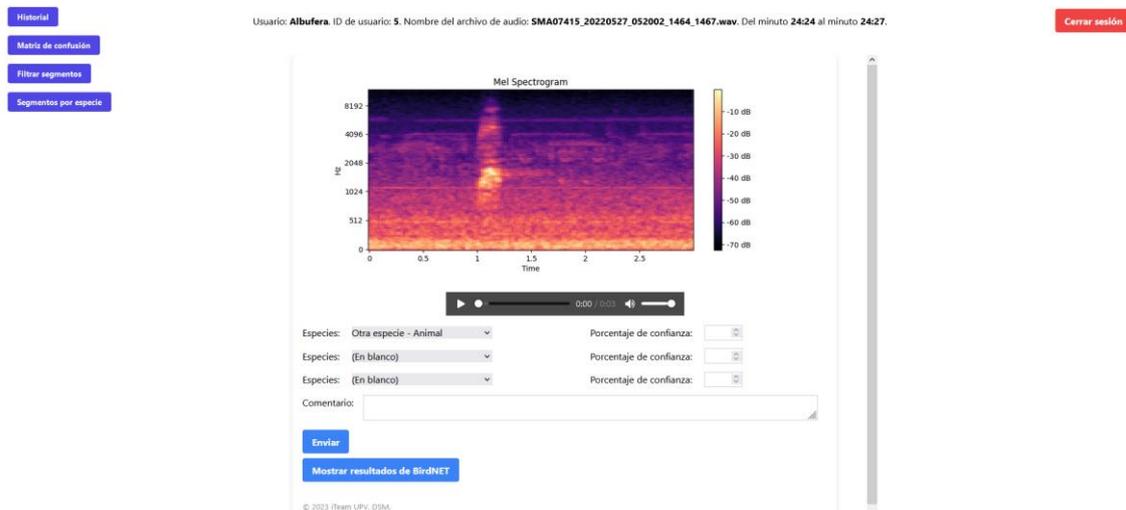


Figura 5.5 Aspecto de *home.php*

La herramienta de etiquetado ocupa la parte central de la página web. En la parte superior de la herramienta el usuario visualizará la información relacionada con el último segmento que tenga pendiente de clasificar, mostrando: el nombre de usuario, el identificador de usuario, nombre del fichero del segmento, minuto en el que inicia el segmento con respecto a la grabación original y minuto en el que termina el segmento dentro de la grabación original. Para obtener la lista de segmentos pendientes por clasificar, se ejecuta una instrucción SQL que obtiene los segmentos a los que el usuario no ha dado aún ninguna etiqueta, ordena estos segmentos de manera descendiente en función del valor de confianza en la que *BirdNET* los ha etiquetado y se limita la lista sólo al primer valor, el segmento que se muestra al usuario en la web. En este caso, se ha tomado la decisión de ordenar los segmentos de mayor a menor confianza según el etiquetado de *BirdNET*. De cara a validar cómo de bueno es el modelo de *BirdNET*, las muestras con un mayor

valor de confianza tendrán más valor en esta validación que las etiquetas que tengan un menor valor de confianza. Dicho de otra manera, tiene más sentido considerar las etiquetas en las que *BirdNET* ha obtenido valores de confianza mayores. A continuación, se muestra la consulta SQL que se utiliza para obtener el segmento pendiente de etiquetar que aparece en la página principal, en este caso, se ha escrito la consulta para un usuario cuyo identificador de usuario es el número 5:

```
SELECT audios.wavname, audios.start_time, audios.end_time,  
classification.Confidence FROM audios INNER JOIN classification  
ON audios.AudioID = classification.AudioID WHERE  
classification.UserID = 1 AND classification.AudioID NOT IN  
(SELECT AudioID from classification WHERE UserID = 5) order by  
classification.Confidence desc;
```

Debajo de los datos relacionados al segmento y al usuario, se muestra el espectrograma en escala Mel del segmento seguido de un reproductor de audio para escuchar el segmento, ambos obtenidos siguiendo el proceso detallado en la sección 5.1. El reproductor tendrá pequeñas variaciones en su aspecto según el navegador web que se esté utilizando. Esto último no afecta en su funcionamiento.

Posteriormente se muestra el formulario que permite añadir hasta tres etiquetas y un comentario para el segmento pendiente de etiquetar. Las etiquetas de especies vienen en un desplegable ordenado alfabéticamente. Estas etiquetas son el nombre científico de la especie. A la derecha de los desplegables, el usuario puede concretar de forma numérica, utilizando una escala del 0 al 100, cómo de seguro está de que la especie que está etiquetando está presente en la grabación. Tanto la etiqueta como su confianza son valores obligatorios en la primera etiqueta, lo que significa que no puede ser enviada sin especificar un valor de confianza o sin especificar a qué especie se refiere. El comentario es un campo de texto en el que se pueden escribir las observaciones relevantes que se deseen incluir. Una vez se rellenan estos valores, el formulario se envía con el botón “Enviar”. Esto es lo mismo que enviar al servidor la etiqueta o las etiquetas del segmento.

Cuando se pulsa el botón “Enviar” para etiquetar un segmento, los datos del formulario se envían al servidor mediante el método POST. Internamente, el servidor procesa los datos enviados a través del formulario. Primero divide el valor de confianza especificado por el usuario entre 100 para que quede en una escala del 0 a 1. Luego se ejecuta la instrucción que inserta la primera etiqueta en la base de datos. Con lo que respecta a los campos del formulario relacionados con la segunda y la tercera etiqueta, se realizan una serie de verificaciones antes de insertar o no estas etiquetas. Estas comprobaciones consisten en verificar que el campo de especie no sea “(En blanco)”, que la segunda o tercera etiqueta no sean igual que la anterior en el campo de especie de ave y que el campo de confianza no esté vacío. Si se cumplen estas condiciones, se lanza una instrucción que inserte la etiqueta, si no, no se inserta. Finalmente, cuando este procesado ya ha terminado, se recarga automáticamente la página principal. Al recargar la página principal, se vuelve a ejecutar la consulta que obtiene los segmentos pendientes de etiquetar y el segmento que se acaba de etiquetar ya no figura en este conjunto. A continuación, se muestra un ejemplo de cómo quedaría una consulta SQL utilizada para insertar una etiqueta proporcionada por un usuario:

```
INSERT INTO classification (AudioID, SpecieID, UserID,  
Confidence, Comments) VALUES ((SELECT AudioID FROM audios WHERE  
wavname = "SMA07415_20220527_072002.wav" AND start_time = 15 AND  
end_time = 18 LIMIT 1), (SELECT SpecieID FROM species WHERE  
ScientificName = "Locustella luscinioides" LIMIT 1), (SELECT  
UserID FROM usuarios WHERE Username = "Daniel" LIMIT 1), 0.95,  
"Comentario sobre la etiqueta");
```

Más abajo del formulario se encuentra un botón que permite mostrar las etiquetas que ha dado *BirdNET* al segmento pendiente de etiquetar por el usuario. Se ha decidido que esta información quede oculta de forma predeterminada al usuario para evitar introducir sesgos en el proceso de etiquetado. Aun así, una vez el usuario ya tenga claras las etiquetas que quiere dar al segmento, puede pulsar este botón para descubrir qué predicciones ha hecho *BirdNET* y con qué valores de confianza las ha hecho. Esto también puede permitir obtener una retroalimentación por parte del usuario sobre cómo de bueno es el modelo clasificando especies de ave. En la Figura 5.6, se observa cómo queda esa parte de la página tras pulsar este botón que permite mostrar las etiquetas que ha dado *BirdNET*. A continuación, se muestra la consulta que se ha utilizado para esto:

```
SELECT species.ScientificName, classification.Confidence FROM
classification INNER JOIN species ON species.SpecieID =
classification.SpecieID WHERE classification.UserID = 1 AND
classification.AudioID IN (SELECT AudioID FROM audios WHERE
wavname = "SMA07415_20220527_072002.wav" AND start_time = 15 AND
end_time= 18);
```

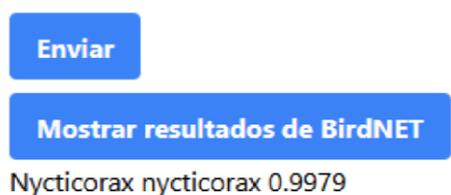


Figura 5.6 Aspecto tras pulsar el botón “Mostrar resultados de BirdNET”.

Otra de las partes por la que está compuesta la página principal es el menú de las páginas secundarias. Las opciones que se presentan en este menú difieren según si el usuario autenticado es un usuario administrador o si es un usuario normal. Los administradores tienen acceso a seis páginas secundarias mientras que los usuarios normales tienen acceso a cuatro. Las páginas a las que tienen acceso todos los usuarios son “Historial” (sección 5.2.3), “Matriz de confusión” (sección 5.2.5), “Filtrar segmentos” () y “Segmentos por especie” (). Las dos páginas que podrán acceder los administradores son “Registrar usuario” (sección 5.2.8) y “Top confianza *BirdNET*” (sección 5.2.9). En la Figura 5.7 se muestra el aspecto que tiene el menú para un usuario normal y el aspecto que tiene para un usuario administrador.



Figura 5.7 En la parte izquierda: Menú usuario normal. En la parte derecha: Menú usuario administrador.

Aunque no exista una variable que indique qué usuarios son administradores y qué usuarios no lo son dentro de la base de datos, dentro del código PHP sí que hay implementado un mecanismo que comprueba si el usuario que inició sesión es administrador o es un usuario normal.

La última parte que compone la página es un botón que se encuentra en la parte superior derecha cuya utilidad es la de cerrar la sesión del usuario. Al pulsar ese botón se carga el *script*

logout.php que se encarga de eliminar las variables de sesión, eliminar la sesión y eliminar la cookie del usuario en caso de que se hubiese establecido. Tras todo esto se redirige al usuario a la página de inicio de sesión.

En las siguientes secciones se explican con más detalle las páginas secundarias que existen y qué aspecto presentan.

### 5.2.3 Historial

La primera página secundaria que se encuentra al usuario es a la que se accede pulsando el botón “Historial” de la página principal. En esta página se ha implementado un historial donde el usuario puede visualizar las últimas etiquetas que dio a los segmentos. Esta página se ha implementado sobre el fichero `historial.php` y en la Figura 5.8 se muestra el aspecto que tiene.

Historial. Usuario: Daniel. ID de usuario: 10.

[Volver a Inicio](#)

Nombre del archivo	Tiempo de inicio	Tiempo de fin	Especie	Confianza	Comentarios	
SMA07415_20220527_072002.wav	03:21	03:24	Otra especie - Ave	50 %	Comentario 3	<a href="#">Editar</a>
SMA07415_20220527_072002.wav	03:21	03:24	Himantopus himantopus	100 %	Comentario 3	<a href="#">Editar</a>
SMA07394_20230522_060000.wav	07:15	07:18	Otra especie - Animal	78 %		<a href="#">Editar</a>
SMA07415_20220527_052002.wav	41:24	41:27	Otra especie - Animal	78 %		<a href="#">Editar</a>
SMA07415_20220527_072002.wav	06:30	06:33	Otra especie - Animal	23 %		<a href="#">Editar</a>
SMA07415_20220527_052002.wav	22:24	22:27	Nycticorax nycticorax	100 %	Canto lejano. Se escucha alguna especie más de fondo	<a href="#">Editar</a>
SMA07415_20220527_072002.wav	07:24	07:27	Otra especie - Animal	0 %		<a href="#">Editar</a>
SMA07415_20220527_072002.wav	00:09	00:12	Otra especie - Animal	3 %		<a href="#">Editar</a>
SMA07415_20220526_230002.wav	50:30	50:33	Otra especie - Animal	0 %		<a href="#">Editar</a>
SMA07415_20220527_062002.wav	58:18	58:21	Locustella luscinioides	100 %	Comentario 2	<a href="#">Editar</a>
SMA07415_20220527_062002.wav	14:33	14:36	Locustella luscinioides	90 %		<a href="#">Editar</a>
SMA07415_20220527_062002.wav	58:15	58:18	Locustella luscinioides	100 %		<a href="#">Editar</a>
SMA07415_20220527_052002.wav	24:21	24:24	Nycticorax nycticorax	100 %	Comentario 1	<a href="#">Editar</a>
SMA07415_20220527_062002.wav	13:51	13:54	Locustella luscinioides	100 %		<a href="#">Editar</a>
SMA07415_20220527_052002.wav	39:54	39:57	Locustella luscinioides	99 %		<a href="#">Editar</a>

Figura 5.8 Aspecto de `historial.php`

En la esquina superior izquierda se muestra el nombre del usuario y su identificador. En la esquina superior derecha se ha puesto un botón (“Volver a Inicio”) que sirve para volver a la página principal (`home.php`).

En el resto de esta página se encuentra el elemento principal de esta página que consiste en una tabla que recopila las últimas etiquetas que el usuario ha puesto a los segmentos. Estas etiquetas van ordenadas de la más reciente a la más antigua. Cada fila equivale a una etiqueta. De esta manera, si a un mismo segmento se le puso más de una etiqueta, éstas aparecerán en diferentes filas. Cada columna representa: el nombre de la grabación original de la que procede el segmento etiquetado, el tiempo de inicio en formato de minutos y segundos del segmento etiquetado dentro de la grabación del nodo, el tiempo en el que acaba el segmento en formato de minutos y segundos del segmento etiquetado dentro de la grabación original, la especie que etiquetó el usuario en esa muestra, la confianza con la que el usuario hizo la etiqueta y el comentario que añadió el usuario a la etiqueta. Una columna más a la derecha de las anteriores se encuentra, para cada fila, el botón “Editar” que llevará al usuario a una página que servirá para editar cada una de las etiquetas correspondientes. Esto servirá en caso de que el usuario desee modificar algún campo relacionado con la etiqueta bien sea porque haya cambiado de opinión, porque desee incluir algún comentario más al respecto o haya puesto una etiqueta por error. El aspecto y las funcionalidades de la página de edición de registro se explican en la sección 5.2.4.

A continuación, se muestra la consulta que se realiza a la base de datos para obtener el historial de etiquetas correspondiente a un usuario cuyo identificador de usuario es el número 5:

```
SELECT audios.wavname, audios.start_time, audios.end_time,
species.ScientificName, classification.Confidence,
classification.Comments, classification.ID FROM ((audios INNER
JOIN classification ON audios.AudioID = classification.AudioID)
INNER JOIN species ON species.SpecieID = classification.SpecieID)
WHERE classification.UserID = 5 ORDER BY classification.ID DESC;
```

#### 5.2.4 Edición de registro

Al elegir la opción de editar una etiqueta dentro de la página del historial, se redirige al usuario a la página `editar_registro.php` que servirá para editar cualquiera de los campos de la etiqueta. Cuando se redirige al usuario a esta página, se envía mediante el método POST el identificador de la etiqueta que se desea modificar. Como se puede ver en la Figura 5.9, esta página tiene un aspecto similar a `home.php` pero con algunas modificaciones:

- El menú de la izquierda está formado solamente por el botón “Historial”, que permite al usuario volver a la página del historial de etiquetas.
- Como se trata de una página que sirve para modificar una etiqueta, sólo hay un campo para elegir la especie y un campo para elegir el valor de confianza.
- Al pulsar el botón enviar se modifican los campos de la etiqueta en lugar de insertar un nuevo valor en la base de datos y se redirige al usuario a la página de historial. Como la base de datos no admite etiquetas duplicadas para un mismo segmento, en caso de intentar insertar una etiqueta que ya exista saltará un error y al usuario le aparecerá una alerta indicándole que para ese segmento ya existe un registro con el nombre de esa especie.

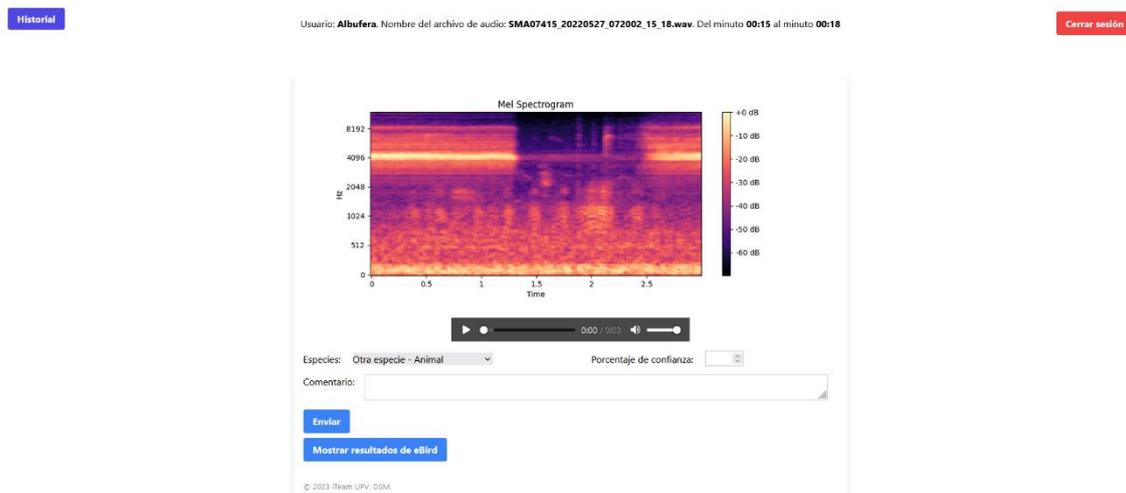


Figura 5.9 Aspecto de `editar_registro.php`

A continuación, se muestra un ejemplo de instrucción que se ejecuta en la base de datos cuando se modifica una etiqueta:

```
UPDATE classification SET SpecieID = (SELECT SpecieID FROM species
WHERE ScientificName = "Locustella luscinioides" LIMIT 1),
Confidence = 1, Comments = "Comentario" WHERE ID = 33850;
```

#### 5.2.5 Matriz de confusión

La segunda página secundaria que se ha implementado es una matriz de confusión que permite al usuario ver los aciertos y los fallos que ha tenido *BirdNET* clasificando segmentos en comparación a las etiquetas proporcionadas por el usuario. Para cada usuario se genera una matriz

de confusión. Esta página ha sido implementada en el fichero `confusion_matrix.php`. Los valores que se leen en vertical corresponden a las etiquetas que ha proporcionado *BirdNET* y los que se leen en horizontal corresponden a las etiquetas proporcionadas por los usuarios. Con esto, se pueden comparar las etiquetas de especies puestas por el modelo y por el usuario. En la Figura 5.10 se muestra el aspecto que tendrá esta página. La matriz consiste en una tabla donde aparecen el número de segmentos que tienen la misma pareja de etiquetas proporcionadas por el usuario y el modelo. En diagonal se han puesto en negrita el número de veces en las que *BirdNET* y el usuario han dado con la misma etiqueta para el mismo segmento, lo que sería el número de aciertos de *BirdNET*. Todos los valores que salen de la diagonal son el número de veces que el modelo no proporcionó una etiqueta correcta. A continuación, se muestra un ejemplo de instrucción que se utiliza para obtener los datos de la matriz de confusión:

```
SELECT S1.ScientificName AS PredictedClass, S2.ScientificName AS ActualClass FROM classification T1 JOIN classification T2 ON T1.AudioID = T2.AudioID JOIN species S1 ON S1.SpecieID = T1.SpecieID JOIN species S2 ON S2.SpecieID = T2.SpecieID WHERE T1.UserID = 1 AND T2.UserID = 10;
```

#### Matriz de confusión

[Volver a Inicio](#)

Usuario: Daniel. ID de usuario: 10.

		Etiquetas BirdNET									
		Otros	Locustella naevia	Otra especie - Ave	Rallus aquaticus	Otra especie - Animal	Nycticorax nycticorax	Locustella luscinoides	Riparia riparia	Himantopus himantopus	Plegadis falcinellus
Tus etiquetas	Otros	0	2	0	0	0	0	0	0	0	0
	Locustella naevia	0	0	0	0	0	0	0	0	0	0
	Otra especie - Ave	0	0	0	1	0	0	0	0	1	0
	Rallus aquaticus	0	0	0	0	0	0	0	0	0	0
	Otra especie - Animal	0	2	0	0	0	1	4	0	0	1
	Nycticorax nycticorax	0	0	0	0	0	3	0	0	0	0
	Locustella luscinoides	0	0	0	0	0	0	7	0	0	0
	Riparia riparia	0	0	0	0	0	0	0	1	0	0
	Himantopus himantopus	0	0	0	0	0	0	0	0	1	0
	Plegadis falcinellus	0	0	0	0	0	0	0	0	0	0

Figura 5.10 Aspecto de `confusion_matrix.php`

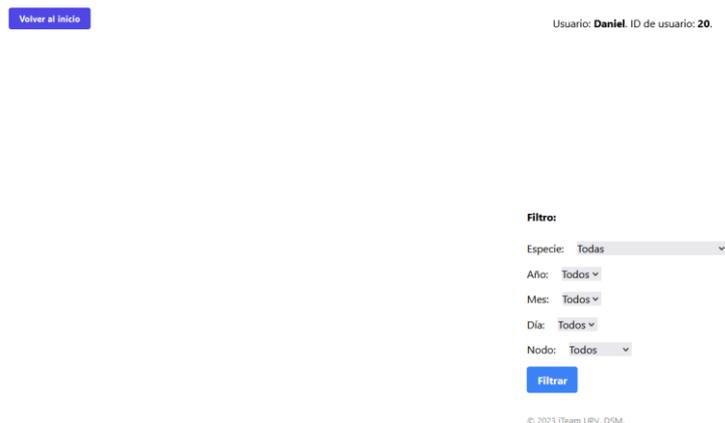
Por último, el botón de la esquina superior derecha de esta página servirá para volver a la página de inicio.

### 5.2.6 Filtrado de segmentos

Una de las funcionalidades sugeridas por el usuario es el etiquetado selectivo. Esta funcionalidad permite al usuario la posibilidad de filtrar, en base a unos parámetros, qué segmentos desea escuchar y etiquetar. En este caso podrá elegir la fecha exacta en la que tuvo lugar la grabación original, la especie que ha detectado *BirdNET* y el nodo que ha realizado esa grabación. Esta funcionalidad se conoce como filtrado de segmentos y permite a los ornitólogos, por ejemplo, enfocarse en muestras de sonido de alguna especie escasa en el *Parc Natural de l'Albufera*. Este filtrado de segmentos ha sido implementado en las páginas `filtro_segmentos.php` y `filtro_segmentos2.php`.

La primera página (`filtro_segmentos.php`) servirá para asignar los valores del filtro. En este caso, el filtro será un desplegable donde estarán disponibles sólo los valores que ya estén dentro de la base de datos. Por ejemplo, en el campo de especies sólo se podrán seleccionar las especies que haya detectado *BirdNET* o en el campo de año sólo se podrán poner los años en los que hay grabaciones, 2022 y 2023 en este caso.

Como se ve en la Figura 5.11, esta primera página se compone de un botón que sirve para volver a la página inicial. Luego se muestra el nombre del usuario y su identificador. En la parte central de la página se encuentran los filtros que se pueden seleccionar, en este caso, se filtrará por especie detectada en *BirdNET*, el nodo que realizó la grabación, y el año, mes y día de la grabación original. Al seleccionar la opción de “Filtrar”, se realiza una consulta a la base de datos para obtener los segmentos que cumplen las características que el usuario desea. En caso de que no exista ningún segmento que cumpla con las características que se han elegido, se redirige al usuario a la página de selección de filtros para que vuelva a elegir otros valores. En el caso de encontrar uno o más segmentos, se redirige al usuario a la siguiente página (`filtro_segmentos2.php`).



**Figura 5.11** Aspecto de `filtro_segmentos.php`

Con los filtros ya establecidos y en caso de que existan segmentos, el usuario visualiza la página `filtro_segmentos2.php`. Como se puede observar en la Figura 5.12, esta segunda página es similar a la página principal (`home.php`) pero con algunas modificaciones:

- El menú de botones de la parte superior izquierda está formado solamente por el botón “Volver a filtro”, que permite al usuario volver a la página de selección de filtro.
- En la parte izquierda se detallan qué filtros hay aplicados en ese momento. También se muestra, más abajo en negrita, la cantidad de muestras que ha encontrado con esas características. Esto es útil para estimar cuántos especímenes de una especie se podrían haber avistado en un lugar durante una temporada del año concreta.

Una vez se etiqueta el segmento y se envía, de forma análoga a la página de inicio, se almacena la etiqueta en la base de datos y se recarga la página con el siguiente segmento que cumple las características del filtro. Una vez se etiqueta el último segmento disponible para etiquetar que cumpla con las características del filtro, se redirige automáticamente al usuario a la página de selección de filtros (`filtro_segmentos.php`).

A continuación, se muestra un ejemplo de instrucción que se ha utilizado para obtener los segmentos filtrados:

```
SELECT audios.wavname, audios.start_time, audios.end_time,
classification.Confidence FROM audios INNER JOIN classification
ON audios.AudioID = classification.AudioID WHERE
classification.UserID = 1 AND classification.AudioID NOT IN
(SELECT AudioID from classification WHERE UserID = 5) AND
classification.SpecieID IN (SELECT SpecieID from species WHERE
```

```
ScientificName = "Acrocephalus arundinaceus") AND
LEFT(SUBSTRING_INDEX(SUBSTRING_INDEX(audios.wavname, '_', 2),
'_', -1), 4) = "2022" AND
RIGHT(LEFT(SUBSTRING_INDEX(SUBSTRING_INDEX(audios.wavname, '_', 2),
'_', -1), 6), 2) = "05" AND
RIGHT(LEFT(SUBSTRING_INDEX(SUBSTRING_INDEX(audios.wavname, '_', 2),
'_', -1), 8), 2) = "27" AND SUBSTRING_INDEX(audios.wavname,
'_', 1) = "SMA07415" ORDER BY classification.Confidence DESC;
```

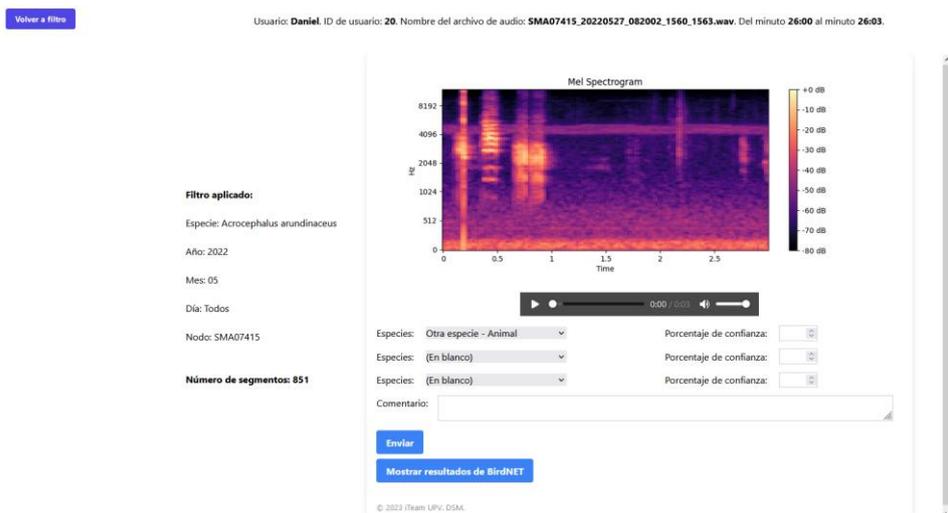


Figura 5.12 Aspecto de *filtro\_segmentos2.php*

### 5.2.7 Segmentos por especie

La cuarta página secundaria que visualizará el usuario será una lista del número de muestras por especie según las etiquetas de *BirdNET*. La lista de segmentos que hay por especie se encuentra en la página *segmentos\_por\_especie.php* y en la Figura 5.13 se muestra el aspecto que presenta.

Especie	Número de segmentos
Plegadis falcinellus	3585
Locustella luscinioides	1512
Acrocephalus arundinaceus	851
Bubulcus ibis	577
Acrocephalus scirpaceus	303
Botaurus stellaris	205
Platalea leucorodia	193
Nycticorax nycticorax	189
Anas platyrhynchos	158
Himantopus himantopus	115
Ardea alba	109
Egretta garzetta	90
Chroicocephalus ridibundus	76
Gallinula chloropus	58
Cyanopica cooki	54
Phoenicopterus roseus	52
Ardea cinerea	46
Phalacrocorax carbo	41
Porphyrio porphyrio	27
Acrocephalus melanopogon	24

Figura 5.13 Aspecto de *segmentos\_por\_especie.php*

A continuación, se muestra la instrucción que se ha utilizado para obtener la cantidad de segmentos que existen por especie según el etiquetado de *BirdNET*:

```
SELECT species.ScientificName AS sn, COUNT(classification.ID) AS
N_Species FROM classification LEFT JOIN species ON
classification.SpecieID = species.SpecieID WHERE
classification.UserID = 1 GROUP BY classification.SpecieID ORDER
BY N_Species DESC;
```

### 5.2.8 Registro de nuevos usuarios (Administrador)

La idea inicial fue la de implementar el registro de usuarios dentro de la página de inicio de sesión. Esta opción no se consideró, dado a que, en este proyecto, es necesario tener un control sobre los usuarios que hay registrados. El hecho de que un usuario pueda registrarse libremente en la aplicación puede introducir valores espurios en las etiquetas que clasifique, alterando así cualquier análisis posterior de las etiquetas que se desea realizar. Por ello, se tomó la decisión de que los administradores de la herramienta sean los únicos que puedan registrar usuarios.

Con lo que respecta a esta página de registro de usuarios, ésta se compone de un formulario que sirve para registrar a un nuevo usuario y un botón para volver a la página principal. En el formulario se pide un nombre de usuario que servirá para autenticarse en la web, un correo electrónico de contacto con el usuario, el nombre del usuario, su apellido y una contraseña. Al pulsar el botón “Regístrate” se registra un nuevo usuario. En la Figura 5.14 se muestra el aspecto que tiene la página de registro de usuarios. Esta página está implementada en el fichero `registration.php`.

Al registrar un nuevo usuario, se envían los datos del formulario a través del método POST y se dirige al administrador a la página implementada `register.php`. En esta página, ejecutan una serie de comandos que almacenan en el servidor los datos del nuevo usuario y se muestra un mensaje avisando que el usuario se registró con éxito. Debajo de este mensaje, hay un enlace que permite al administrador volver a la página de registro de usuarios en caso de que se desee registrar otro usuario más. En la Figura 5.15 se muestra el aspecto que tiene esta página.

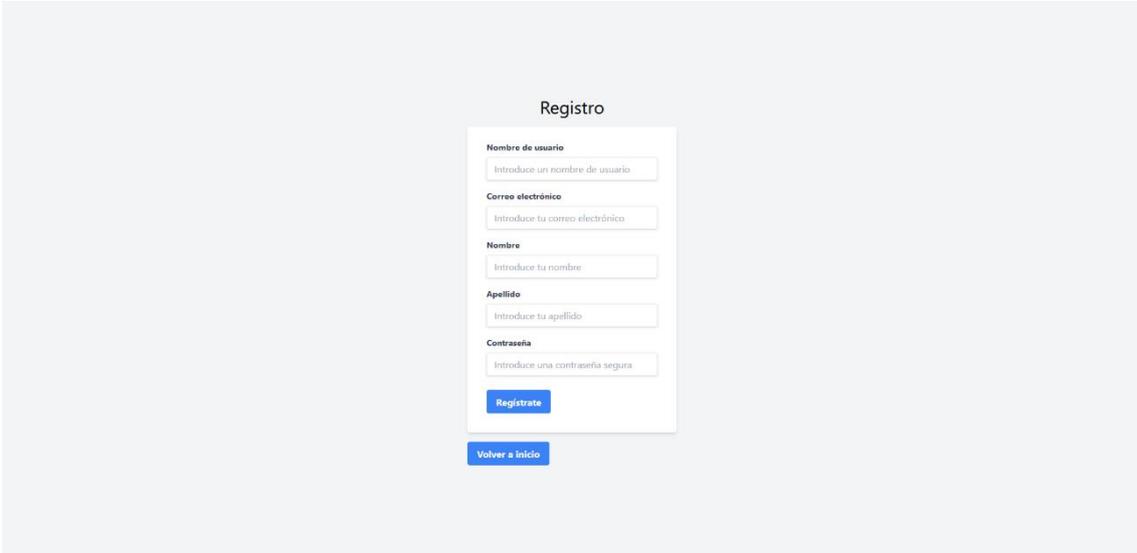


Figura 5.14 Aspecto de `registration.php`

El usuario fue registrado exitosamente  
[Volver al registro de usuarios](#)

Figura 5.15 Aspecto de `register.php`

### 5.2.9 Top confianza BirdNET (Administrador)

La última página secundaria consiste en una lista que muestra en orden descendente la media de confianza que obtiene *BirdNET* por especie, dicho de otra manera, muestra en qué especies ha

tenido más confianza el modelo. La motivación de que esta página sólo esté disponible para administradores es que no se desea introducir sesgos en el usuario acerca del funcionamiento de *BirdNET*. En la Figura 5.16 se observa el aspecto que presenta esta página.

**Top confianza media de BirdNET** Volver a Inicio

Usuario: Daniel. ID de usuario: 10.

Especie	Confianza
Larus michahellis	0.8300999999046326
Charadrius dubius	0.7461000084877014
Cettia cetti	0.7445000211397806
Tadorna tadorna	0.6274999976158142
Numenius phaeopus	0.6150000095367432
Chroicocephalus genei	0.5946999788284302
Riparia riparia	0.5756624974310398
Ardea cinerea	0.49831304180881253
Acrocephalus arundinaceus	0.48304970628996713
Asio otus	0.4309999942779541
Plegadis falcinellus	0.4150093167500682
Acrocephalus scirpaceus	0.40836534742573305
Locustella luscinioides	0.4082013892216815
Numenius arquata	0.38909998536109924
Fulica atra	0.3717384556165108
Himantopus himantopus	0.3611634782474974
Sternula albifrons	0.35589998960494995
Sterna hirundo	0.35131428869707243
Falco peregrinus	0.3431999981403351

Figura 5.16 Aspecto de *top\_confidence.php*

Esta página secundaria se compone de un título, un subtítulo que indica el nombre de usuario y el identificador de usuario, una tabla que muestra la especie y la confianza media que ha tenido el modelo en todas las etiquetas que ha puesto hasta la fecha, y un botón para volver a la página de inicio. A continuación, se muestra la consulta que se ha utilizado para obtener los datos que se muestran en la tabla:

```
SELECT      species.ScientificName      AS      sn,
AVG(classification.Confidence) AS conf FROM classification LEFT
JOIN species ON classification.SpecieID = species.SpecieID WHERE
classification.UserID = 1 GROUP BY classification.SpecieID ORDER
BY conf DESC;
```

### 5.3 Despliegue de la web

Una vez la herramienta ya está implementada en local y funcionando en local, el siguiente paso ha sido alojarla en una web. La plataforma utilizada para alojar la web ha sido *Plesk* [31]. De todas las posibilidades que ofrece esta plataforma, se ha utilizado una base de datos para alojar la base de datos implementada y el directorio *httpdocs* para alojar la web.

Para alojar la web, se ha subido al directorio *httpdocs* todos los archivos implementados en PHP y se han creado los directorios que contendrán los segmentos y los espectrogramas. Además, se ha implementado el fichero *index.html* que servirá para redireccionar al usuario a la página *login.php* automáticamente cuando introduzca el enlace de la web en el navegador.

Por otro lado, para poner en marcha la base de datos de *Plesk*. El primer paso ha sido exportar la base de datos que se ha implementado en XAMPP. Esto genera un archivo en formato SQL que se importa al servidor *MySQL* de *Plesk*. Con ello, la base de datos ya está en la web de la misma manera que está en local.



## 6. Conclusiones

Después de haber mostrado el proceso de diseño y de implementación de la herramienta, se puede concluir que la herramienta cumple con las especificaciones establecidas por el usuario final. Se ha investigado cómo funciona un modelo entrenado para etiquetar cantos de aves y cómo se configura para que se adapte a las necesidades del proyecto. Además, ha sido necesario realizar un diseño en el que se tiene en cuenta la retroalimentación dada por el usuario final, el que sería un equivalente a un cliente en un entorno profesional. Asimismo, se ha pensado diferentes maneras en las que se puede implementar esta herramienta para escoger la forma de hacerlo que se ha considerado como la más adecuada, en este caso una aplicación web con una conexión a una base de datos.

En cuanto a los resultados de esta herramienta, por la manera en la que está implementada, optimizará con creces el tiempo dedicado al proceso de etiquetado de segmentos de audio. Asimismo, esta herramienta convierte al etiquetado en un proceso más llevadero y ágil que una metodología convencional en la que se tienen que escribir las etiquetas y los minutos en los que aparece la especie de forma manual en un fichero Excel.

En cuanto a la experiencia personal, la totalidad de este proyecto ha servido para enfrentarse al proceso de diseño y desarrollo de una herramienta partiendo de unas especificaciones previas dadas por el usuario. Además, ha sido necesario documentarse, repasar y aprender acerca de las tecnologías que se han utilizado en este trabajo, tanto su funcionamiento como su aplicación en un proceso de desarrollo. Este aprendizaje será de gran utilidad en el futuro.

## 7. Líneas futuras

A partir de este trabajo se pueden obtener distintas vías en las que se puede seguir trabajando en cuanto a características adicionales de la herramienta o ámbitos en los que se puede aplicar su uso:

- La implementación de una herramienta que pueda obtener segmentos de más de 3 segundos en base a los resultados obtenidos por los usuarios y obtener audios de mayor duración donde, varios ornitólogos identifiquen a una misma especie. Al haber un consenso de que una especie realmente está presente en ese segmento, se concluye que esa especie realmente se encuentra ahí. Este audio se podría aportar a la plataforma *eBird*.
- Procesar masivamente todas las grabaciones de aves que se tiene del parque con todos los nodos. Debido a limitaciones en recursos y temporales, sólo ha sido posible procesar una parte de todo el conjunto de grabaciones. Esta tarea consistirá en descargar todas las grabaciones, importarlas a la web y actualizar la base de datos. De aquí podría salir un trabajo que consista en crear una herramienta que automatice este proceso mediante el uso del protocolo SFTP y de APIs que permitan subir los segmentos, los espectrogramas, y actualizar la base de datos en el servidor.
- Entrenar con los resultados obtenidos de los usuarios un algoritmo que tenga la misma funcionalidad que *BirdNET* pero que esté especializado en las especies de *l'Albufera*. Esto va en la línea de lo que se ha mencionado a lo largo de este trabajo, donde se tiene que *BirdNET* utiliza siempre el mismo modelo independientemente de la parametrización que se introduce cuando se configura. De esta manera, se tendría un modelo especializado para las especies de *l'Albufera*. La cantidad de muestras es suficiente dado lo grande que es el conjunto de datos que se tiene.
- A partir de las fechas de las grabaciones y de los nombres de los nodos, elaborar una página que permita visualizar las especies que han encontrado los ornitólogos o *BirdNET* en cada uno de los nodos de *l'Albufera* en cada mes o semana del año. La granularidad temporal puede modificarse al gusto y según los datos que se deseen visualizar. La información de las fechas y los nodos pueden encontrarse en el mismo nombre del archivo WAV, sólo será necesario realizar un procesado en el nombre para extraer dicha información y mostrarla. Para la visualización de la información de base de datos se pueden utilizar herramientas como *Power BI* o *Tableau*. Las herramientas previamente mencionadas permiten realizar consultas sobre una base de datos (en local o en la nube) y elaborar visualizaciones o informes de una forma sencilla e intuitiva.
- En línea con el punto anterior, realizar una exploración de la base datos cuando los ornitólogos ya hayan clasificado una cantidad de segmentos significativa y poder validar los resultados proporcionados por *BirdNET*. También se tendría que estudiar qué especies se escuchan en distintas épocas del año.
- Añadir funcionalidades extras a la web. Aunque la web sea funcional y cumple con el objetivo del trabajo, siempre cabe la opción de añadir funcionalidades extra que permitan enriquecer el proceso de exploración de los datos. A priori, una de las funcionalidades a plantear sería la realización de un mapa que permita visualizar la densidad de especies concretas en los diferentes nodos, así como, poder elegir en qué época del año se quiere visualizar. Otra de estas funcionalidades extra es que el usuario pueda explorar los segmentos de forma cronológica, pudiendo estudiar los segmentos anteriores o posteriores en el tiempo del segmento que se está escuchando. O, dicho de otra manera, que se pueda proporcionar al usuario un fragmento de audio más amplio para obtener más contexto acerca del segmento.
- Estudiar de qué maneras se puede comprimir u optimizar el espacio dentro de la herramienta. Por ejemplo, se podría aplicar un algoritmo de compresión sin pérdidas (en inglés, *lossless*) a las grabaciones y a los segmentos generados para que ocupen menos espacio. Estos algoritmos permiten convertir los ficheros WAV a otros formatos que ocupan menos espacio



conservando la información original. Dos ejemplos de formatos de compresión de audio sin pérdidas son *WavPack* [32] y *FLAC* [33]. Al usar formatos de compresión sin pérdidas no se degradada la calidad de la grabación. Asimismo, las imágenes correspondientes a los espectrogramas generados para cada uno de los segmentos pueden ser comprimidas para que ocupen menos espacio.

- A su vez, se podría estudiar qué umbral de confianza de *BirdNET* es el óptimo para almacenar etiquetas comparándolas con las clasificaciones que realicen los ornitólogos.



## 8. Objetivos de Desarrollo Sostenible

En esta sección se describirá la relación que tiene este trabajo con los Objetivos de Desarrollo Sostenible (ODS) establecidos en el año 2015 por la Asamblea General de las Naciones Unidas [34].

El primer objetivo al que se puede relacionar este trabajo es el **Objetivo 15: Vida de ecosistemas terrestres** donde se pretende administrar los bosques de forma sostenible, combatir la desertificación, parar degradación de las tierras y frenar la pérdida de biodiversidad. Esto es así porque la herramienta que se ha implementado en este TFM ayudará optimizar el control de las especies de ave que habitan en el *Parc Natural de l'Albufera*, tal y como se ha detallado en la sección 1.2. Esta aplicación va alineada con las metas 15.1, 15.5 y 15.8, teniendo que ayudará a proteger este ecosistema, a frenar la pérdida de biodiversidad y a detectar posibles especies invasoras.

Este trabajo podría ayudar a alcanzar la meta 15.4 si se aplicara esta herramienta en entornos montañosos ayudando a preservar la biodiversidad en éstos. Entrenando un modelo basado en *Machine Learning* que permita clasificar distintas fuentes sonoras y aplicándolo al conjunto de datos que se tiene de grabaciones en *l'Albufera*, es posible detectar la presencia de cazadores furtivos por la zona.

## Bibliografía

- [1] Generalitat Valenciana, «Parcs Naturals de la Comunitat Valenciana,» 4 Agosto 2023. [En línea]. Available: <https://parquesnaturales.gva.es/es/web/pn-l-albufera/conocenos>. [Último acceso: 4 Agosto 2023].
- [2] Generalitat Valenciana; SEO/BirdLife, «Entre Picos y Alas. Aves del Parque Natural de L'Albufera,» [En línea]. Available: [https://parquesnaturales.gva.es/es/web/pn-l-albufera/l-albufera/-/asset\\_publisher/ic4LwWyNBGLJ/content/entre-picos-y-alas](https://parquesnaturales.gva.es/es/web/pn-l-albufera/l-albufera/-/asset_publisher/ic4LwWyNBGLJ/content/entre-picos-y-alas). [Último acceso: 6 Agosto 2023].
- [3] SEO/BirdLife, P. Vera y J. Varela, Aves de l'Albufera, Valencia, 2018, p. 13.
- [4] S. Kahl, C. M. Wood, M. Eibl y H. Klinck, «BirdNET: A deep learning solution for avian diversity monitoring,» *Ecological Informatics*, vol. 61, p. 101236, 1 Marzo 2021.
- [5] D. Sanz Montrull, «Estudio de un clasificador de eventos acústicos sobre Raspberry,» 25 Noviembre 2019. [En línea]. Available: <http://hdl.handle.net/10251/133821>. [Último acceso: 06 08 2023].
- [6] E. Alpaydin, Introduction to Machine Learning, Cuarta ed., MIT Press, 2020.
- [7] Y. LeCun, Y. Bengio y G. Hinton, «Deep learning,» *Nature*, vol. 521, n° 7553, p. 436–444, 1 Mayo 2015.
- [8] Z. Yang y Z. Yang, «6.01 - Artificial Neural Networks,» de *Comprehensive Biomedical Physics*, A. Brahme, Ed., Oxford, Elsevier, 2014, pp. 1-17.
- [9] S. J. Russell, P. Norvig y M.-W. Chang, Artificial Intelligence: A Modern Approach, Cuarta ed., Harlow: Pearson Education, 2022, pp. 801-804.
- [10] P. K. Mall, P. K. Singh, S. Srivastav, V. Narayan, M. Paprzycki, T. Jaworska y M. Ganzha, «A comprehensive review of deep neural networks for medical image processing: Recent developments and future opportunities,» *Healthcare Analytics*, vol. 4, p. 100216, 2023.
- [11] F. Rong, «Audio Classification Method Based on Machine Learning,» de *2016 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, 2016, pp. 81-84.
- [12] J. O. Smith, Mathematics of the Discrete Fourier Transform (DFT) with Audio Applications, Segunda ed., W3K Publishing, 2007.
- [13] T. Giannakopoulos y A. Pikrakis, Introduction to audio analysis : a MATLAB approach, Primera ed., Oxford: Academic Press, 2014, pp. 33-40.
- [14] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg y O. Nieto, «librosa: Audio and music signal analysis in python,» de *Proceedings of the 14th python in science conference*, 2015, pp. 18-25.
- [15] S. S. Stevens, J. Volkman y E. B. Newman, «A Scale for the Measurement of the Psychological Magnitude Pitch,» *Journal of the Acoustical Society of America*, vol. 8, n° 3, pp. 185-190, 1937.
- [16] D. O'Shaughnessy, Speech communication: human and machine, Addison-Wesley, 1987, p. 150.

- [17] E. F. Codd, «A Relational Model of Data for Large Shared Data Banks,» *Communications of the ACM*, vol. 13, n° 6, p. 377–387, Junio 1970.
- [18] Cornell Lab of Ornithology, «eBird España,» 2023. [En línea]. Available: <https://ebird.org/spain/about>. [Último acceso: 6 Agosto 2023].
- [19] Cornell Lab of Ornithology, «BirdNET Sound ID,» [En línea]. Available: <https://birdnet.cornell.edu/>. [Último acceso: 06 08 2023].
- [20] S. Kahl, «Repositorio de GitHub de BirdNET,» [En línea]. Available: <https://github.com/kahst/BirdNET-Analyzer>. [Último acceso: 06 08 2023].
- [21] The PHP Group, «PHP: Documentation,» 2001-2023. [En línea]. Available: <https://www.php.net/docs.php>. [Último acceso: 6 Agosto 2023].
- [22] Wildlife Acoustics, «Grabadora Song Meter Mini,» [En línea]. Available: <https://www.wildlifeacoustics.com/products/song-meter-mini>. [Último acceso: 6 Agosto 2023].
- [23] B. McFee, M. McVicar, D. Faronbi, I. Roman, M. Gover, S. Balke, S. Seyfarth, A. Malek, C. Raffel, V. Lostanlen, B. v. Niekirk, D. Lee, F. Cwitkowitz, F. Zalkow, O. Nieto, D. Ellis, ... y W. Pimienta, «ibrosa/librosa: 0.10.0.post2 (0.10.0.post2),» Zenodo, 2023. [En línea]. Available: <https://doi.org/10.5281/zenodo.7746972>. [Último acceso: 14 Agosto 2023].
- [24] Microsoft Coporation, «What is the Windows Subsystem for Linux?,» 07 Julio 2023. [En línea]. Available: <https://learn.microsoft.com/en-us/windows/wsl/about>. [Último acceso: 18 Agosto 2023].
- [25] M. Marqués, Bases de datos, Primera ed., Castelló de la Plana: Universitat Jaume I. Servei de Comunicació i Publicacions, 2011.
- [26] S. Kahl, Identifying Birds by Sound: Large-scale Acoustic Event Recognition for Avian Activity Monitoring, Chemnitz: Universitätsverlag Chemnitz, 2019, pp. 55-56.
- [27] D. Merkel, «Docker: lightweight linux containers for consistent development and deployment,» *Linux journal*, vol. 2014, n° 239, p. 2, 2014.
- [28] Docker, Inc., «Documentación de Docker,» 2013-2023. [En línea]. Available: <https://docs.docker.com/>. [Último acceso: 22 Agosto 2023].
- [29] Apache Friends, «Acerca del proyecto XAMPP,» 2023. [En línea]. Available: <https://www.apachefriends.org/es/about.html>. [Último acceso: 25 Agosto 2023].
- [30] phpMyAdmin contributors, «Web de phpMyAdmin,» 2003-2023. [En línea]. Available: <https://www.phpmyadmin.net/>. [Último acceso: 25 Agosto 2023].
- [31] Plesk International GmbH, «Sitio web de Plesk,» 2023. [En línea]. Available: <https://www.plesk.com/>. [Último acceso: 28 Agosto 2023].
- [32] D. Bryant, «Web de WavPack,» 2022. [En línea]. Available: <https://www.wavpack.com/>. [Último acceso: 20 Agosto 2023].
- [33] Josh Coalson; Xiph.Org Foundation, «Web de FLAC,» 2000-2009, 2011-2022. [En línea]. Available: <https://xiph.org/flac/>. [Último acceso: 20 Agosto 2023].



- [34] Naciones Unidas, «Objetivos de Desarrollo Sostenible,» 2023. [En línea]. Available: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>. [Último acceso: 02 Septiembre 2023].

## Anexo

### A. Lista de especies consideradas por *BirdNET*

**Tabla A.1** Lista de especies que ha utilizado *BirdNET* como lista de especies cuando se le indican las coordenadas de *l'Albufera* como argumento.

Nombre científico	Nombre común	Nombre científico	Nombre común	Nombre científico	Nombre común
Hirundo rustica	Golondrina Común	Chroicocephalus ridibundus	Gaviota Reidora	Falco columbarius	Esmerejón
Sturnus unicolor	Estornino Negro	Aquila chrysaetos	Águila Real	Pluvialis apricaria	Chorlito Dorado Europeo
Carduelis carduelis	Jilguero Europeo	Mareca strepera	Ánade Friso	Myiopsitta monachus	Cotorra Argentina
Turdus merula	Mirlo Común	Corvus corone	Corneja Negra	Ixobrychus minutus	Avetorillo Común
Passer domesticus	Gorrión Común	Athene noctua	Mochuelo Europeo	Cinclus cinclus	Mirlo Acuático Europeo
Delichon urbicum	Avión Común	Vanellus vanellus	Avefría Europea	Fringilla montifringilla	Pinzón Real
Columba palumbus	Paloma Torcaz	Corvus monedula	Grajilla Occidental	Columba oenas	Paloma Zurita
Serinus serinus	Serín Verdecillo	Egretta garzetta	Garceta Común	Clamator glandarius	Críalo Europeo
Streptopelia decaocto	Tórtola Turca	Sitta europaea	Trepador Azul	Aegypius monachus	Buitre Negro
Apus apus	Vencejo Común	Curruca undata	Curruca Rabilarga	Sternula albifrons	Charrancito Común
Motacilla alba	Lavandera Blanca	Alcedo atthis	Martín Pescador Común	Regulus regulus	Reyezuelo Sencillo
Parus major	Carbonero Común	Accipiter nisus	Gavilán Común	Chlidonias hybrida	Fumarel Cariblanco
Pica pica	Urraca Común	Nycticorax nycticorax	Martinete Común	Tringa erythropus	Archibebe Oscuro
Fringilla coelebs	Pinzón Vulgar	Burhinus oedicephalus	Alcaraván Común	Chlidonias niger	Fumarel Común
Merops apiaster	Abejaruco Europeo	Spinus spinus	Jilguero Lúgano	Tyto alba	Lechuza Común
Gyps fulvus	Buitre Leonado	Troglodytes troglodytes	Chochín Paleártico	Morus bassanus	Alcatraz Atlántico
Falco tinnunculus	Cernícalo Vulgar	Gallinago gallinago	Agachadiza Común	Phylloscopus sibilatrix	Mosquitero Silbador
Curruca melanocephala	Curruca Cabecinegra	Ardea purpurea	Garza Imperial	Anas acuta	Ánade Rabudo

Upupa epops	Abubilla Común	Regulus ignicapilla	Reyezuelo Listado	Limosa lapponica	Aguja Colipinta
Phoenicurus ochruros	Colirrojo Tizón	Melanocorypha calandra	Calandria Común	Stercorarius pomarinus	Págalo Pomarino
Sylvia atricapilla	Curruca Capirozada	Curruca communis	Curruca Zarcera	Aythya fuligula	Porrón Moñudo
Linaria cannabina	Pardillo Común	Riparia riparia	Avión Zapador	Sterna hirundo	Charrán Común
Erithacus rubecula	Petirrojo Europeo	Acrocephalus arundinaceus	Carricero Tordal	Prunella collaris	Acentor Alpino
Chloris chloris	Verderón Común	Falco naumanni	Cernícalo Primilla	Chlidonias leucopterus	Fumarel Aliblanco
Phylloscopus collybita	Mosquitero Común	Charadrius alexandrinus	Chorlitejo Patinegro	Haematopus ostralegus	Ostrero Euroasiático
Columba livia	Paloma Bravía	Passer hispaniolensis	Gorrión Moruno	Psittacula krameri	Cotorra de Kramer
Cyanistes caeruleus	Herrerillo Común	Coturnix coturnix	Codorniz Común	Asio otus	Búho Chico
Anas platyrhynchos	Ánade Azulón	Emberiza cia	Escribano Montesino	Arenaria interpres	Vuelvepedras Común
Circus aeruginosus	Aguilucho Lagunero Occidental	Anas crecca	Cerceta Común	Aythya nyroca	Porrón Pardo
Ficedula hypoleuca	Papamoscas Cerrojillo	Saxicola rubetra	Tarabilla Norteña	Scolopax rusticola	Chocha Perdiz
Luscinia megarhynchos	Ruiseñor Común	Tringa nebularia	Archibebe Claro	Phasianus colchicus	Faisán Vulgar
Emberiza calandra	Escribano Triguero	Rallus aquaticus	Rascón Europeo	Locustella naevia	Buscarla Pintoja
Galerida cristata	Cogujada Común	Coccothraustes coccothraustes	Picogordo Común	Elanus caeruleus	Elanio Común
Cettia cetti	Cetia Ruiseñor	Pernis apivorus	Abejero Europeo	Stercorarius parasiticus	Págalo Parásito
Picus sharpei	Pito Ibérico	Podiceps nigricollis	Zampullín Cuellinegro	Estrilda astrild	Estrilda Común
Saxicola rubicola	Tarabilla Europea	Porphyrio porphyrio	Calamón Común	Pterocles alchata	Ganga Ibérica
Alectoris rufa	Perdiz Roja	Monticola solitarius	Roquero Solitario	Monticola saxatilis	Roquero Rojo
Corvus corax	Cuervo Grande	Ardea alba	Garceta Grande	Remiz pendulinus	Pájaro Moscón Europeo
Hieraaetus pennatus	Águila Calzada	Anthus campestris	Bisbita Campestre	Hydroprogne caspia	Pagaza Piquirroja
Aegithalos caudatus	Mito Común	Calidris alpina	Correlimos Común	Lymnocyptes minimus	Agachadiza Chica

Muscicapa striata	Papamoscas Gris	Anthus spinoletta	Bisbita Alpino	Chroicocephalus genei	Gaviota Pico fina
Gallinula chloropus	Gallineta Común	Podiceps cristatus	Somormujo Lavanco	Glareola pratincola	Canastera Común
Himantopus himantopus	Cigüeñuela Común	Calidris pugnax	Combatiente	Hippolais icterina	Zarcero Icterino
Milvus migrans	Milano Negro	Oenanthe hispanica	Collalba Rubia Occidental	Dryobates minor	Pico Menor
Oriolus oriolus	Oropéndola Europea	Phoenicurus phoenicurus	Colirrojo Real	Carduelis citrinella	Verderón Serrano
Phylloscopus trochilus	Mosquitero Musical	Grus grus	Grulla Común	Turdus torquatus	Mirlo Capiblanco
Ptyonoprogne rupestris	Avión Roquero	Sturnus vulgaris	Estornino Pinto	Emberiza hortulana	Escribano Hortelano
Anthus pratensis	Bisbita Pratense	Loxia curvirostra	Piquituerto Común	Acrocephalus schoenobaenus	Carricerín Común
Lanius senator	Alcaudón Común	Falco peregrinus	Halcón Peregrino	Caprimulgus europaeus	Chotacabras Europeo
Ardea cinerea	Garza Real	Thalasseus sandvicensis	Charrán Patinegro	Oenanthe leucura	Collalba Negra
Phylloscopus bonelli	Mosquitero Papialbo	Anthus trivialis	Bisbita Arbóreo	Anser anser	Ánsar Común
Turdus philomelos	Zorzal Común	Circus cyaneus	Aguilucho Pálido	Ciconia nigra	Cigüeña Negra
Spatula clypeata	Cuchara Común	Plegadis falcinellus	Morito Común	Larus canus	Gaviota Cana
Fulica atra	Focha Común	Platalea leucorodia	Espátula Común	Calonectris diomedea	Pardela Cenicienta
Phalacrocorax carbo	Cormorán Grande	Cyanopica cooki	Rabilargo Ibérico	Asio flammeus	Búho Campestre
Motacilla flava	Lavandera Boyera	Recurvirostra avosetta	Avoceta Común	Hydrocoloeus minutus	Gaviota Enana
Buteo buteo	Busardo Ratonero	Calidris minuta	Correlimos Menudo	Pluvialis squatarola	Chorlito Gris
Milvus milvus	Milano Real	Netta rufina	Pato Colorado	Calidris temminckii	Correlimos de Temminck
Galerida theklae	Cogujada Montesina	Prunella modularis	Acentor Común	Acrocephalus melanopogon	Carricerín Real
Oenanthe oenanthe	Collalba Gris	Circus pygargus	Aguilucho Cenizo	Anthus petrosus	Bisbita Costero
Acrocephalus scirpaceus	Carricero Común	Ichthyaetus audouinii	Gaviota de Audouin	Puffinus puffinus	Pardela Pichoneta
Bubulcus ibis	Garcilla Bueyera	Mareca penelope	Silbón Europeo	Gavia arctica	Colimbo Ártico

Cuculus canorus	Cuco Común	Pandion haliaetus	Águila Pescadora	Curruca curruca	Curruca Zarcerilla
Certhia brachydactyla	Agateador Europeo	Calandrella brachydactyla	Terrera Común	Tringa stagnatilis	Archibebe Fino
Dendrocopos major	Pico Picapinos	Charadrius hiaticula	Chorlitejo Grande	Stercorarius skua	Págalo Grande
Lanius meridionalis	Alcaudón Real	Curruca hortensis	Curruca Mirlona Occidental	Lanius collurio	Alcaudón Dorsirrojo
Turdus viscivorus	Zorzal Charlo	Numenius phaeopus	Zarapito Trinador	Mergus serrator	Serreta Mediana
Motacilla cinerea	Lavandera Cascadeña	Spatula querquedula	Cerceta Carretona	Alca torda	Alca Común
Actitis hypoleucos	Andarríos Chico	Gelochelidon nilotica	Pagaza Piconegra	Caprimulgus ruficollis	Chotacabras Cuellirrojo
Cecropis daurica	Golondrina Dáurica	Tringa glareola	Andarríos Bastardo	Pyrrhula pyrrhula	Camachuelo Común
Charadrius dubius	Chorlitejo Chico	Jynx torquilla	Torcecuello Euroasiático	Ficedula albicollis	Papamoscas Acollarado
Larus fuscus	Gaviota Sombría	Calidris alba	Correlimos Tridáctilo	Botaurus stellaris	Avetoro Común
Emberiza cirius	Escribano Soteño	Emberiza schoeniclus	Escribano Palustre	Porzana porzana	Polluela Pintoja
Petronia petronia	Gorrión Chillón	Aquila adalberti	Águila Imperial Ibérica	Emberiza citrinella	Escribano Cerillo
Periparus ater	Carbonero Garrapinos	Apus melba	Vencejo Real	Larus marinus	Gavión Atlántico
Tringa ochropus	Andarríos Grande	Limosa limosa	Aguja Colinegra	Anthus cervinus	Bisbita Gorjirrojo
Curruca iberiae	Curruca Carrasqueña Occidental	Apus pallidus	Vencejo Pálido	Ptyonoprogne fuligula	Avión Isabel
Tachybaptus ruficollis	Zampullín Común	Tringa totanus	Archibebe Común	Strix aluco	Cárabo Común
Pyrrhocorax pyrrhocorax	Chova Piquirroja	Coracias garrulus	Carraca Europea	Rissa tridactyla	Gaviota Tridáctila
Passer montanus	Gorrión Molinero	Luscinia svecica	Ruiseñor Pechiazul	Phalaropus lobatus	Falaropo Picofino
Lophophanes cristatus	Herrerillo Capuchino	Sylvia borin	Curruca Mosquitera	Euodice cantans	Capuchino Picoplata Africano
Cisticola juncidis	Cistícola Buitrón	Calidris canutus	Correlimos Gordo	Calidris melanotos	Correlimos Pectoral
Lullula arborea	Alondra Totovía	Numenius arquata	Zarapito Real	Pterocles orientalis	Ganga Ortega



Hippolais polyglotta	Zarcero Políglota	Ardeola ralloides	Garcilla Cangrejera	Iduna opaca	Zarcero Bereber
Ciconia ciconia	Cigüeña Blanca	Ichthyaetus melanocephalus	Gaviota Cabecinegra	Podiceps auritus	Zampullín Cuellirrojo
Tadorna tadorna	Tarro Blanco	Calidris ferruginea	Correlimos Zarapitín	Tchagra senegalus	Chagra del Senegal
Larus michahellis	Gaviota Patiamarilla	Falco subbuteo	Alcotán Europeo	Eremophila alpestris	Alondra Cornuda
Streptopelia turtur	Tórtola Europea	Phylloscopus ibericus	Mosquitero Ibérico	Ardenna grisea	Pardela Sombría
Circaetus gallicus	Culebrera Europea	Otus scops	Autillo Europeo	Carduelis corsicana	Verderón Corso
Garrulus glandarius	Arrendajo Euroasiático	Locustella luscinioides	Buscarla Unicolor	Podiceps griseana	Somormujo Cuellirrojo
Aythya ferina	Porrón Europeo	Accipiter gentilis	Azor Común	Melanitta nigra	Negrón Común
Alauda arvensis	Alondra Común	Turdus iliacus	Zorzal Alirrojo	Oxyura jamaicensis	Malvasía Canela
Phoenicopterus roseus	Flamenco Común				