



E4.1. Capa común de Inteligencia Artificial y técnicas de visualización



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



Plan de Recuperación,
Transformación
y Resiliencia



Financiado por
la Unión Europea
NextGenerationEU

Contenido

E4.1. Capa común de Inteligencia Artificial y técnicas de visualización	1
1 Introducción	3
1.1 Objetivo	3
1.2 Alcance.....	3
2 Módulo de gestión de información, elaboración de directrices y criterios de planificación ambiental	3
2.1 Investigación en técnicas de visualización	3
2.1.1 Despliegue y validación en escenario controlado	4
2.2 Investigación y selección del framework más apropiado para las interfaces responsivas.....	4
2.2.1 Arquitectura en la nube y microservicios	5
2.3 Diseño e implementación de interfaces responsivas.....	6
2.3.1 Visualizador de Mapa Principal: Contexto Territorial y Avistamientos	6
2.3.2 Lista de Cámaras: Gestión de Dispositivos de Monitorización	7
2.4 Diseño del diagrama global de interacción, el modelo de vistas de visualización de datos y predicciones y diseño de la interfaz de usuario de la plataforma.	7
2.4.1 APIs y flujos de acceso a los datos.....	7
2.4.2 Integración de conectores en la capa de publicación.....	8
2.5 Diseño de sistema de roles y permisos a las diversas vistas de la plataforma.	9
2.5.1 Autorización Basada en Roles y Vistas.....	9
2.5.2 Aislamiento de Red.....	10
2.6 Implementación de capas de acceso y transporte seguras (autenticación, cifrado). 10	
2.6.1 Seguridad, cumplimiento y operatividad	10
2.6.2 API Coordinación	10
3 Tecnologías de análisis visualización de datos masivos	11
3.1 Técnicas de visualización de datos para la presentación de información	11
3.1.1 Visualizador de Mapa Principal	11
3.2 Estudio de diversas herramientas de software para la visualización de datos.....	14
3.2.1 OPENAPI/Swagger (interfaz documental/operativa)	15
3.2.2 API de Ingesta (ai4birds-ingest-service)	15
3.3 Diseño de técnicas de caché de visualizaciones para reducir consultas al Big Data y mejorar el rendimiento al acceder a la plataforma.	17
3.4 Pruebas unitarias y validación del framework de visualización de datos	18
4 Conclusión	19
5 Referencias.....	20

1 Introducción

1.1 Objetivo

Este documento corresponde al entregable:

E4.1 - Capa común de Inteligencia Artificial y técnicas de visualización.

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

A4.1 Módulo de gestión de información, elaboración de directrices y criterios de planificación ambiental y A4.2 Tecnologías de análisis de visualización de datos masivos para la predicción y protección de aves cuyo objetivo es investigar en diversas técnicas para la presentación de la información de la monitorización de aves, así como las predicciones y recomendaciones, que permita maximizar la accesibilidad, usabilidad y la experiencia de usuario. También en investigar en las diversas técnicas de minería de datos, Data Analytics basadas en Machine Learning y visualización de datos con el fin de construir los módulos de análisis de datos e interfaces gráficas de usuario para la representación de datos sobre las aves y su actividad de cada estimar predicciones y proteger las especies

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, la investigación aquí plasmada trata de presentar el trabajo realizado para la consecución de los objetivos del proyecto de diseñar una plataforma de IA explicable para optimizar la evaluación de espacios destinados a ser parques eólicos.

2 Módulo de gestión de información, elaboración de directrices y criterios de planificación ambiental

El objetivo de esta actividad es estudiar e Investigar en diversas técnicas para la presentación de la información de la monitorización de aves, así como las predicciones y recomendaciones, que permita maximizar la accesibilidad, usabilidad y la experiencia de usuario

2.1 Investigación en técnicas de visualización

El objetivo de esta actividad ha sido investigar, comparar y seleccionar las técnicas de visualización más adecuadas para representar la información generada por el motor cognitivo del proyecto IA4Birds. Este motor procesa datos obtenidos a partir de sensores IoT, fuentes de ciencia ciudadana y variables ambientales, transformándolos en indicadores que permiten planificar y evaluar estrategias de conservación de aves en zonas de influencia eólica.

Durante la fase de análisis se evaluaron diversos enfoques de visualización de datos con el fin de optimizar la comprensión, accesibilidad y usabilidad de la información para los distintos

perfiles de usuario definidos (gestores medioambientales, investigadores y público general). Se consideraron tanto representaciones espaciales como temporales, con el propósito de integrar en una misma interfaz el comportamiento de las aves, las predicciones del modelo y las condiciones ambientales del entorno.

El análisis comparativo permitió seleccionar una estrategia de visualización híbrida que combina la dimensión espacial (mapas y capas temáticas) con la dimensión temporal (cronogramas y series de tiempo). Esta aproximación facilita la exploración de los resultados desde distintas perspectivas —geográfica, temporal y funcional— y permite una interpretación intuitiva de los patrones detectados por los algoritmos de inteligencia artificial.

Como resultado, se definió una estructura visual basada en un visualizador geoespacial principal complementado con paneles de análisis temporal y estadístico, que en conjunto ofrecen una visión completa del estado del sistema y sus predicciones. Estas visualizaciones se implementaron bajo principios de interacción humano-computador (HCI) y experiencia de usuario (UX), con el objetivo de reducir la carga cognitiva, favorecer la exploración libre de datos y mejorar la toma de decisiones basada en evidencias.

2.1.1 Despliegue y validación en escenario controlado

Tras la selección de las técnicas de visualización, se procedió al despliegue del entorno de validación para evaluar el rendimiento, la estabilidad y la usabilidad de las interfaces desarrolladas.

El sistema se implementó sobre una infraestructura contenerizada coordinada mediante Docker Compose, compuesta por los servicios Nginx, frontend React, backend Node.js, coordinación e ingesta (Flask), MediaMTX, Mosquitto, Redis y tres instancias PostgreSQL. Las redes internas aíslan el tráfico de los contenedores, y solo el gateway publica los puertos 80/443. Los datos y configuraciones críticas se almacenan en volúmenes persistentes y se gestionan automáticamente los certificados TLS, garantizando la seguridad en el intercambio de información.

Sobre este entorno se realizaron pruebas unitarias y de integración, desarrolladas en Python y JavaScript, así como tests E2E (end-to-end) que abarcaron el flujo completo del usuario: desde el inicio de sesión hasta la visualización de los heatmaps y paneles de predicción. Además, se realizaron simulaciones de tráfico MQTT para validar la recepción, ingesta y persistencia de los datos en la base operativa.

Los resultados de la validación confirmaron la correcta interoperabilidad entre componentes, la respuesta fluida de las visualizaciones y la robustez del sistema frente a cargas concurrentes, garantizando la estabilidad y fiabilidad del módulo de visualización en un entorno controlado.

2.2 Investigación y selección del framework más apropiado para las interfaces responsivas

Uno de los objetivos de la A4 es implementar una plataforma full-stack que consume los resultados de la A3. **Machine learning para la detección y monitorización de aves** y los expone de forma controlada a los perfiles que toman decisiones. En paralelo al diseño del sistema de visualización, se llevó a cabo una investigación comparativa de frameworks reactivos con el propósito de seleccionar la tecnología más adecuada para la implementación de las interfaces responsivas de la plataforma IA4Birds.

Se evaluaron los principales entornos de desarrollo modernos —Angular 2, React.js y Vue.js— atendiendo a criterios de rendimiento, modularidad, soporte comunitario, compatibilidad con APIs RESTful y facilidad de integración con librerías geoespaciales y de visualización.

Tras las pruebas de rendimiento y usabilidad, se seleccionó React.js, combinada con Vite para el empaquetado y compilación optimizada. Esta elección se justificó por su reactividad nativa, su compatibilidad con Progressive Web Applications (PWA) y su amplia integración con librerías gráficas avanzadas, como Leaflet para mapas interactivos y Recharts para dashboards analíticos.

React se integró en la arquitectura full-stack de la plataforma, la cual consume los resultados generados por los módulos de inteligencia artificial (Actividad 3) y los expone de forma controlada a los diferentes perfiles de usuario definidos. La plataforma integra tres fuentes de información principales:

- i) monitorización con sensores IoT (cámara AXIS[1] y micrófonos audiomoth[2])
- ii) ciencia ciudadana (Xeno-canto[1], eBird[2])
- iii) contexto territorial (Mapa Eólico Ibérico [3] y zonas de exclusión de IDECYL[4]).

La arquitectura elegida es híbrida Edge + Cloud. En el borde (Jetson u otro dispositivo acelerado) se ejecuta la detección en tiempo real; en la nube residen ingesta, almacenamiento, agregación y visualización. El reparto resuelve restricciones de ancho de banda y latencia: los streams no se trasladan a la nube; lo que viaja son mensajes ligeros con resultados (detecciones, conteos, heatmaps) sobre MQTT. Este canal asíncrono tolera cortes de red y evita pérdidas.

El flujo de datos inicia en el Edge: el vídeo RTSP de la cámara se procesa con los modelos de detección. El dispositivo publica resultados estructurados en Mosquitto. En la nube, el servicio de ingesta, suscrito a los tópicos, valida y persiste en PostgreSQL. El backend expone estos datos por API y el frontend los representa en mapa, series y paneles.

El vídeo en vivo se captura en RTSP desde la cámara y se entrega al servicio MediaMTX, que lo redistribuye como HLS. El uso de HLS garantiza compatibilidad con navegadores sin extensiones.

2.2.1 Arquitectura en la nube y microservicios

La plataforma se despliega como microservicios contenerizados coordinados por redes internas. Nginx actúa como puerta de entrada, termina TLS y enruta al frontend (React, construida con Vite) y al backend (Node.js). El backend autentica, autoriza por rol y compone respuestas agregando datos de servicios internos.

El ai4birds-coordinate-service (Flask) aplica control de acceso: valida JWT, registra acceso y reenvía llamadas autorizadas al ai4birds-ingest-service.

El ingest-service escucha MQTT para recepcionar detecciones y heatmaps, persiste en la base operativa y ejecuta tareas periódicas con Celery para consultar Xeno-canto y eBird, obtener el Mapa Eólico Ibérico para coordenadas y cargar capas de IDECYL. La API está documentada con Flask-RestX/Swagger.

El almacenamiento se estructura en tres instancias de PostgreSQL: configuración (usuarios, cámaras, roles), ingesta operativa (segmentos, estadísticas, observaciones, referencias de heatmap) y análisis (vistas materializadas y agregados temporoespaciales). Redis hace de broker de Celery y de caché de resultados. MediaMTX recibe RTSP y publica HLS; su API permite dar de alta orígenes sin tocar componentes adyacentes.

2.3 Diseño e implementación de interfaces responsivas

Para alcanzar los objetivos de la plataforma IA4Birds y garantizar una interacción eficiente entre los usuarios y los datos generados por el sistema, se ha desarrollado un conjunto de visualizaciones e interfaces interactivas diseñadas bajo principios de diseño responsivo y tecnologías Progressive Web Applications (PWA). Estas interfaces permiten transformar los datos procedentes de los sensores IoT y de las fuentes de contexto (climáticas, territoriales y biológicas) en información comprensible, accesible y accionable, orientada a tres tipos de usuarios: gestores medioambientales, investigadores y público general.

2.3.1 Visualizador de Mapa Principal: Contexto Territorial y Avistamientos

La interfaz principal de la plataforma corresponde al Visualizador de Mapa, concebido como un entorno geoespacial interactivo que integra y representa las distintas fuentes de información territorial y ambiental del proyecto. Este componente constituye el punto de acceso central a la plataforma, ofreciendo una visión panorámica del territorio y la distribución de los avistamientos de aves.

El sistema se basa en una infraestructura cartográfica tipo Leaflet/Mapbox, que permite la carga dinámica de capas de información y la personalización de la visualización mediante un panel lateral de control.

Las capas del mapa se dividen en dos categorías:

Capas por defecto, siempre activas al cargar la interfaz:

- Exclusión eólica (IDECYL), que muestra las zonas restringidas para la instalación de parques eólicos en Castilla y León.
- Avistamientos de aves (DataBird), que integra registros de observación de plataformas de ciencia ciudadana como XenoCanto y eBird, unificando taxonomías y coordenadas de ubicación.

Capas opcionales, que pueden activarse de forma manual por el usuario:

- Recurso eólico, que habilita la consulta directa del Mapa Eólico Ibérico, permitiendo el análisis de la velocidad y distribución del viento en coordenadas seleccionadas.
- Cuadrícula mesoesalar, que superpone un mapa recomendador para la instalación de aerogeneradores, categorizando la idoneidad territorial mediante una codificación cromática:
 - Rojo: zona no disponible (más del 25% de solapamiento con áreas de exclusión).
 - Naranja: zona viable con viento insuficiente (menor a 5.5 m/s).
 - Verde: zona viable con viento favorable.

El mapa es completamente interactivo: el usuario puede desplazarse y hacer zoom (funciones pan/zoom), consultar parámetros eólicos y visualizar en tiempo real las celdas recomendadas para la instalación de turbinas. Al seleccionar una celda, se despliega un popup informativo con datos específicos como la velocidad media del viento, cobertura de exclusión, especies observadas, viabilidad del aerogenerador (WTG) y porcentaje de área afectada.

Esta interfaz cumple la función de integrar información ambiental, climática y biológica en un entorno visual unificado, permitiendo la toma de decisiones informada en el contexto de la planificación eólica y la conservación de aves.

2.3.2 Lista de Cámaras: Gestión de Dispositivos de Monitorización

Complementariamente, se ha implementado un módulo de gestión de dispositivos IoT que centraliza la información de las cámaras conectadas a la red de IA4Birds. Esta interfaz facilita el seguimiento, control y administración de los equipos instalados en campo, y adapta su funcionalidad según el perfil del usuario:

Para usuarios no registrados, se ofrece una galería pública con las cámaras declaradas como accesibles, mostrando únicamente su localización y características básicas.

Para usuarios registrados, la interfaz amplía el acceso a cámaras públicas y privadas, y habilita la funcionalidad de alta de nuevos dispositivos, permitiendo la configuración, monitorización y vinculación con los sistemas de almacenamiento de datos.

Ambas interfaces (el visualizador de mapa y la lista de cámaras) están integradas en un entorno de diseño responsivo y multiplataforma, optimizado para su uso en navegadores de escritorio y dispositivos móviles. Su desarrollo bajo el enfoque de Progressive Web Application (PWA) garantiza rendimiento, accesibilidad y actualización continua, permitiendo al usuario acceder a la monitorización de aves, análisis de datos y predicciones sin necesidad de instalaciones locales.

2.4 Diseño del diagrama global de interacción, el modelo de vistas de visualización de datos y predicciones y diseño de la interfaz de usuario de la plataforma.

El diseño del diagrama global de interacción y del modelo de vistas se ha orientado a garantizar una comunicación fluida entre los diferentes niveles de la arquitectura (Edge, Cloud, API y Frontend), asegurando la consistencia en la visualización y la trazabilidad de la información en todo el sistema.

La arquitectura funcional se basa en un modelo cliente-servidor multi-nivel, en el que los módulos de procesamiento, almacenamiento y visualización se integran bajo un enfoque microservicio, coordinado a través de redes seguras y servicios de mensajería asíncrona (MQTT).

2.4.1 APIs y flujos de acceso a los datos

La API de ingesta ofrece los recursos de explotación: segmentos filtrables por cámara, ventana y confianza; estadísticas agregadas por especie y franja horaria; descriptores de heatmap con `image_url` y `bounds`; resultados combinados de `Xeno-canto` y `eBird`; y datos de viento/exclusión para contexto. La cadena de responsabilidades queda separada: Edge detecta y publica; Ingesta valida, persiste y sirve; Coordinación asegura; BFF compone; Frontend presenta.

La API de coordinación es la puerta segura de uso por la interfaz. El frontend adjunta el JWT en cada llamada. El coordinador verifica token y permisos y, si procede, reenvía al servicio de ingesta. Esta consulta la base operativa y devuelve JSON. Con este patrón, al abrir el detalle de una cámara, la plataforma recupera detecciones del intervalo, estadísticas por especie y el descriptor del heatmap más reciente; el frontend los presenta en sus vistas.

2.4.2 Integración de conectores en la capa de publicación

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:

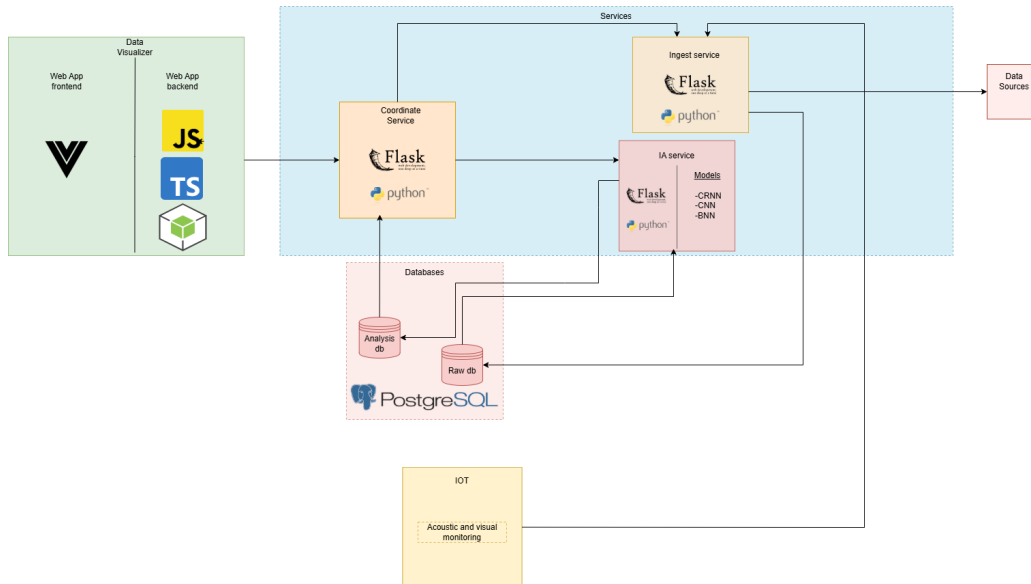


Fig.1. Arquitectura

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.

2.5 Diseño de sistema de roles y permisos a las diversas vistas de la plataforma.

El modelo de seguridad se basa en JSON Web Tokens (JWT) para gestionar la autenticación y la autorización.

El Backend en Node.js y TypeScript actúa como el servidor de autenticación. Cuando un usuario inicia sesión con sus credenciales, el backend las valida y, si son correctas, genera y firma un JWT. Este token, que contiene la identidad del usuario y su rol, se envía al cliente (frontend).

El cliente almacena este JWT de forma segura (normalmente en localStorage o httpOnly cookie). A partir de ese momento, para cada petición a una ruta protegida, el frontend debe adjuntar el token en la cabecera HTTP: Authorization: Bearer <token>.

Todas las peticiones protegidas son interceptadas por la API de Coordinación (Flask). Este servicio actúa como una pasarela de seguridad que:

1. Verifica la firma y la expiración del JWT para asegurar su validez e integridad.
2. Aplica la política de autorización basada en el rol.
3. Si el usuario tiene permiso, la API de Coordinación reenvía la petición al servicio interno correspondiente (como la API de Ingesta). Si no, devuelve un error (ej. 401 Unauthorized o 403 Forbidden).

2.5.1 Autorización Basada en Roles y Vistas

La plataforma define distintos niveles de acceso según el estado de autenticación y el rol del usuario:

Usuario No Autenticado (Público):

- Tiene acceso a las vistas públicas de la plataforma.
- Puede interactuar con el mapa principal y visualizar todas las capas de contexto (IDECYL, Mapa Eólico, ciencia ciudadana).
- Puede ver la información y el streaming de cámaras que estén registradas como "públicas".
- Puede acceder a la vista del blog.

Usuario Autenticado:

- Hereda todos los permisos del usuario público.
- Puede registrar nuevas cámaras, tanto "públicas" (visibles para todos) como "privadas" (visibles solo para él).
- Puede acceder a la vista de detalle de sus cámaras (públicas o privadas), donde puede consultar datos sensibles como las estadísticas detalladas de aves y los mapas de calor (heatmaps) generados.
- Tiene acceso a un panel de administración en el que puede visualizar los datos en crudo que se ingesta de los datos, que son la fuente para las visualizaciones agregadas del mapa.

2.5.2 Aislamiento de Red

La seguridad se refuerza a nivel de red. Todos los servicios internos sensibles (API de Ingesta, API de Coordinación, PostgreSQL, Redis, Mosquitto) no están expuestos a Internet. Residen exclusivamente en redes internas de Docker.

La única exposición externa se limita al gateway Nginx (puertos 80/443), que actúa como proxy inverso, gestiona TLS (HTTPS) y enruta el tráfico únicamente a los servicios diseñados para recibirlo (el Frontend y el Backend de Node.js).

2.6 Implementación de capas de acceso y transporte seguras (autenticación, cifrado).

La seguridad del sistema IA4Birds se ha diseñado atendiendo a los principios de confidencialidad, integridad y disponibilidad (CIA), asegurando tanto el flujo de datos entre componentes como el acceso controlado a la información por parte de los usuarios.

2.6.1 Seguridad, cumplimiento y operatividad

Todas las comunicaciones externas e internas se cifran mediante TLS 1.3, garantizando la privacidad y autenticidad de los intercambios de datos entre servicios. La autenticación se gestiona mediante JSON Web Tokens (JWT), que incluyen información del usuario y su rol, firmados con claves seguras almacenadas en el backend (Node.js/TypeScript).

Las políticas de autorización se aplican según el principio de mínimo privilegio, limitando cada operación al rol asignado: público, usuario registrado o administrador. Para cada solicitud HTTP, la API de Coordinación (Flask) verifica la validez del token, la expiración y los permisos asociados antes de reenviar la petición a la capa de Ingesta.

Se aplican medidas de resiliencia y mitigación de riesgos mediante:

- Rate limiting: previene ataques de denegación de servicio (DoS).
- Circuit breakers: aíslan temporalmente servicios externos lentos o con errores.
- Caché Redis: reduce la latencia y el coste de consultas repetitivas.
- Colas MQTT/Celery: amortiguan picos de tráfico y mantienen el procesamiento asíncrono ante cortes de red.

El cumplimiento de buenas prácticas de seguridad sigue las recomendaciones de OWASP (Open Web Application Security Project), garantizando la protección frente a vulnerabilidades comunes (inyecciones, XSS, CSRF, etc.).

Además, se aplican estrategias de minimización de datos y ocultación espacial para limitar la exposición de coordenadas precisas cuando el usuario carece de permisos avanzados.

2.6.2 API Coordinación

La API de Coordinación actúa como la única puerta de entrada segura a los datos de la plataforma. Su función principal es aplicar la seguridad antes de permitir el acceso a los datos sensibles.

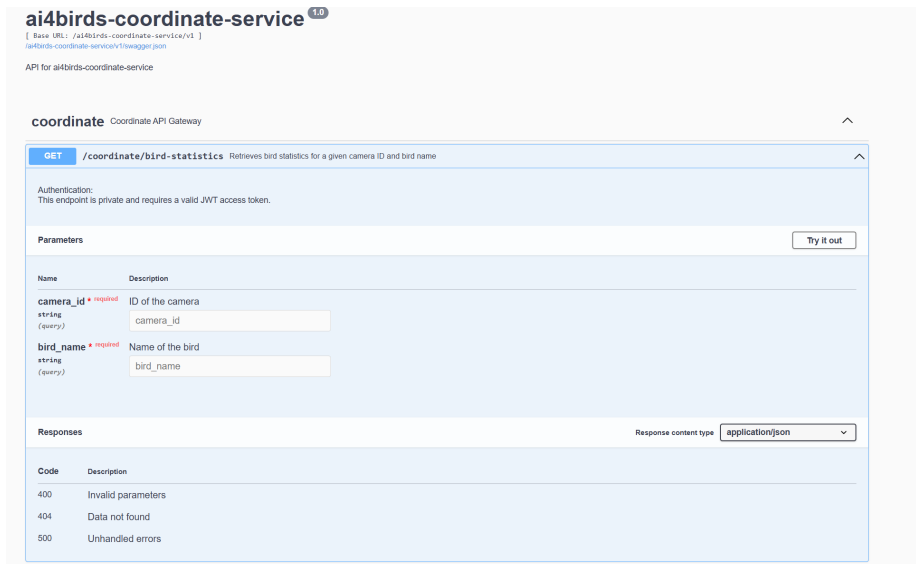


Fig.2. API de coordinación

Como se observa en su documentación de Swagger, la mayoría de los endpoints están protegidos y marcados con un icono de candado. Esto significa que cada petición debe incluir un token JWT (JSON Web Token) válido en la cabecera Authorization. Este servicio valida el token, verifica los permisos del rol del usuario y, si la autorización es correcta, actúa como una pasarela (gateway) reenviando la petición a la API de Ingesta.

3 Tecnologías de análisis visualización de datos masivos

El objetivo de esta actividad es Investigar en las diversas técnicas de minería de datos, Data Analytics basadas en Machine Learning y visualización de datos con el fin de construir los módulos de análisis de datos e interfaces gráficas de usuario para la representación de datos sobre las aves y su actividad de cada estimar predicciones y proteger las especies.

3.1 Técnicas de visualización de datos para la presentación de información

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).

3.1.1 Visualizador de Mapa Principal

Se busca 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.

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.

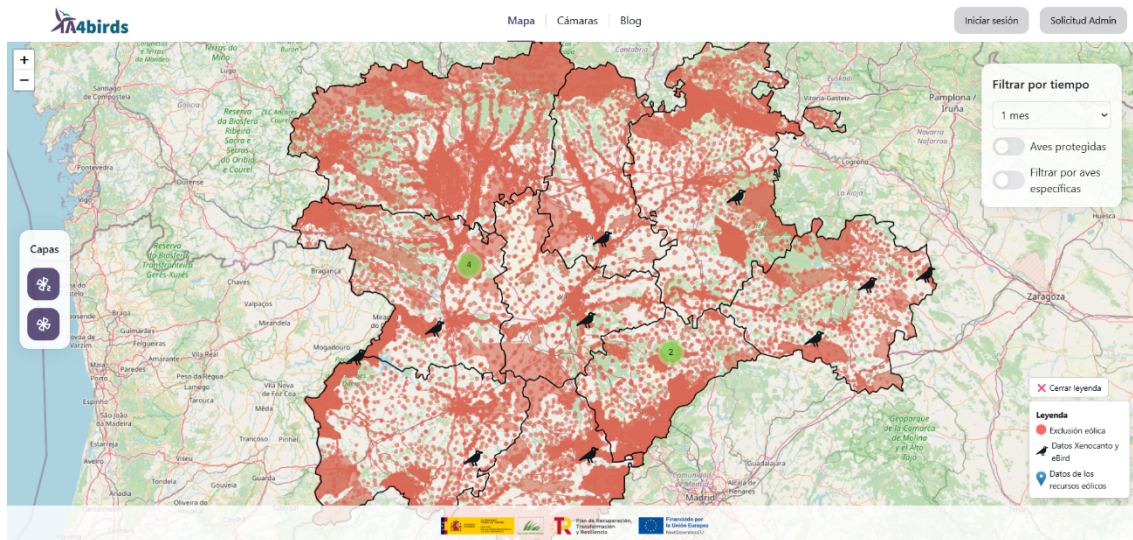


Fig.3. Pantalla principal

Las capas de información se dividen en:

Capas por Defecto (siempre visibles al cargar):

- Exclusión Eólica (IDECYL): Muestra las zonas de exclusión ambiental de Castilla y León.
- Avistamientos de Aves (DataBird): Muestra los registros de ciencia ciudadana, cruzando los datos de XenoCanto y eBird (endpoint dataBird) para unificar la taxonomía.

Capas Opcionales (activables por el usuario):

- Recurso Eólico: Activa la funcionalidad de consulta del Mapa Eólico Ibérico.
- Cuadrícula Mesoescalar: Activa un mapa recomendador para la instalación de aerogeneradores.

Funcionalidad

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

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.4. Perfil medio diario velocidad del viento en m/s

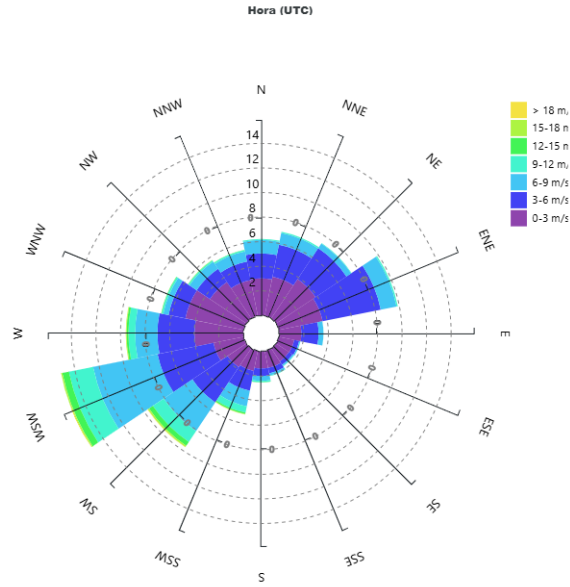


Fig.5. Rosa de los vientos

Distribución de Frecuencia y Densidad en función de la Velocidad

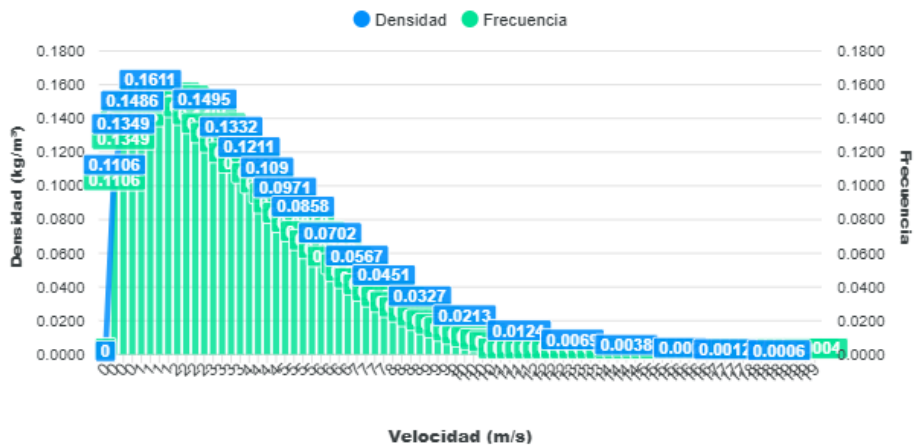


Fig.6. Distribución de frecuencia y densidad de la velocidad.

Perfil vertical medio de la velocidad del viento

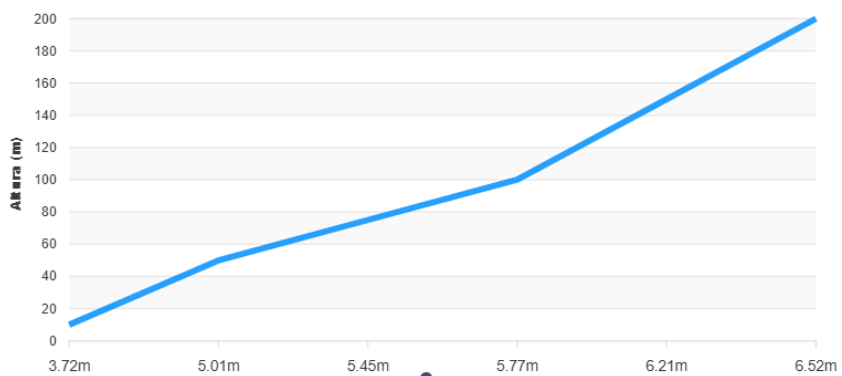


Fig.7. Perfil vertical medio de la velocidad del viento

Consulta de Cuadrícula Mesoescalar (Botón 2)

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.

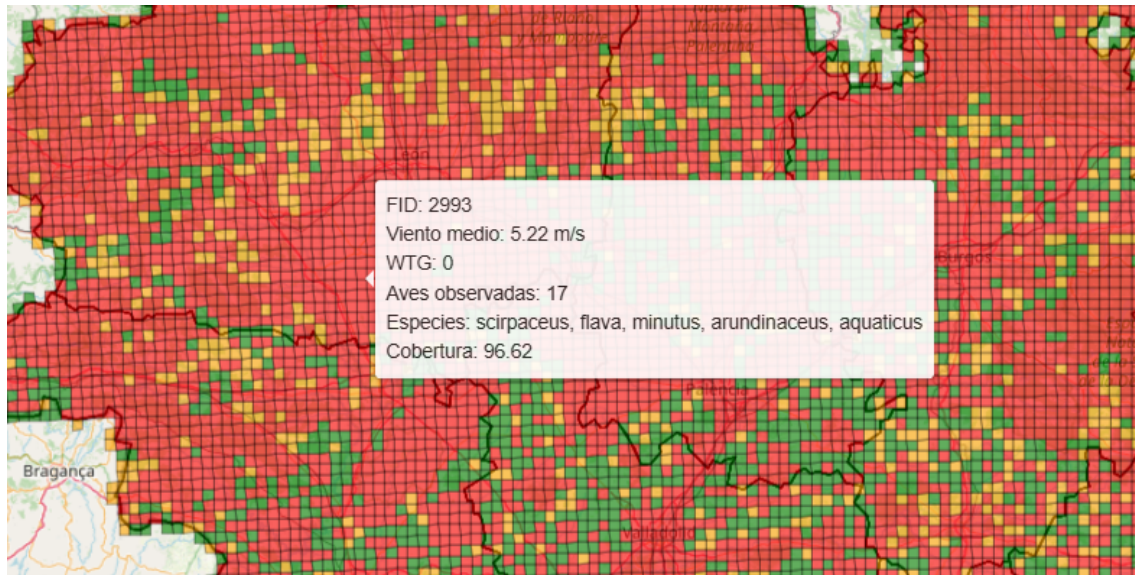


Fig.8. Cuadrícula mesoescalar

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)

3.2 Estudio de diversas herramientas de software para la visualización de datos

Con el fin de garantizar la eficiencia, interoperabilidad y calidad visual de los resultados mostrados en la plataforma IA4Birds, se ha realizado una investigación exhaustiva de herramientas de software y entornos de desarrollo orientados a la visualización, documentación y explotación de datos procedentes de las distintas capas del sistema.

Este estudio tuvo como objetivo identificar las soluciones tecnológicas más adecuadas para representar y compartir grandes volúmenes de información ambiental y biológica de forma dinámica, segura y estandarizada. Se evaluaron plataformas de desarrollo, librerías de visualización, frameworks de documentación de APIs y sistemas de gestión de datos geoespaciales, valorando aspectos como la escalabilidad, la modularidad, la facilidad de integración y la adopción de estándares abiertos.

Entre los criterios de selección se consideraron especialmente la compatibilidad con arquitecturas de microservicios, el soporte para formatos JSON y GeoJSON, la capacidad de generar visualizaciones interactivas de tipo dashboard o mapa, y la posibilidad de automatizar la documentación de los endpoints y la validación de las peticiones API.

Como resultado de esta investigación, se definió una infraestructura tecnológica de referencia para IA4Birds basada en herramientas de documentación activa de APIs (OpenAPI/Swagger), servicios RESTful y frameworks geoespaciales y analíticos capaces de manejar datos tanto en tiempo real como históricos. Esta combinación asegura la trazabilidad de las fuentes, la transparencia de los procesos y la facilidad de integración con los módulos de análisis y predicción basados en inteligencia artificial desarrollados en las fases anteriores del proyecto.

3.2.1 OPENAPI/Swagger (interfaz documental/operativa)

Naturaleza: Es la especificación estándar (formalmente OAS 3) para describir y definir APIs REST. Swagger es el conjunto de herramientas más popular que implementa esta especificación, incluyendo la interfaz de usuario (Swagger UI).

Función

- Documentación viva: Documenta automáticamente todos los endpoints de las APIs de Ingesta y Coordinación.
- Interfaz operativa: Facilita pruebas interactivas directamente desde el navegador, permitiendo a los desarrolladores validar la lógica de la API.
- Herramienta de desarrollo: Permite la validación automática de peticiones y la generación de código cliente en diferentes lenguajes.

Características

- Define formalmente los esquemas de petición/respuesta (qué datos enviar y esperar).
- Documenta los esquemas de seguridad (ej. cómo usar el JWT).
- Especifica todos los códigos de estado posibles (200, 401, 404, etc.) y provee ejemplos.

3.2.2 API de Ingesta (ai4birds-ingest-service)

En el stack de la nube, MediaMTX convierte el RTSP a HLS para el consumo web. El servicio ai4birds-ingest-service (Flask) recibe los mensajes MQTT y sirve la API REST principal. La seguridad se delega en ai4birds-coordinate-service (Flask), que actúa como pasarela validando tokens JWT. Finalmente, un backend en Node.js y TypeScript orquesta la lógica de aplicación y la autenticación, mientras que un frontend en React (Vite) presenta los datos al usuario.

La API de Ingesta es el motor de datos de la plataforma; es la que se comunica con la base de datos y con los conectores de terceros. Sus endpoints están organizados por espacios de nombres (namespaces) para agrupar funcionalidades:

- ns_xenocanto (GET /xenocanto): Consulta y recupera registros de grabaciones de aves de la base de datos de XenoCanto[7].
- ns_ebird (GET /ebird): Consulta y recupera registros de avistamientos de aves de la base de datos de eBird[8].
- ns_dataBird (GET /dataBird): Expone un endpoint que combina los resultados de XenoCanto y eBird en una única respuesta.
- ns_windmap (POST /windmap): Recibe coordenadas (lat, lon, z) y devuelve datos del Mapa Eólico Ibérico, como el perfil de viento, la rosa de vientos y datos Weibull.
- ns_exclusionmap: Ofrece varios endpoints para consumir los datos de exclusión eólica de IDECYL:
 - o POST /: Devuelve los datos en un formato JSON paginado.
 - o GET /all: Devuelve el conjunto de datos completo en un solo JSON.
 - o GET /zip: Proporciona un fichero .zip descargable con los datos.
 - o POST /stream-exclusion-data: Permite el streaming de datos por lotes.
- ns_sensitivity (GET /sensitivity): Recupera la información de las capas de sensibilidad de aves en Castilla y León.

Datos Operativos (Ingesta desde el Edge):

- ns_device_status: Gestiona el estado de los dispositivos Edge.
 - o POST /: Permite al Edge reportar un nuevo estado (almacenamiento, GPS, etc.).
 - o GET /latest: Obtiene el último estado reportado por un dispositivo.
 - o GET /health: Comprueba el estado de salud del sistema.
- ns_segment_data (GET /segment-data): Es uno de los endpoints principales; recupera los datos de detecciones de aves (segmentos), permitiendo filtrar por camera_id.
- ns_heatmap_data (GET /heatmap-data): Recupera los metadatos del mapa de calor (heatmap) más reciente generado para una camera_id específica.
- ns_bird_statistics: Proporciona estadísticas agregadas:
 - o GET /: Devuelve estadísticas filtradas por camera_id y por bird_name (especie).
 - o GET /by-camera: Devuelve todas las estadísticas agrupadas por una camera_id.
- ns_heatmap_files: Gestiona el acceso a los ficheros de imagen de los heatmaps.
 - o GET /files: Lista todos los ficheros de heatmap disponibles.
 - o GET /download/<filename>: Permite la descarga directa de un fichero de imagen de heatmap específico (ej. un .png).

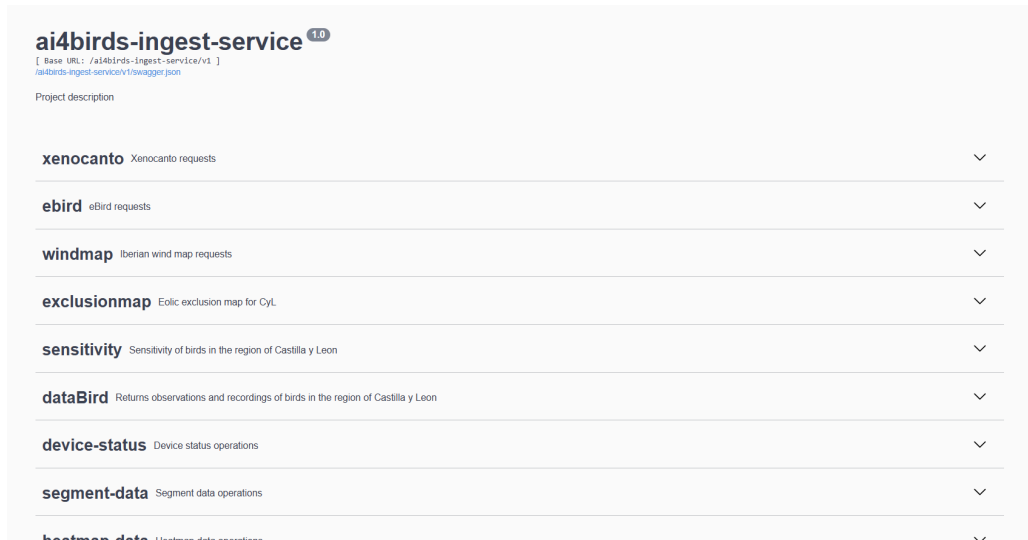


Fig.9. Servicio de ingesta

Para emplear estos endpoints desde la interfaz de Swagger, solo es necesario seleccionar el que se desea usar y presionar “Try it out”, seguido de “Execute”. Dentro de esta visual se encuentran ejemplos de uso y las posibles respuestas. El usuario puede modificar el valor de los “Parameters” (como camera_id o bird_name) para personalizar la petición.

3.3 Diseño de técnicas de caché de visualizaciones para reducir consultas al Big Data y mejorar el rendimiento al acceder a la plataforma.

El IA4Birds Coordinate Service, implementado sobre el framework Flask, constituye el punto de control y gestión del flujo de información dentro de la arquitectura de servicios del sistema. Este componente actúa como puerta de entrada segura a la plataforma, aplicando mecanismos de autenticación y autorización mediante JSON Web Tokens (JWT). Cada solicitud entrante es validada antes de ser procesada, registrando las trazas de acceso y redirigiendo únicamente aquellas llamadas que cumplen con los permisos establecidos hacia el IA4Birds Ingest Service. De este modo, se garantiza la integridad, trazabilidad y seguridad de todas las operaciones, manteniendo un control centralizado sobre el acceso a los recursos y servicios del ecosistema IA4Birds.

El Ingest Service funciona como el núcleo de recepción y procesamiento de datos. Este componente escucha en tiempo real mensajes a través del protocolo MQTT, lo que le permite integrar de forma continua los flujos de datos provenientes de los sensores desplegados —como cámaras IoT, micrófonos y dispositivos de detección ambiental—. A partir de esta información, el servicio gestiona la persistencia de detecciones, metadatos y heatmaps en la base de datos operativa. Además, mediante la integración del sistema de tareas distribuidas Celery, el servicio ejecuta procesos periódicos automatizados que incluyen:

- La consulta y sincronización con fuentes externas de ciencia ciudadana como XenoCanto y eBird, que aportan observaciones acústicas y visuales de aves en la región de estudio.

- La obtención dinámica del Mapa Eólico Ibérico, con el fin de asociar datos de recurso eólico (velocidad y dirección del viento) a las coordenadas geográficas de cada observación.
- La actualización automática de capas geospaciales de IDECYL, incorporando información ambiental relevante como zonas de exclusión eólica y áreas protegidas.

Toda la API del sistema está documentada mediante Flask-RESTX, integrando una interfaz Swagger que facilita la exploración, prueba y validación de los endpoints. Esto permite a los desarrolladores e investigadores interactuar fácilmente con los servicios del sistema, consultar la estructura de los datos y automatizar procesos analíticos o de ingestión.

En cuanto a la infraestructura de almacenamiento, el sistema se apoya en una arquitectura PostgreSQL distribuida en tres instancias independientes, cada una optimizada para un propósito específico:

1. Instancia de Configuración: almacena la información estructural de la plataforma, incluyendo usuarios, roles, credenciales y metadatos de cámaras, garantizando una gestión segura y segmentada de los accesos.
2. Instancia de Ingesta Operativa: contiene los datos brutos generados por los sensores (segmentos de vídeo, estadísticas, observaciones acústicas, detecciones visuales y referencias a mapas de calor), sirviendo como la base de datos activa para la explotación inmediata.
3. Instancia de Análisis: se especializa en la generación de vistas materializadas y agregados temporoespaciales, que permiten realizar análisis avanzados sobre la distribución de especies, frecuencia de avistamientos, comportamiento de vuelo y patrones de migración.

El sistema emplea Redis como broker de Celery y como caché de resultados, optimizando la latencia en la ejecución de tareas distribuidas y acelerando las consultas frecuentes sobre datos de gran volumen. Este enfoque de almacenamiento en memoria reduce significativamente el tiempo de respuesta y aumenta la eficiencia del sistema frente a operaciones concurrentes o de alta demanda.

Por otro lado, el componente MediaMTX gestiona la recepción y retransmisión de flujos de vídeo desde los dispositivos IoT desplegados en campo. Este servicio recibe señales RTSP (Real Time Streaming Protocol) desde las cámaras y las convierte en flujos HLS (HTTP Live Streaming), garantizando la compatibilidad con navegadores y reproductores web. Además, su API expone endpoints REST que permiten registrar o eliminar orígenes de vídeo dinámicamente, sin necesidad de reiniciar servicios ni modificar otros componentes del sistema. Esta característica facilita la escalabilidad del ecosistema IA4Birds, permitiendo incorporar nuevos sensores o cámaras sin interrumpir el funcionamiento operativo de la plataforma.

3.4 Pruebas unitarias y validación del framework de visualización de datos

Con el objetivo de garantizar la estabilidad, fiabilidad y usabilidad de la plataforma IA4Birds, se ha llevado a cabo una batería de pruebas funcionales exhaustivas orientadas a verificar el comportamiento de las interfaces de usuario (UI/UX) y la correcta implementación de los flujos de interacción previstos para cada tipo de usuario (público, registrado y administrador).

Estas pruebas se han diseñado siguiendo un enfoque metodológico de ingeniería de software, evaluando los componentes críticos de la aplicación —mapas interactivos, gestión de dispositivos IoT, autenticación, seguridad y acceso a datos—, bajo distintos escenarios de uso.

Cada prueba documenta el estado de funcionamiento, la salida esperada y las observaciones obtenidas durante la validación, sirviendo como evidencia de la funcionalidad alcanzada y de la conformidad con los requisitos técnicos definidos en la arquitectura del sistema.

Los resultados de las pruebas se muestran en una tabla en el ***E4.3 Integración de componentes y despliegue de la plataforma en un escenario controlado***

4 Conclusión

El proyecto IA4Birds ha desarrollado una plataforma completa para la monitorización ambiental que combina sensores IoT, inteligencia artificial y análisis geoespacial con el fin de predecir y proteger la biodiversidad de las aves.

Gracias a una arquitectura híbrida que combina el procesamiento en el borde (Edge) y en la nube (Cloud), y al uso de microservicios y APIs seguras, el sistema puede gestionar grandes volúmenes de datos visuales y sonoros en tiempo real de forma ágil y escalable.

Las herramientas de visualización avanzada, como los mapas interactivos y los paneles de control dinámicos, permiten transformar los datos en información clara y útil para los gestores ambientales y planificadores de parques eólicos, facilitando la toma de decisiones.

Además, las pruebas técnicas exhaustivas y el desarrollo en formato Progressive Web Application (PWA) garantizan un sistema estable, accesible desde cualquier dispositivo y compatible con fuentes externas de datos como Xeno-Canto, eBird o IDECYL.

5 Referencias

- [1]. AXIS Q6225-LE PTZ Camera Axis Communications, dirección: <https://www.axis.com/es-es/products/axis-q6225-le>
- [2]. <https://www.openacousticdevices.info/audiomoth>
- [3]. Xeno-Canto – base de datos de grabaciones de cantos de aves: <https://xeno-canto.org/>
- [4]. E-Bird Ciencia ciudadana. <https://ebird.org/home>
- [5]. APECyL – Asociación de Promotores de Energía Eólica de Castilla y León: <https://apecyl.com/>
- [6]. IDECyL – Infraestructura de Datos Espaciales de Castilla y León, zonas de exclusión eólica: <https://idecyl.jcyl.es/geonetwork/static/api/records/spagobcyltemenerenexceol>
- [7]. API :: xeno-canto., dirección: <https://xeno-canto.org/explore/api> (visitado 17-05-2024).
- [8]. eBird – base de datos de observaciones de aves con API de acceso: <https://documenter.getpostman.com/view/664302/S1ENwy59>