



## **E4.2. Versión Alpha de la plataforma para la predicción y protección de aves destinados a ser parques eólicos**



VICEPRESIDENCIA  
TERCERA DEL GOBIERNO  
MINISTERIO  
PARA LA TRANSICIÓN ECOLÓGICA  
Y EL RETO DEMOGRÁFICO



Fundación Biodiversidad



Plan de Recuperación,  
Transformación  
y Resiliencia



Financiado por  
la Unión Europea  
NextGenerationEU

## E4.2. Versión Alpha de la plataforma para la predicción y protección de aves destinados a ser parques eólicos.....1

|        |                                                                             |    |
|--------|-----------------------------------------------------------------------------|----|
| 1      | Introducción .....                                                          | 3  |
| 1.1    | Objetivo .....                                                              | 3  |
| 1.2    | Alcance.....                                                                | 3  |
| 2      | Versión Alpha de la plataforma.....                                         | 3  |
| 2.1    | Integración de componentes de la plataforma en un escenario controlado..... | 3  |
| 2.2    | Despliegue de la Plataforma .....                                           | 4  |
| 2.2.1  | Puerta de Enlace (Gateway) y Frontend .....                                 | 4  |
| 2.2.2  | Servicios de Lógica de Aplicación y API .....                               | 5  |
| 2.2.3  | Almacenamiento de Datos (PostgreSQL).....                                   | 6  |
| 2.2.4  | Servicios Asíncronos (IoT y Tareas).....                                    | 6  |
| 2.2.5  | Streaming de Vídeo .....                                                    | 7  |
| 2.2.6  | Redes y Persistencia: .....                                                 | 8  |
| 2.3    | Despliegue automatizado en Github .....                                     | 8  |
| 2.4    | Visualizador de Mapa Principal.....                                         | 8  |
| 2.5    | Funcionalidad:.....                                                         | 9  |
| 2.5.1  | Consulta de Recurso Eólico .....                                            | 9  |
| 2.5.2  | Consulta de Cuadrícula Mesoescalar.....                                     | 11 |
| 2.6    | Lista de Cámaras (Gestión de Dispositivos) .....                            | 11 |
| 2.6.1  | Usuario No Registrado:.....                                                 | 11 |
| 2.6.2  | Usuario Registrado .....                                                    | 12 |
| 2.7    | Formulario de Registro de Cámara .....                                      | 13 |
| 2.8    | Vista de Detalle de Cámara (Datos y Analíticas) .....                       | 13 |
| 2.8.1  | Datos del Dispositivo .....                                                 | 14 |
| 2.8.2  | Análisis de Detecciones .....                                               | 14 |
| 2.8.3  | Visualización de Mapas de Calor .....                                       | 14 |
| 2.9    | Vista del Blog (Divulgación y Noticias) .....                               | 15 |
| 2.10   | Panel de Administración .....                                               | 16 |
| 2.11   | Interfaces de Autenticación y Registro .....                                | 16 |
| 2.11.1 | Inicio de Sesión.....                                                       | 16 |
| 2.11.2 | Proceso de Registro.....                                                    | 17 |
| 2.11.3 | Flujo de Aprobación .....                                                   | 18 |
| 3      | Conclusión .....                                                            | 18 |

## 1 Introducción

### 1.1 Objetivo

Este documento corresponde al entregable:

#### **E4.2 - Versión Alpha de la plataforma para la predicción y protección de aves destinados a ser parques eólicos**

Refleja los trabajos realizados y los resultados alcanzados durante la ejecución de la actividad:

#### **A4.3 Módulo de gestión de información, elaboración de directrices y criterios de planificación ambiental**

Esta tarea se encuadra dentro de LA ACTIVIDAD A4 y la línea de investigación:

**A4. Sistema de gestión de la información de aves y tecnologías de visualización**, cuyo objetivo se centra en investigar y diseñar tecnologías de análisis y visualización de datos masivos para la predicción y protección de aves que permita optimizar la gestión de la información y elaborar directrices y criterios de planificación ambiental.

### 1.2 Alcance

Este documento se encuentra en la versión 1.0, y aunque está sujeta a nuevas evoluciones si en las siguientes fases del proyecto se hacen necesarios ajustes, la investigación aquí plasmada trata de presentar el trabajo realizado para la consecución de los objetivos de la tarea de diseñar una plataforma de IA explicable para optimizar la evaluación de espacios destinados a ser parques eólicos.

## 2 Versión Alpha de la plataforma

Para materializar los objetivos de la plataforma, se ha diseñado un conjunto de visualizaciones e interfaces orientadas a transformar los datos (tanto de los sensores IoT como de las fuentes de contexto) en información comprensible y accionable para los perfiles de usuario definidos (Gestores, Investigadores y público general).

### 2.1 Integración de componentes de la plataforma en un escenario controlado

El objetivo de esta arquitectura es la interoperabilidad y una clara separación de responsabilidades. En lugar de tener un único servicio "monolítico" que haga todo, el trabajo se divide en capas especializadas. Esto hace que el sistema sea más seguro, eficiente y fácil de mantener.

Cada componente tiene un trabajo muy específico:

1. El Edge (Dispositivo Jetson) Su única tarea es analizar el vídeo y detectar eventos (aves, etc.). Una vez que detecta algo, publica un mensaje (ej. "ave detectada en coordenadas X,Y") usando MQTT.
2. El Servicio de Ingesta (Ingest - Flask) Es el primer receptor en la nube. Su trabajo es escuchar los mensajes MQTT, validarlos (asegurarse de que los datos son correctos),

persistirlos (guardarlos en la base de datos) y exponerlos a través de una API REST. También es el único servicio que habla con las API de terceros (Xeno-canto, IDECYL).

3. El Servicio de Coordinación (Coordinate - Flask) Actúa como el guardia de seguridad de la API. No maneja datos. Su única función es asegurar el acceso. Intercepta todas las peticiones y comprueba el *token* JWT para verificar que el usuario tiene permiso para ver los datos que solicita.
4. El Backend (Node.js y TypeScript) Su trabajo es optimizar la información para la interfaz de usuario (UI). No tiene acceso directo a MQTT; solo habla con el servicio de Coordinación para pedir los datos que necesita. Luego, puede agregar o formatear esos datos para que el frontend los consuma fácilmente.
5. El Frontend (React) Es la capa de presentación. Su única responsabilidad es consumir los datos ya optimizados que le entrega el Backend y representarlos visualmente (en el mapa, en gráficos) y reproducir el vídeo en vivo (usando HLS).

Los conectores están versionados, documentados en OpenAPI y sujetos a *rate-limiting*. La caché reduce coste/latencia para cargas intensivas. Las llamadas a terceros se encapsulan en Ingest, manteniendo un único punto de salida y política de reintento.

## 2.2 Despliegue de la Plataforma

Para lograr una integración eficiente y reproducible de la arquitectura híbrida de IA4Birds, todos los componentes de la nube han sido contenedorizados utilizando Docker. Esta estrategia es fundamental, ya que encapsula cada microservicio (bases de datos, APIs, streaming, etc.) con todas sus dependencias, garantizando un entorno de ejecución coherente y aislado.

La orquestación de esta compleja pila de servicios se gestiona mediante Docker Compose. El fichero docker-compose.yml define la arquitectura completa, especificando cómo cada servicio se construye, se configura y se interconecta.

A diferencia de una pila simple, la arquitectura de IA4Birds es granular y está diseñada para la separación de responsabilidades, tal como se define en el docker-compose.yml:

### 2.2.1 Puerta de Enlace (Gateway) y Frontend

- webserver (Nginx): Actúa como el proxy inverso y único punto de entrada a la plataforma. Termina la conexión TLS (puertos 443) y redirige el tráfico de forma segura al frontend o al backend según corresponda.
- frontend (React): La interfaz de usuario construida con Vite, servida como un contenedor estático o de desarrollo.

```
frontend:
  build: ./ia4birds-all/ai4birds-web/ai4birds-web-app-frontend
  container_name: frontend
  ports:
    - 8080:8080
  env_file:
    - /etc/envs/.env
  networks:
    - bird-net

webserver:
  depends_on:
    - frontend
    - backend
  build:
    context: .
    dockerfile: nginx_dockerfile/Dockerfile
  args:
    ENVIRONMENT: "${ENVIRONMENT}"
  container_name: webserver
  restart: unless-stopped
  ports:
    - "80:80"
    - "443:443"
  volumes:
    - /etc/letsencrypt:/etc/letsencrypt
  networks:
    - bird-net
```

Fig.1. Puerta de enlace

## 2.2.2 Servicios de Lógica de Aplicación y API

- backend (Node.js): El orquestador principal (BFF) que gestiona la autenticación de usuarios y la lógica de negocio, comunicándose con los servicios internos.
- ai4birds-coordinate-service (Flask): La pasarela de seguridad. Valida los tokens JWT antes de permitir el acceso a los datos.
- ai4birds-ingest-service (Flask): El núcleo de la ingesta de datos. Se conecta a las BBDD, ejecuta los conectores a terceros y está suscrito a Mosquitto para recibir datos del Edge.

```
backend:
  build: ./ia4birds-all/ai4birds-web/ai4birds-web-app-backend
  container_name: backend
  env_file:
    - /etc/envs/.env
  volumes:
    - /etc/letsencrypt/archive/ia4birds-platform.air-institute.com:/etc/letsencrypt/archive/ia4birds-platform.air-institute.com
    - /hls-stream:/home/node/app/hls-stream
  ports:
    - 5030:3030
  depends_on:
    - postgresql
    - redis
  networks:
    - bird-net

ai4birds-coordinate-service:
  build: ./ia4birds-all/ai4birds-coordinate-service
  container_name: ai4birds-coordinate
  env_file:
    - /etc/envs/.env
  ports:
    - 5001:5000
  networks:
    - bird-net

ai4birds-ingest-service:
  build: ./ia4birds-all/ai4birds-ingest-service
  container_name: ai4birds-ingest
  mem_limit: 6g
  memswap_limit: 6g
  env_file:
    - /etc/envs/.env
  volumes:
    - /home/bisite/exclusion_eolica.csv:/home/exclusion_eolica.csv
    - ./ia4birds-all/ai4birds-ingest-service/ai4birds_ingest_service/heatmaps:/app/ai4birds_ingest_service/heatmaps
  ports:
    - 5002:5000
  networks:
    - bird-net
```

Fig.2. Servicios de lógica

## 2.2.3 Almacenamiento de Datos (PostgreSQL)

El sistema utiliza una estrategia de tres bases de datos PostgreSQL separadas para aislar las cargas de trabajo:

- postgresql: La base de datos principal de configuración (usuarios, cámaras, roles).
- postgresql-ingest: La base de datos operativa de alto volumen donde se guardan las detecciones, estadísticas y heatmaps provenientes del Edge.
- postgresql-analysis: Una base de datos dedicada a vistas materializadas y agregados para análisis.

```
postgresql:
  image: "postgres:14-alpine"
  container_name: postgres
  env_file:
    - /etc/envs/.env
  volumes:
    - ./data/postgresDB/init.sql:/docker-entrypoint-initdb.d/init.sql
  networks:
    - bird-net

postgresql-ingest:
  image: "postgres:14-alpine"
  container_name: postgres-ingest
  env_file:
    - /etc/envs/.env
  volumes:
    - /etc/ai4birds-ingest-db/data:/var/lib/postgresql/data
    - ./data/postgresIngestDB/init.sql:/docker-entrypoint-initdb.d/init.sql
  environment:
    - POSTGRES_USER=${POSTGRES_INGEST_USER}
    - POSTGRES_DB=${POSTGRES_INGEST_DB}
    - POSTGRES_PASSWORD=${POSTGRES_INGEST_PASSWORD}
  networks:
    - bird-net

postgresql-analysis:
  image: "postgres:14-alpine"
  container_name: postgres-analysis
  env_file:
    - /etc/envs/.env
  volumes:
    - /etc/ai4birds-analysis-db/data:/var/lib/postgresql/data
  environment:
    - POSTGRES_USER=${POSTGRES_ANALYSIS_USER}
    - POSTGRES_DB=${POSTGRES_ANALYSIS_DB}
    - POSTGRES_PASSWORD=${POSTGRES_ANALYSIS_PASSWORD}
  networks:
    - bird-net
```

Fig.3. Almacenamiento de datos

## 2.2.4 Servicios Asíncronos (IoT y Tareas)

- mosquitto: El broker MQTT que recibe los mensajes (detecciones, estado) publicados por los dispositivos Edge.
- redis: Un servicio de doble propósito: actúa como caché de alto rendimiento y como broker de mensajes para Celery.
- celery-worker y celery-beat: Los componentes de tareas en segundo plano. beat programa tareas periódicas (ej. "consultar Xeno-canto cada 24h") y worker las ejecuta, basándose en el código del servicio de ingesta.

```
redis:
  image: redis:6.2-alpine
  container_name: redis
  env_file:
    - /etc/envs/.env
  ports:
    - '6379:6379'
  command: redis-server --save 20 1 --loglevel warning --requirepass "${REDIS_PASSWORD}"
  volumes:
    - cache:/data
  networks:
    - bird-net
    - redis-net

mosquitto:
  image: eclipse-mosquitto
  container_name: mosquitto
  volumes:
    - ${MQTT_CONFIG_VOLUME}
    - ${MQTT_DATA_VOLUME}
    - ${MQTT_LOG_VOLUME}
  env_file:
    - /etc/envs/.env
  ports:
    - "5083:1883"
    - "9001:9001"
  networks:
    - bird-net

celery-worker:
  build: ./ai4birds-all/ai4birds-ingest-service
  container_name: celery-worker
  command: celery -A ai4birds_celery.app worker --loglevel=info
  env_file:
    - /etc/envs/.env
  depends_on:
    - ai4birds-ingest-service
    - redis
  networks:
    - redis-net
    - bird-net
```

Fig.4. Servicios asíncronos

## 2.2.5 Streaming de Vídeo

- mediamtx: El servidor de streaming que ingesta las señales RTSP de las cámaras y las repubblica en formato HLS para su consumo en el frontend.

```
mediamtx:
  build: ./mediamtx-custom
  container_name: mediamtx
  ports:
    - "8554:8554"      # RTSP
    - "9997:9997"      # RTMP / API
    - "8889:8889"      # RTMP (in)
    - "8890:8890/udp"  # UDP (SRT / RTP)
    - "8189:8189"      # Control API (REST)
  expose:
    - "8888"           # HLS (solo accesible desde NGINX)
  volumes:
    - camara-config:/app/config
  networks:
    - bird-net
```

Fig.5. Streaming de vídeo

### 2.2.6 Redes y Persistencia:

Todos los servicios están conectados a una red interna principal (bird-net), utilizando un driver: bridge. Esta red virtualizada permite que los contenedores se comuniquen entre sí de forma segura usando sus nombres de servicio (ej. backend puede llamar a ai4birds-coordinate), sin exponer sus puertos internos al exterior. Se definen volúmenes persistentes para los datos críticos (como las bases de datos PostgreSQL, la caché de Redis y las configuraciones), asegurando que la información no se pierda si los contenedores se reinician.

## 2.3 Despliegue automatizado en Github

El proceso de despliegue continuo para IA4Birds se gestiona de forma automatizada mediante GitHub Actions. Todo el código fuente, la configuración y los flujos de trabajo están centralizados en el repositorio principal del proyecto: <https://github.com/AIRInstitute/IA4Birds>.

La plataforma utiliza flujos de trabajo definidos en archivos YAML que se ejecutan en *self-hosted runners*, permitiendo la compilación, prueba y despliegue del complejo *stack* de microservicios en los servidores gestionados.

Este método asegura que cualquier commit o pull request realizado en el repositorio active automáticamente un *pipeline* de integración y despliegue continuo (CI/CD). La estrategia abarca las siguientes etapas automatizadas:

- Compilación y Pruebas: Verificación de dependencias (Python/JS) y ejecución de la batería de pruebas, que incluye pruebas unitarias, pruebas de integración (ej. con Postman) y publicaciones MQTT simuladas.
- Generación de Imágenes Docker: Cada microservicio (frontend, backend, ai4birds-coordinate-service, ai4birds-ingest-service, etc.) se encapsula en una imagen Docker independiente a partir de su Dockerfile.
- Publicación en el Registro: Las imágenes de los servicios de IA4Birds se cargan en un registro de contenedores.
- Despliegue: Implementación del *stack* completo de docker-compose en los entornos de prueba o producción, orquestado mediante los *self-hosted runners*.
- Notificación y Registro: Captura de los resultados de cada fase (compilación, éxito/fallo de pruebas, despliegue) directamente en la interfaz de GitHub Actions.

## 2.4 Visualizador de Mapa Principal

**Objetivo:** Ofrecer una visión geoespacial panorámica e interactiva que sirva como punto de entrada principal a la plataforma. Permite a todos los perfiles de usuario (públicos y registrados) explorar las capas de contexto territorial, los datos de ciencia ciudadana y la ubicación de los dispositivos de monitorización.

**Descripción:** Esta interfaz es la vista central de la aplicación. Se compone de un mapa interactivo (tipo Leaflet/Mapbox) que carga por defecto con varias capas de información superpuestas. En el lateral, dispone de un panel de control que permite al usuario activar o desactivar capas adicionales para personalizar el análisis.

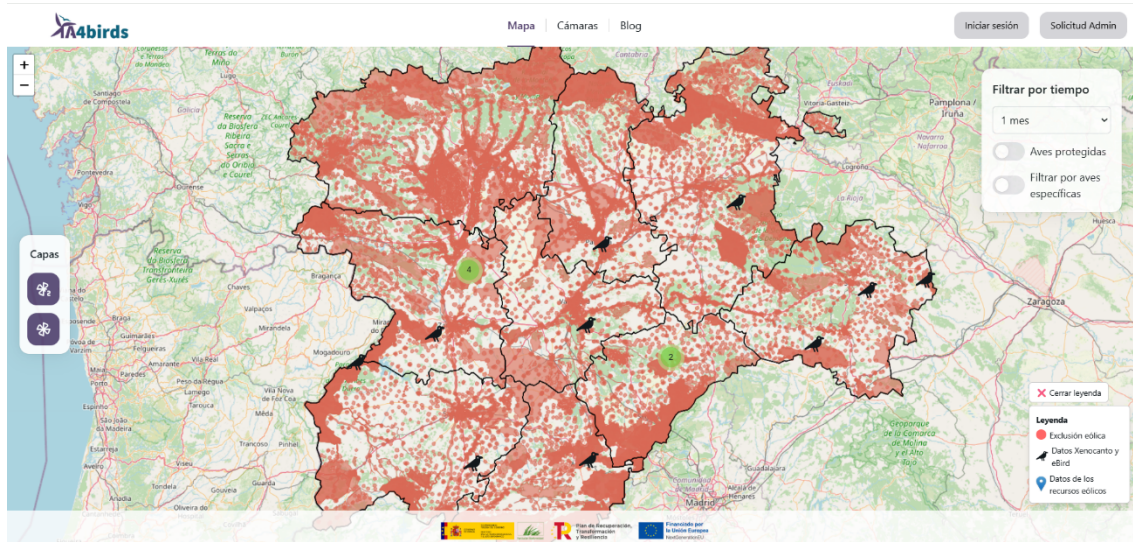


Fig.6. Pantalla principal

## 2.5 Funcionalidad:

El usuario puede navegar (pan/zoom) por el mapa. Por defecto, visualiza la capa de exclusión eólica y los puntos de avistamiento de aves.

### 2.5.1 Consulta de Recurso Eólico

Al activar esta capa, el usuario puede hacer clic en cualquier punto del mapa. Esta acción consulta el endpoint del Mapa Eólico Ibérico para esas coordenadas y despliega una ventana modal con cuatro visualizaciones gráficas clave:



Fig.7. Perfil medio diario velocidad del viento en m/s

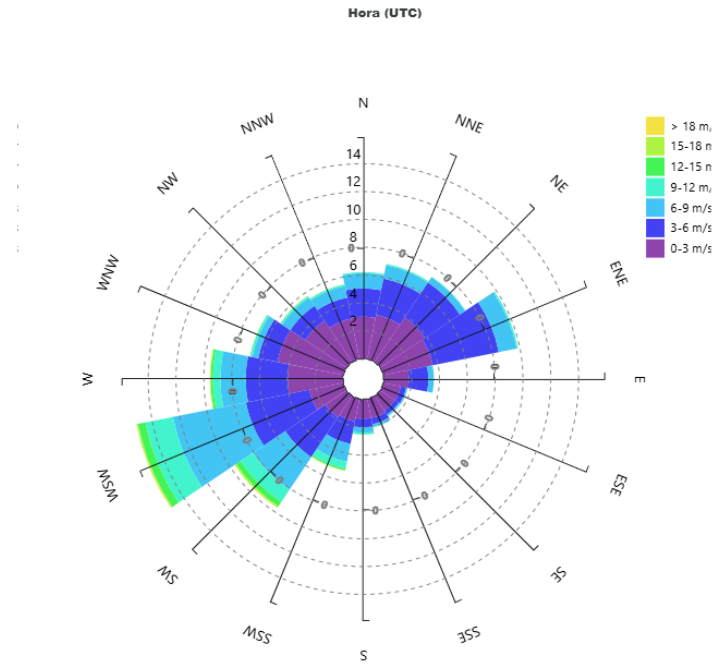


Fig.8. Rosa de los vientos

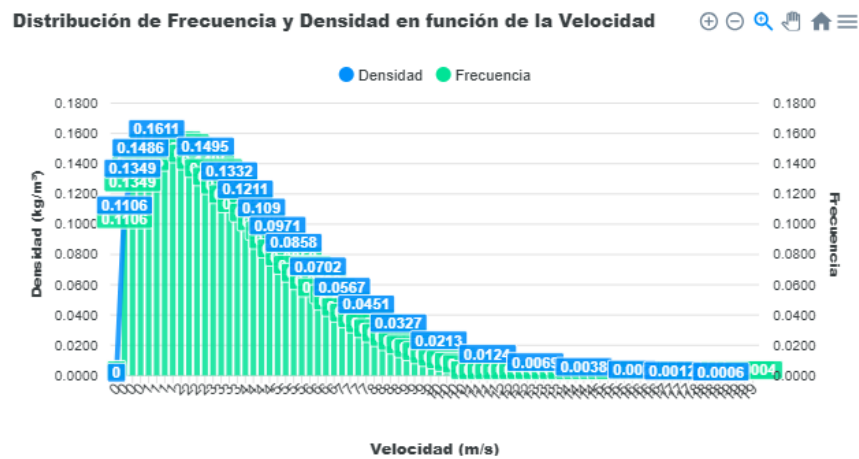


Fig.9. Distribución de frecuencia y densidad de la velocidad.

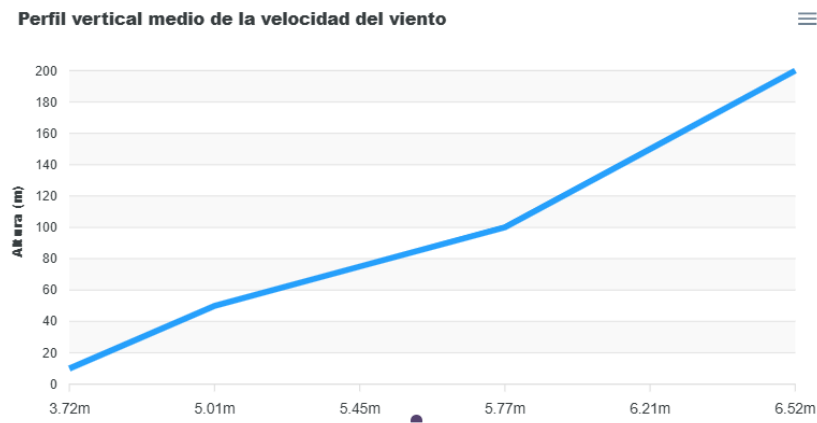


Fig.10. Perfil vertical medio de la velocidad del viento

## 2.5.2 Consulta de Cuadrícula Mesoescalar

Al activar esta capa, la interfaz superpone un "mapa recomendador" (cuadrícula visual .jpg) que evalúa la idoneidad para la instalación de aerogeneradores basándose en reglas predefinidas:

- Rojo (No Disponible): La celda tiene más del 25% de solapamiento con la capa de exclusión eólica.
- Naranja (Disponible, Viento Bajo): La celda es viable (menos del 25% de exclusión) pero los vientos son desfavorables (inferiores a 5.5 m/s).
- Verde (Disponible, Viento Favorable): La celda es viable y presenta vientos favorables.

Además, la capa es interactiva. Al hacer clic sobre cualquiera de las celdas de la cuadrícula, se despliega una ventana emergente (*popup*) con la "Información de la celda" específica. Este *popup* muestra los datos detallados que justifican su estado, incluyendo:

- id (identificador de la celda)
- viento medio
- WTG (viabilidad del aerogenerador)
- aves observadas
- Especies
- Cobertura (porcentaje de exclusión)

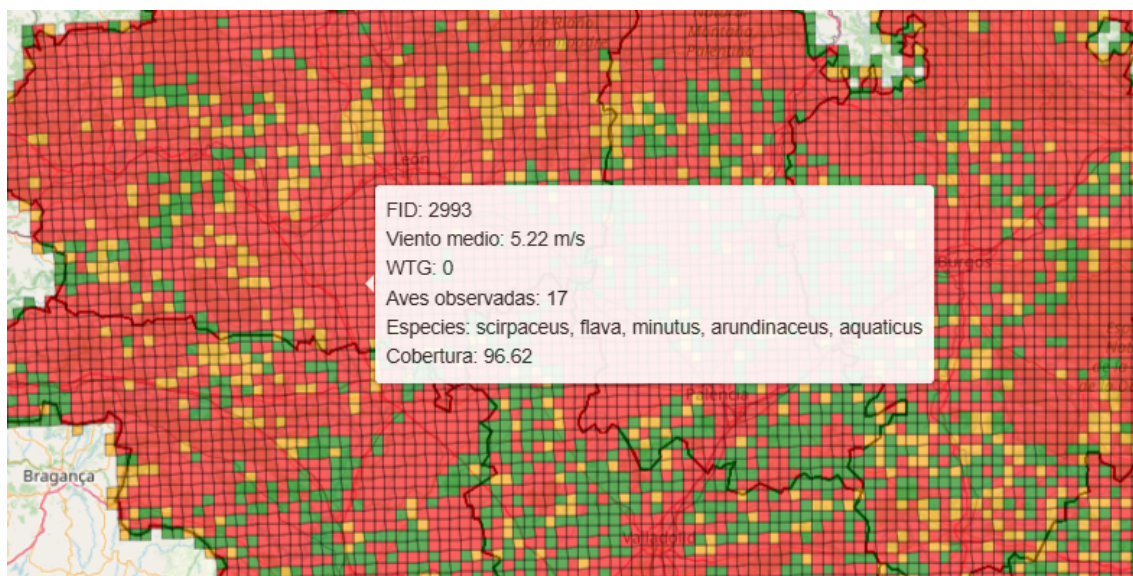


Fig.11. Cuadrícula mesoescalar

## 2.6 Lista de Cámaras (Gestión de Dispositivos)

**Objetivo:** Proveer un catálogo centralizado de los dispositivos de monitorización (cámaras) conectados a la plataforma.

Esta interfaz adapta su funcionalidad según el perfil del usuario:

### 2.6.1 Usuario No Registrado:

La vista muestra una galería de las cámaras que han sido registradas como "**Públicas**".

Cámaras Públicas

AIR INSTITUTE  
N/A  
camara\_air



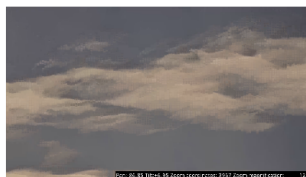
Fig.12. Cámaras públicas

## 2.6.2 Usuario Registrado

El usuario ve una vista unificada que incluye tanto las cámaras Públicas como las cámaras Privadas que él mismo ha registrado. Además, esta vista habilita la funcionalidad de "Añadir Cámara".

Cámaras Públicas Cámaras Privadas

AIR INSTITUTE  
N/A  
camara\_air



+ Añadir Cámara

Fig.13. Cámaras privadas

## 2.7 Formulario de Registro de Cámara

Al seleccionar "Añadir Cámara", se despliega una ventana modal que permite al usuario registrado configurar un nuevo dispositivo. El formulario solicita la siguiente información:

- Nombre de la Cámara: Un identificador legible (ej. "Nido Gredos").
- Ubicación: Descripción textual de la localización.
- URL de la Cámara: La dirección del stream (ej. rtsp://...).
- Tipo de Fuente: Un selector para definir el protocolo del stream (RTSP, RTMP, HLS, YouTube).
- Latitud y Longitud: Coordenadas GPS para posicionar la cámara en el mapa principal.
- Estado Cámara: Un interruptor (Switch) para definir si la cámara está "Activa" o "Inactiva".
- Datos de Almacenamiento: Información sobre el estado del almacenamiento del dispositivo Edge.
- Visibilidad de la cámara: Un interruptor (Switch) crucial que define si la cámara será "Pública" (visible para todos) o "Privada" (visible solo para el usuario que la registra).

Fig.14. Registro de nueva cámara

## 2.8 Vista de Detalle de Cámara (Datos y Analíticas)

El acceso a la vista de detalle de una cámara es una funcionalidad exclusiva para los usuarios registrados, solo un usuario que ha iniciado sesión puede acceder a esta pantalla de análisis avanzado, ya sea seleccionando una cámara pública o una de sus propias cámaras privadas.

Una vez dentro, la interfaz presenta un panel completo que combina la telemetría del dispositivo con los resultados analíticos generados en el Edge.

### 2.8.1 Datos del Dispositivo

En la parte superior, se muestra un resumen de los datos estáticos de la cámara, proporcionando contexto sobre el estado y configuración del dispositivo. Esta información incluye:

- Datos del vídeo
- Datos GPS (ej. "No disponible")
- Estado cámara (ej. "pending")
- Datos de almacenamiento (ej. "No disponible")
- Visibilidad (ej. "Pública")

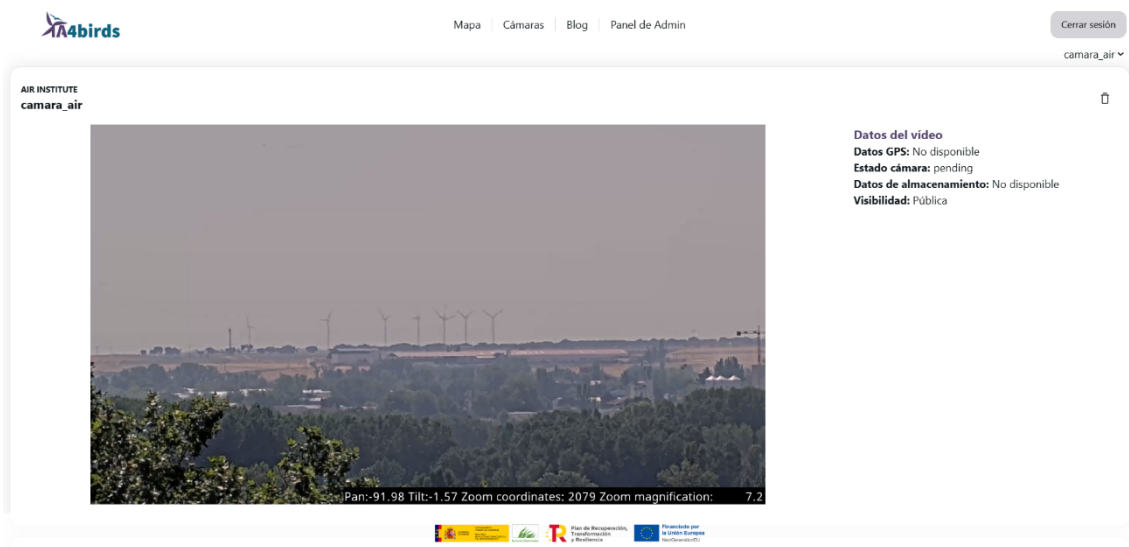


Fig.15. Visor de cámara en directo

### 2.8.2 Análisis de Detecciones

El núcleo de esta vista se divide en dos niveles de información:

- Detecciones por Frame: Se presenta la "Información Detallada de Especies", que es un desglose granular de las detecciones en crudo reportadas por el dispositivo Edge. Esta tabla permite un análisis forense, mostrando cada detección individual con sus coordenadas y clasificación (ej. 13AVE002X1: 1817, Y1: 10131026.2 px²halcon peregrino-117.7%).
- Agregado por Especies: Para un resumen gerencial, se muestra la tabla de "Información por Especies". Esta vista agrega los datos, mostrando un conteo total por especie para esa cámara. Incluye el "Nombre Común" (ej. buitres\_negro, halcon\_peregrino), el total de "Detecciones" y la marca temporal de la "Última Detección".

### 2.8.3 Visualización de Mapas de Calor

Finalmente, la interfaz ofrece la herramienta de análisis visual más potente: el mapa de calor (heatmap). Esta visualización representa gráficamente las zonas de mayor actividad y tráfico de aves frente a la cámara. Junto al heatmap, se muestran las estadísticas asociadas, permitiendo

al investigador interpretar rápidamente los patrones de vuelo y las áreas de mayor interés sin necesidad de revisar los datos en crudo.

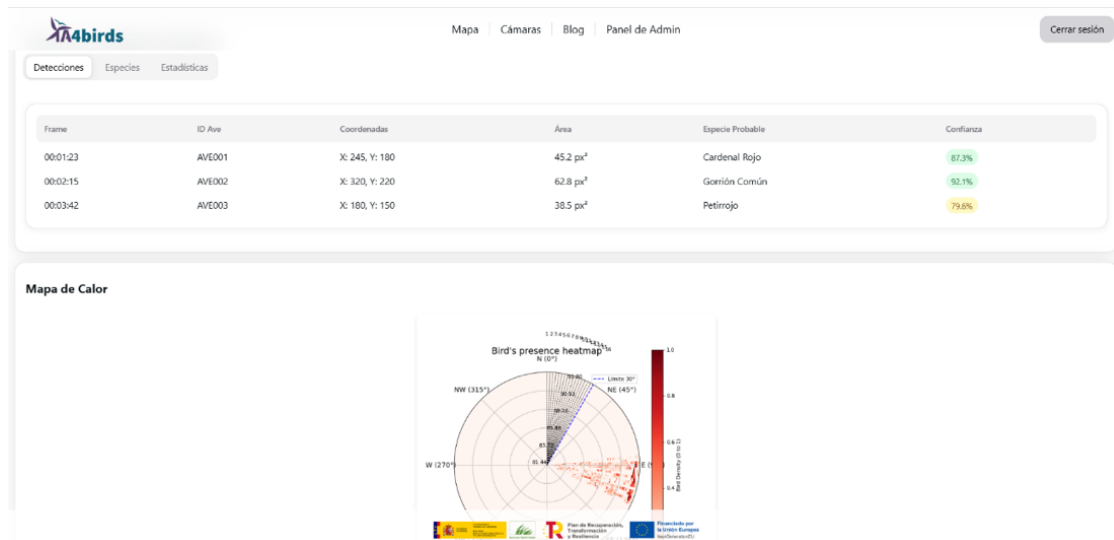


Fig.16. Mapa de calor

## 2.9 Vista del Blog (Divulgación y Noticias)

**Objetivo:** Ofrecer un canal de comunicación y divulgación accesible para todos los usuarios, donde se publican las últimas noticias, artículos de interés y explicaciones detalladas sobre los avances y funcionalidades del proyecto IA4Birds.

**Descripción y Funcionalidad:** Esta interfaz es una sección pública de la plataforma, accesible para usuarios no registrados y registrados por igual. Está diseñada como un blog tradicional donde se presentan los contenidos de forma clara y cronológica.

Los usuarios pueden navegar y leer artículos que explican en profundidad los objetivos del proyecto. Por ejemplo, se publican entradas que detallan el funcionamiento de la plataforma y su aplicación práctica, como el análisis de "qué zonas son adecuadas para instalar parques eólicos", permitiendo así una mayor transferencia de conocimiento no solo a los gestores, sino también al público general interesado en la tecnología y el medio ambiente.

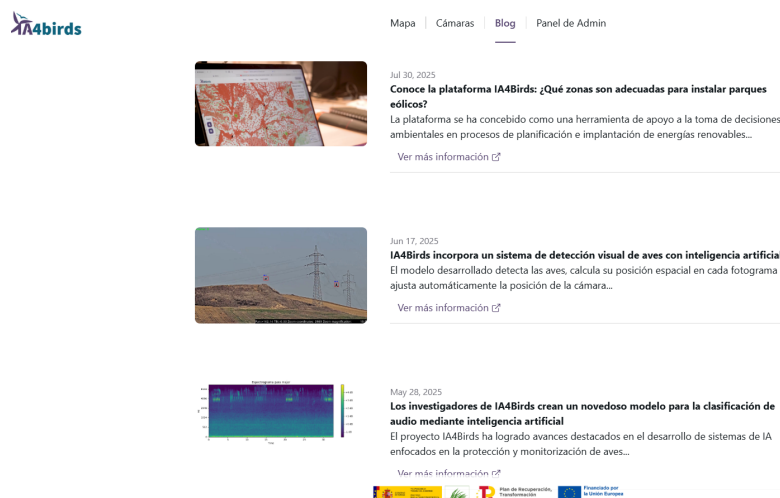


Fig.17. Blog

## 2.10 Panel de Administración

**Objetivo:** Proporcionar una vista técnica destinada a la validación y depuración de los flujos de datos de la plataforma.

**Descripción y Funcionalidad:** Esta interfaz es una sección de acceso restringido, visible únicamente para usuarios registrados (probablemente con rol de administrador). Su propósito no es la analítica visual, sino la inspección de los datos en crudo.

Como se observa en la imagen, el panel permite al administrador:

- Validar los Conectores: Muestra la respuesta JSON en crudo que devuelven los endpoints de ingesta de datos externos, como XenoCanto, eBird y la capa de Exclusión Mutua (IDECYL). Esto permite verificar que los servicios de terceros están respondiendo correctamente.
- Supervisar Datos de Cámaras: Muestra la información de los dispositivos (cámaras) registrados en el sistema.

Esta vista es una herramienta que permite a los administradores validar la integridad de los datos que se están ingiriendo antes de que estos sean procesados y mostrados en las visualizaciones públicas.

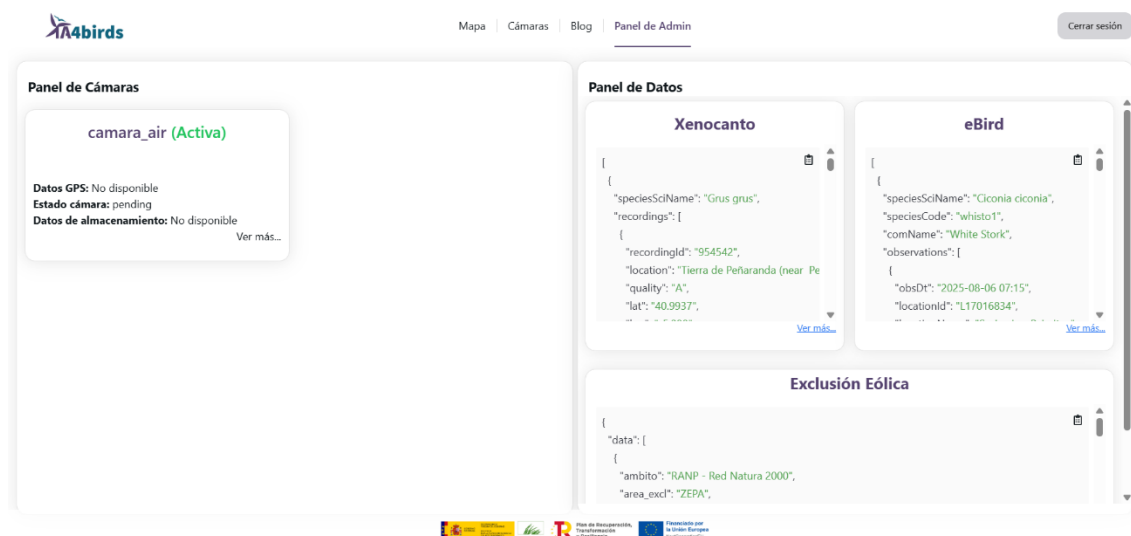


Fig.18. Panel de administración flujo de datos

## 2.11 Interfaces de Autenticación y Registro

El acceso a las funcionalidades avanzadas de la plataforma está protegido y requiere un proceso de autenticación.

### 2.11.1 Inicio de Sesión

Los usuarios que ya poseen una cuenta pueden acceder a la plataforma a través de un formulario de inicio de sesión estándar. Esta interfaz solicita el correo electrónico y la contraseña del usuario para autenticarlo y generar su *token* de sesión.

Inicio de Sesión

**Fig.19.** Inicio de sesión

## 2.11.2 Proceso de Registro

A diferencia de un registro público abierto, el alta en IA4Birds está mediatizada para asegurar que los usuarios tengan fines de investigación, institucionales o de gestión ambiental.

El proceso se basa en un formulario de "Solicitud de rol admin" donde el interesado debe proporcionar información detallada sobre su perfil:

- Email
- Organización
- Entidad (un selector con opciones como "Privada", "Pública", etc.)
- Ocupación (ej. "Project manager")
- Descripción (un campo de texto libre para justificar el motivo de la solicitud, ej. "Pruebas del funcionamiento de la plataforma").



[Mapa](#) | [Cámaras](#) | [Blog](#)

[Iniciar sesión](#)

Solicitar rol admin

Entidad:

Descripción  
Ejemplo: Somos la Junta de Castilla y León y queremos solicitar una cuenta para ver la información de las aves...

**Fig.20.** Proceso de registro

### 2.11.3 Flujo de Aprobación

Una vez enviado el formulario, la solicitud no crea una cuenta automáticamente. En su lugar, se envía un correo electrónico de notificación al correo corporativo de los administradores de IA4Birds.



Fig.21. Flujo de aprobación de acceso

Este correo contiene toda la información proporcionada por el solicitante. Un administrador revisa la solicitud y, si considera que el perfil es adecuado, hace clic en el enlace "Activar cuenta" contenido en el correo. Solo tras esta aprobación manual se crea la cuenta del usuario y se le notificará para que pueda establecer su contraseña e iniciar sesión.

## 3 Conclusión

El desarrollo de la plataforma IA4Birds representa un avance significativo en la integración de inteligencia artificial, análisis geoespacial e Internet de las Cosas (IoT) aplicados a la conservación de la biodiversidad. A través de un enfoque modular, seguro y escalable, se ha logrado construir un ecosistema tecnológico capaz de monitorizar, analizar y visualizar en tiempo real la actividad de las aves en entornos afectados por la implantación de parques eólicos, contribuyendo de forma directa a la planificación ambiental sostenible.

La arquitectura híbrida Edge + Cloud, junto con el uso de microservicios contenerizados y el despliegue automatizado mediante GitHub Actions, ha permitido garantizar la interoperabilidad entre componentes, la trazabilidad de los datos y la reproducibilidad de los resultados. Esta infraestructura, basada en Docker, Flask, Node.js, React y PostgreSQL, constituye una base sólida para el procesamiento eficiente de grandes volúmenes de información acústica y visual, asegurando la continuidad operativa y la flexibilidad del sistema frente a distintos escenarios de uso.

Las interfaces gráficas y visualizaciones interactivas desarrolladas —como el visualizador de mapa principal, las analíticas de cámaras y los mapas de calor— han transformado los datos de monitorización en información comprensible y accionable. Esto permite que tanto los gestores medioambientales como los investigadores puedan tomar decisiones fundamentadas sobre la ubicación óptima de infraestructuras eólicas y la protección de especies sensibles. Además, la

inclusión del blog divulgativo y los mecanismos de acceso público han contribuido a la transferencia del conocimiento científico y a la sensibilización de la sociedad sobre la importancia del equilibrio entre energía renovable y biodiversidad.

El cumplimiento de las normas de seguridad, autenticación y protección de datos mediante JWT, TLS y redes aisladas, junto con las pruebas exhaustivas de funcionalidad (unitarias, de integración y E2E), aseguran un sistema robusto, transparente y fiable. A ello se suma la adopción de estándares abiertos como OpenAPI/Swagger, que facilita la interoperabilidad y el acceso de otros investigadores o instituciones a los datos y APIs de la plataforma.