

GeoBio Colombia: WebApp de Geolocalización y Catálogo de Fauna de la Región del Pacífico



Gengis Salvador Hidalgo Guevara

Universidad del Valle
Escuela de Ingeniería en Sistemas y Computación
Ingeniería en Sistemas y Computación
Tuluá - Valle del Cauca
Agosto - Diciembre 2024

GeoBio Colombia: WebApp de Geolocalización y Catálogo de Fauna de la Región del Pacífico



Gengis Salvador Hidalgo Guevara

Codigo: 1866167 - 3743

gengis.hidalgo@correounivalle.edu.co

Director:

Msc. Miguel Ángel Askar Rodriguez

Universidad del Valle
Escuela de Ingeniería en Sistemas y Computación
Ingeniería en Sistemas y Computación
Tuluá - Valle del Cauca
Agosto - Diciembre 2024

Proyecto de grado presentado por
Gengis Salvador Hidalgo Guevara
Como requisito parcial para la obtención del título de Ingeniería en Sistemas

Msc. Miguel Ángel Askar Rodríguez
Director

Jurado

Jurado

Quiero dar un agradecimiento principalmente a Dios por darme la fuerza y la constancia para seguir adelante. A mis padres, que con su espíritu emprendedor y actitud siempre positiva, han sido mi mayor fuente de inspiración. Gracias también a los profesores de la universidad por su esfuerzo y dedicación al compartir su conocimiento y experiencia, que han sido una parte importante de mi formación. Un agradecimiento especial al profesor Miguel Askar, mi director, por su guía y apoyo durante todo el proceso. A mis compañeros, con quienes compartí este camino académico y personal. Y por último, gracias a la Universidad del Valle por darme la oportunidad de aprender y crecer durante estos cinco años, una etapa que siempre llevaré conmigo.

Tabla de Contenido

1. Introducción	9
1.1. Formulación del Problema	9
1.2. Descripción del Problema	9
1.3. Objetivos	10
1.3.1. Objetivo General	10
1.3.2. Objetivos Específicos y Resultados Esperados	10
1.4. Metodología	10
1.4.1. Recolección de Requisitos (2 semanas)	10
1.4.2. Diseño del Sistema (4 semanas)	11
1.4.3. Desarrollo e Implementación (12 semanas)	11
1.4.4. Despliegue (1 semana)	11
1.4.5. Pruebas y Validación (2 semanas)	11
1.5. Información sobre los capítulos	12
2. Justificación y alcance del proyecto	13
2.1. Justificación	13
2.2. Alcance del Proyecto	13
2.2.1. Características Principales	13
2.2.2. Exclusiones del Proyecto	14
2.2.3. Limitaciones y Suposiciones	14
3. Marco Referencial	15
3.1. Marco Teórico	15
3.1.1. Aplicaciones Web	15
3.1.2. Geolocalización y Mapas Interactivos	15
3.1.3. Arquitectura Monolítica	15
3.1.4. Seguridad en Aplicaciones Web	15
3.1.5. Bases de Datos NoSQL	16
3.1.6. Tecnologías Utilizadas	16
3.1.7. Impacto del Proyecto	16
3.1.8. Marco Conceptual	16
4. Diseño e implementación	18
4.1. Formato de Datos de las APIs Consultadas	18
4.1.1. Criterios de Selección de los Datos	18
4.1.2. Tabla Comparativa de APIs Consultadas	19
4.1.3. Conclusión	19

4.2.	Diseño del Sistema Frontend	19
4.2.1.	Integración con APIs y Base de Datos	20
4.2.2.	Prototipos e Interfaces de Usuario	22
4.2.3.	Conclusiones del Diseño y Desarrollo	27
4.3.	Factores Considerados en el Diseño de las Interfaces	27
4.3.1.	Relación de las Interfaces con el Usuario Final	27
4.3.2.	Razonamiento Detrás del Esquema de Navegación	28
4.3.3.	Características de Experiencia de Usuario Basadas en Resultados Esperados	28
4.4.	Arquitectura de la Aplicación	29
4.4.1.	Introducción	29
4.4.2.	Esquema y Organización de la Base de Datos	29
4.4.3.	Patrones de Diseño e Integración	30
4.4.4.	Proceso de Funcionamiento de la Aplicación	30
4.4.5.	Estructura del Código y Función de las Carpetas	31
4.4.6.	Conclusión de la Arquitectura	32
4.5.	Implementación de Algoritmos	32
4.5.1.	Algoritmo de Haversine	32
4.5.2.	Descripción del Algoritmo	32
4.5.3.	Aplicación del Algoritmo	33
4.5.4.	Ventajas del Algoritmo	33
4.5.5.	Conclusión	33
5.	Despliegue	34
5.1.	Proceso de Despliegue de la Aplicación Web	34
5.1.1.	Planeación y Selección de Herramientas	34
5.1.2.	Configuración de la Instancia EC2	34
5.1.3.	Contenerización con Docker	35
5.1.4.	Configuración de Cloudflare	35
5.1.5.	Flujo de Despliegue	35
5.1.6.	Pruebas y Validación del Despliegue	36
5.1.7.	Dificultades y Soluciones	36
5.1.8.	Diagrama de Despliegue	36
5.1.9.	Conclusión	36
6.	Pruebas	37
6.1.	Pruebas de Usabilidad y Funcionalidad	37
6.1.1.	Introducción	37
6.1.2.	Metodología	37
6.1.3.	Pruebas de Usabilidad	38
6.1.4.	Pruebas de Funcionalidad	39
6.1.5.	Análisis de los Resultados Más Bajos	40
7.	Conclusiones	41
7.1.	Conclusiones	41

8. Trabajos futuros	43
8.1. Trabajos Futuros	43
8.1.1. Mejoramiento del Sistema de Recomendaciones	43
8.1.2. Optimización del Sistema de Búsqueda	43
8.1.3. Migración a la Nube	43
8.1.4. Mejora de la Interfaz de Usuario	44
8.1.5. Integración de Nuevas Funcionalidades	44
8.1.6. Fortalecimiento del Panel de Administradores	44
8.1.7. Actualización y Evaluación Continua	44
Referencias	45
9. Anexos	47

Lista de Figuras

4.1. Mockup hecho en Figma.	20
4.2. Zona delimitada para la región del Pacífico en el proyecto.	21
4.3. Mapa interactivo donde se muestran las especies y sidebar de los filtros.	22
4.4. Formulario para agregar datos de nuevas especies.	23
4.5. Panel de administración de solicitudes de usuarios.	24
4.6. Visualización de especie para aceptar o rechazar solicitud.	24
4.7. Página de resultados de búsqueda.	25
4.8. Página de detalles de especies.	26
4.9. Panel de administración del sistema.	26
4.10. Arquitectura monolítica[1].	30
5.1. Diagrama de despliegue que ilustra la relación entre los componentes principales.	36
6.1. Principios de usabilidad de Jacob Nielsen[32].	38
6.2. Gráfica sobre la facilidad de navegación en la aplicación	38
6.3. Gráfica sobre el tiempo de carga y rendimiento	39
6.4. Gráfica sobre la precisión y utilidad de la información de las especies	39
6.5. Gráfica sobre la mejora en la experiencia gracias a la geolocalización	40

Resumen

El presente proyecto de grado consiste en el desarrollo de una plataforma web monolítica[1] denominada GeoBioColombia, cuyo objetivo principal es facilitar la gestión y visualización de especies biológicas de la región del pacífico. Este sistema ha sido implementado utilizando MeteorJS[2] como framework principal para el backend, lo que permite una estructura integral que soporta la rápida construcción de aplicaciones en tiempo real. Para el frontend, se ha integrado ReactJS[3], conocido por su capacidad de crear interfaces de usuario dinámicas y modulares. La combinación de estas tecnologías garantiza un desarrollo ágil y una experiencia de usuario optimizada.

El proyecto se despliega en una instancia de AWS EC2[4], lo que permite una infraestructura escalable y de alta disponibilidad. Además, se utilizó Docker[5] para la contenedorización de la aplicación, facilitando su portabilidad y gestión de versiones. La seguridad de la plataforma se reforzó mediante la configuración de Cloudflare, protegiendo el sitio de ataques DDoS[6] y restringiendo el acceso a usuarios fuera de Colombia. Asimismo, se implementó un certificado HTTPS para asegurar la transmisión de datos y se adquirió el dominio geobiocolombia.online a través de GoDaddy[7] para un acceso fácil y profesional.

El sistema incluye funcionalidades avanzadas como la integración con Leaflet y React-Leaflet para la visualización de mapas interactivos que permiten a los usuarios localizar y explorar especies en diferentes regiones de Colombia. Las dependencias utilizadas, como Axios[8], Flowbite[9], y otras bibliotecas de soporte, garantizan la interactividad y la presentación de la información de manera eficiente.

El proyecto no solo busca ser una herramienta de consulta para investigadores y entusiastas de la biodiversidad, sino también un espacio colaborativo donde los usuarios puedan agregar información relevante sobre las especies.

El proceso de desarrollo se llevó a cabo en varias etapas, desde la planificación y la investigación inicial descritas en el anteproyecto hasta la implementación y despliegue final. La experiencia adquirida a lo largo de este proyecto representa un hito importante en mi formación académica y profesional, reflejando tanto los desafíos como las soluciones encontradas durante su desarrollo.

Las pruebas realizadas a 41 estudiantes de la Universidad del Valle evidenciaron una percepción positiva de GeoBioColombia. Más del 80 % calificaron como “Muy fácil” o “Fácil” el registro y la navegación, destacando la intuitividad de la interfaz. La información sobre las especies fue valorada como “Muy precisa y útil”, mientras que la función de geolocalización mejoró significativamente la experiencia del usuario. Estos resultados confirman que la plataforma cumple sus objetivos educativos y funcionales, con oportunidades de mejora en la carga en dispositivos antiguos y guías más interactivas.

Keywords

Biodiversidad, Geolocalización, Conservación, WebApp, Ciencia Ciudadana, Educación Ambiental, Fauna del Pacífico, Mapa Interactivo, Sostenibilidad.

Capítulo 1

Introducción

1.1. Formulación del Problema

¿Cómo puede una WebApp de geolocalización y catálogo, específicamente diseñada para la región del Pacífico colombiano, contribuir eficazmente a la difusión y comprensión de su biodiversidad entre turistas, investigadores y el sector educativo?

1.2. Descripción del Problema

Colombia enfrenta un marcado desafío por la falta de conocimiento y comprensión sobre su reconocida biodiversidad y la importancia de su riqueza natural. Este desafío afecta tanto a turistas y residentes como a profesionales[10]. La insuficiente conciencia pública sobre la biodiversidad colombiana y sus implicaciones ecológicas limita la experiencia del turismo consciente y puede llevar a interacciones dañinas con el medio ambiente. Tales interacciones, basadas en información errónea, pueden causar la degradación de hábitats y la perturbación de especies protegidas o endémicas, reduciendo la biodiversidad[11].

Las herramientas digitales actuales, diseñadas para difundir información sobre la biodiversidad, a menudo generalizan. Proporcionan datos que no se ajustan a la ubicación específica del usuario y omiten variaciones estacionales o migratorias de las especies. Esto aumenta el riesgo de diseminar información desactualizada o incorrecta[12].

La comunidad científica también enfrenta desafíos. Biólogos e investigadores carecen de una plataforma digital centralizada para consultar y corregir datos sobre especies en áreas específicas de Colombia. Esta falta obstaculiza la investigación y fomenta la dispersión de datos, propagando información errónea[13].

Además, el ámbito educativo se ve afectado por la ausencia de herramientas digitales enfocadas en la biodiversidad colombiana. Esto limita los métodos pedagógicos y reduce las oportunidades de aprendizaje[14].

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar una WebApp dedicada a la categorización y visualización de la diversa fauna de la región del Pacífico de Colombia.

1.3.2. Objetivos Específicos y Resultados Esperados

A continuación, se presenta la relación entre los objetivos específicos y los resultados esperados:

Objetivos Específicos	Resultados esperados
1) Recopilar datos científicamente validados sobre las especies de fauna de la región del Pacífico, garantizando así la precisión y actualidad de la información.	Investigar y seleccionar bases de datos externas y APIs abiertas relevantes, para la integración de datos sobre biodiversidad de la fauna de la región del Pacífico.
2) Diseñar la WebApp de acuerdo a los datos recopilados sobre especies de fauna de la región del Pacífico.	Diseñar la interfaz de usuario y experiencia de usuario (UI/UX) y desarrollar prototipos.
3) Implementar una WebApp que integre geolocalización y el catálogo de fauna de la región del Pacífico.	Programar la plataforma, integrando la base de datos con la interfaz, ejecutar pruebas de usabilidad y funcionalidad, y realizar correcciones técnicas.
4) Evaluar el impacto de la WebApp en los usuarios finales mediante encuestas y pruebas de usabilidad.	Se llevarán a cabo encuestas y pruebas de usabilidad para recopilar feedback de usuarios, con el objetivo de medir y mejorar el impacto de la WebApp en la comprensión de la biodiversidad de la fauna de la región del Pacífico, optimizando así su experiencia y funcionalidad.

Cuadro 1.1: Objetivos específicos vs resultados esperados. Fuente: Elaboración propia.

1.4. Metodología

Para la elaboración del proyecto se adoptaron principios ágiles, lo que permitió ajustar los requisitos y objetivos a medida que avanzaba el desarrollo. Se aplicaron principios de adaptabilidad, flexibilidad y retroalimentación constante. La elección de una arquitectura integrada permitió una implementación más sencilla y unificada, facilitando el desarrollo y el despliegue del sistema. Se consultaron diversas fuentes de información, incluyendo artículos científicos, libros y tutoriales, para guiar y apoyar el desarrollo. Como resultado del análisis de la información recopilada, se implementó un sistema de geolocalización y catálogo de fauna, aprovechando la integración de Leaflet y React-Leaflet[15] para la presentación de mapas interactivos. El proyecto se diseñó, implementó y desplegó de forma autónoma. Se mantuvo retroalimentación con el director del proyecto semanalmente y se presentó el progreso cada dos semanas.

1.4.1. Recolección de Requisitos (2 semanas)

- Definir requisitos funcionales y no funcionales.

- Identificar componentes con casos de uso e historias de usuario para trazar la funcionalidad.

1.4.2. Diseño del Sistema (4 semanas)

- Diseñar wireframes para la interfaz de usuario.
- Diseñar la arquitectura que permita la integración de funcionalidades de geolocalización, búsquedas y gestión de datos.

1.4.3. Desarrollo e Implementación (12 semanas)

- Desarrollar el frontend de manera concurrente con el backend, utilizando React.js[3] y MeteorJS[2].
- Implementar la base de datos con MongoDB para gestionar información sobre fauna y usuarios.

1.4.4. Despliegue (1 semana)

- Desplegar la aplicación web en un entorno de producción en una instancia de AWS EC2, utilizando Docker para la contenedorización y Cloudflare para la seguridad y configuración HTTPS.

1.4.5. Pruebas y Validación (2 semanas)

- Realizar pruebas de funcionalidad y rendimiento.
- Evaluar los resultados de las pruebas para asegurar el correcto funcionamiento de la aplicación.

1.5. Información sobre los capítulos

En la siguiente tabla se muestra una corta descripción de cada capítulo del documento y se explica qué objetivos específicos aborda.

Capítulo	Descripción	Objetivo específico
Capítulo 2: Justificación y alcance del proyecto	Expone las razones del proyecto, estableciendo las funcionalidades principales de la aplicación web y las restricciones del sistema.	Ninguno
Capítulo 3: Marco referencial	Se establecen y definen los términos y conceptos clave relacionados con la geolocalización, la biodiversidad del Pacífico colombiano y las tecnologías utilizadas en el proyecto.	Ninguno
Capítulo 4: Diseño e implementación	Presenta la construcción del sistema, incluyendo diagramas de arquitectura, el esquema de la base de datos MongoDB y el código implementado en MeteorJS y ReactJS.	1, 2, 3
Capítulo 5: Despliegue	Documenta el proceso de despliegue de la aplicación en AWS EC2, incluyendo la configuración de Docker, Cloudflare y el dominio geobiocolombia.online.	3
Capítulo 6: Pruebas	En este capítulo se documentan las pruebas funcionales, de rendimiento y de seguridad realizadas para garantizar el correcto funcionamiento del sistema.	4
Capítulo 7: Conclusiones	Se presentan las conclusiones derivadas del desarrollo del proyecto, evaluando los resultados obtenidos.	Ninguno
Capítulo 8: Trabajos futuros	Se proponen mejoras y expansiones para la aplicación, como el desarrollo de una versión móvil y nuevas funcionalidades para usuarios y administradores.	Ninguno
Capítulo 9: Bibliografía	Se presentan las referencias utilizadas para el desarrollo y documentación del proyecto.	Ninguno

Cuadro 1.2: Estructura de capítulos

Capítulo 2

Justificación y alcance del proyecto

2.1. Justificación

El desarrollo de la aplicación web *GeoBio Colombia* surge de la necesidad de centralizar y facilitar el acceso a información relevante y detallada sobre la fauna de la región del Pacífico colombiano. La región es conocida por su rica biodiversidad, pero esta información suele estar dispersa y de difícil acceso, lo que limita las oportunidades de educación y preservación. Esta plataforma busca solventar estas deficiencias ofreciendo un espacio intuitivo para la visualización y consulta de especies geolocalizadas, promoviendo así el conocimiento y la concienciación ambiental.

La aplicación está diseñada con un enfoque monolítico para simplificar la integración y despliegue de sus componentes, maximizando la estabilidad y el rendimiento en un entorno de servidor. La seguridad y accesibilidad están garantizadas mediante el uso de Cloudflare[16] para protección contra ataques DDoS[6] y la configuración de HTTPS, así como un firewall configurado para limitar el acceso no autorizado. Este proyecto contribuirá a la divulgación de la riqueza biológica local, beneficiando a investigadores, estudiantes y público en general.

2.2. Alcance del Proyecto

Este proyecto consiste en el desarrollo de una aplicación web responsiva orientada al usuario. La aplicación permitirá a los usuarios explorar un catálogo interactivo de fauna con información detallada de cada especie, incluyendo su geolocalización. Además, los administradores tendrán la capacidad de gestionar los datos de las especies mediante un sistema CRUD (crear, leer, actualizar y eliminar).

El proyecto utiliza MeteorJS[2] y ReactJS[3], lo que simplifica su desarrollo y mantenimiento. La base de datos MongoDB se emplea para almacenar información de las especies y usuarios. El despliegue se realiza en una instancia de AWS EC2[4], utilizando Docker[5] para la contenedorización y asegurando la protección mediante Cloudflare[16]. Además, se proporcionará documentación técnica y un manual de usuario para facilitar el uso y la comprensión del sistema.

2.2.1. Características Principales

- **Interfaz de Usuario Responsiva:**
 - Página inicial accesible tanto para usuarios públicos como para administradores.
 - Visualización interactiva de mapas con geolocalización de especies.

- Catálogo detallado de especies con información relevante.
- Funcionalidades de búsqueda para filtrar especies según criterios específicos.
- Páginas de gestión para administradores con operaciones CRUD.
- **Base de Datos:** MongoDB, utilizada para gestionar la información de fauna y usuarios.
- **Seguridad y Protecciones:** Uso de firewall, protección DDoS y HTTPS mediante Cloudflare.
- **Documentación del Proyecto:**
 - Documentación técnica de los componentes.
 - Manual de usuario.

2.2.2. Exclusiones del Proyecto

- Integración de pagos en línea.
- Aplicaciones móviles nativas.

2.2.3. Limitaciones y Suposiciones

- **Limitaciones:**
 - El sistema se diseñará exclusivamente para navegadores web.
 - Las operaciones de gestión se limitarán a CRUD.
- **Suposiciones:**
 - Los usuarios contarán con acceso a internet y navegadores compatibles.
 - El personal administrativo recibirá capacitación adecuada.
 - La infraestructura de AWS EC2 será suficiente para soportar la aplicación.

Capítulo 3

Marco Referencial

3.1. Marco Teórico

3.1.1. Aplicaciones Web

Una aplicación web es un software que se ejecuta en el navegador, proporcionando acceso a funcionalidades avanzadas sin necesidad de instalación. En el contexto de *GeoBio Colombia*, la aplicación permite a los usuarios interactuar con un catálogo de fauna, utilizando mapas interactivos para la geolocalización y funcionalidades CRUD para la gestión de datos. Las aplicaciones web son ideales para brindar acceso remoto a la información, garantizando la accesibilidad y escalabilidad del sistema.

3.1.2. Geolocalización y Mapas Interactivos

La geolocalización permite identificar la ubicación geográfica de especies en el mapa. Herramientas como Leaflet y React-Leaflet facilitan la creación de mapas interactivos en aplicaciones web, permitiendo a los usuarios explorar la distribución de especies de manera visual e intuitiva. Estas tecnologías fueron seleccionadas por su simplicidad y eficiencia en la integración con ReactJS.

3.1.3. Arquitectura Monolítica

El sistema está desarrollado como una solución unificada que centraliza todas las funcionalidades en un solo entorno. Esta arquitectura simplifica el desarrollo, despliegue y mantenimiento inicial del sistema. Aunque no es tan flexible como la arquitectura de microservicios, es adecuada para proyectos pequeños y medianos, asegurando un rendimiento óptimo en un entorno controlado.

3.1.4. Seguridad en Aplicaciones Web

La seguridad es un aspecto crítico en el desarrollo de aplicaciones web. Para *GeoBio Colombia*, se implementaron medidas como:

- Configuración de HTTPS mediante Cloudflare para garantizar la comunicación segura entre el cliente y el servidor.
- Protección contra ataques DDoS mediante los servicios de Cloudflare.
- Configuración de un firewall para restringir el acceso no autorizado.

Estas medidas aseguran la integridad y confidencialidad de los datos.

3.1.5. Bases de Datos NoSQL

MongoDB es una base de datos NoSQL utilizada en el proyecto para almacenar información sobre especies y usuarios. Su flexibilidad y escalabilidad permiten gestionar datos no estructurados de manera eficiente, adaptándose a los requerimientos del sistema.

3.1.6. Tecnologías Utilizadas

El proyecto combina varias tecnologías modernas para garantizar una experiencia de usuario eficiente y una implementación robusta:

- **MeteorJS:** Framework para el desarrollo del backend que facilita la creación de aplicaciones en tiempo real.
- **ReactJS:** Biblioteca para la construcción de interfaces de usuario interactivas.
- **Leaflet y React-Leaflet:** Herramientas para la integración de mapas interactivos.
- **Docker:** Tecnología de contenedorización utilizada para simplificar el despliegue.
- **AWS EC2:** Plataforma de computación en la nube para alojar la aplicación.

3.1.7. Impacto del Proyecto

El proyecto *GeoBio Colombia* contribuye al conocimiento y preservación de la biodiversidad del Pacífico colombiano, proporcionando una herramienta educativa y accesible. Además, promueve la concienciación ambiental y la investigación científica al centralizar información clave sobre especies geolocalizadas.

3.1.8. Marco Conceptual

Geolocalización: Proceso tecnológico que determina la ubicación geográfica de un dispositivo o individuo en la Tierra. Utiliza señales GPS, triangulación de redes móviles y otras tecnologías para establecer coordenadas exactas.[17]

Sistema de Información Geográfica (SIG): Un sistema diseñado para capturar, almacenar, manipular, analizar, gestionar y presentar datos espaciales o geográficos. Combina cartografía, bases de datos y análisis estadístico para estudiar y gestionar información georreferenciada.[18]

Biodiversidad: Diversidad de vida en todas sus formas y niveles, incluyendo la diversidad dentro de las especies, entre las especies y de los ecosistemas. Refleja la variabilidad entre los organismos vivos y sus complejas interacciones en el entorno natural.[19]

Ciencia Ciudadana: Práctica científica colaborativa en la que individuos o grupos de no científicos participan voluntariamente en el proceso científico, contribuyendo en actividades como la recopilación de datos, análisis y divulgación de resultados.[20]

Conservación de la Biodiversidad: Conjunto de prácticas y estrategias destinadas a proteger y preservar la diversidad biológica frente a amenazas como la destrucción del hábitat, el cambio climático y la invasión de especies exóticas.[21]

Flora y Fauna: Términos que se refieren colectivamente a las plantas (flora) y a los animales (fauna) de una región o período específico.[22]

Interfaz de Usuario (UI) y Experiencia del Usuario (UX): Definición UI: Diseño de interfaces en software o dispositivos digitalizados, enfocado en la estética y en la disposición de los elementos para facilitar la interacción del usuario.

Definición UX: Se refiere a la experiencia general del usuario al interactuar con un producto o servicio, enfocado en la optimización de la satisfacción y la usabilidad.[23]

Tecnologías Web y Móviles: Conjunto de tecnologías y aplicaciones diseñadas para el uso en Internet y dispositivos móviles, incluyendo aplicaciones web, plataformas en línea y aplicaciones móviles.[24]

Capítulo 4

Diseño e implementación

4.1. Formato de Datos de las APIs Consultadas

La API de GBIF[25] utiliza el formato JSON para estructurar y entregar datos relacionados con especies biológicas. Este formato permite manejar datos de manera sencilla y eficiente, facilitando su integración con la base de datos MongoDB y su visualización en la interfaz web. Por ejemplo, un registro típico incluye información detallada como el nombre científico, coordenadas geográficas, y datos multimedia.

Por otro lado, la API de IUCN[26] utiliza también el formato JSON, pero los datos ofrecidos son más limitados, ya que no incluyen información geográfica específica, lo que resulta fundamental para el objetivo del proyecto, que es la visualización de las especies en un mapa interactivo.

- **Nombre científico:** Ejemplo: *Pterinoxylus spinulosus* Redtenbacher, 1908.
- **Identificadores únicos:** Claves como gbifID y taxonID, que permiten rastrear y manejar datos con precisión.
- **Clase, familia y género:** Clasificación taxonómica relevante.
- **Datos geográficos:** Latitud, longitud, provincia y localidad, indispensables para la funcionalidad del mapa interactivo.
- **Multimedia:** Imágenes con licencia abierta (CC-BY) para mejorar la experiencia del usuario.
- **Categoría de la lista roja de la UICN:** Indica el estado de conservación de las especies.

Por otro lado, la API de IUCN ofrece datos más simples, con atributos básicos como el nombre científico y la categoría de conservación. Sin embargo, no proporciona información geográfica ni multimedia, lo que limita su utilidad para los objetivos del proyecto.

4.1.1. Criterios de Selección de los Datos

Los criterios específicos para seleccionar los datos incluidos en la aplicación final fueron:

- **Datos Geográficos:** Se priorizaron registros que incluyen latitud y longitud, ya que son esenciales para la representación en el mapa interactivo.
- **Multimedia:** Especies con imágenes disponibles mejoran significativamente la experiencia del usuario.

- **Estado de Conservación:** La categoría de la lista roja de la UICN es utilizada para informar sobre el estado de las especies.
- **Clasificación Taxonómica Completa:** Los datos taxonómicos detallados facilitan la navegación y el filtrado en la aplicación.

4.1.2. Tabla Comparativa de APIs Consultadas

Característica	API de GBIF	API de IUCN	Selección para la Aplicación
Formato de Datos	JSON	JSON	GBIF
Datos Geográficos	Latitud, longitud, localidad	No disponibles	GBIF
Multimedia	Imágenes con licencia abierta	No disponibles	GBIF
Estado de Conservación	Categoría de la lista roja	Categoría de la lista roja	Ambas
Clasificación Taxonómica	Nombre científico, familia, género, clase	Nombre científico	GBIF
Complejidad de los Datos	Información detallada y rica en atributos	Información limitada	GBIF
Relevancia para la Aplicación	Alta: ideal para mapas y datos detallados	Baja: insuficiente para el objetivo del mapa	GBIF

Cuadro 4.1: Objetivos específicos vs resultados esperados. Fuente: Elaboración propia.

4.1.3. Conclusión

La API de GBIF fue seleccionada como la principal fuente de datos debido a su riqueza de información, que incluye atributos geográficos, taxonómicos y multimedia necesarios para cumplir con los objetivos del proyecto. Aunque la API de IUCN también proporciona datos útiles, su falta de soporte para ubicaciones geográficas específicas limita su aplicabilidad para la funcionalidad del mapa interactivo.

4.2. Diseño del Sistema Frontend

El desarrollo del frontend de *GeoBio Colombia* comenzó con la creación de mockups hechos en Figma[27] que delinearon el diseño y la funcionalidad de la interfaz de usuario. Estos mockups sirvieron como una guía visual preliminar, facilitando la planificación de los elementos y la disposición general de las páginas.

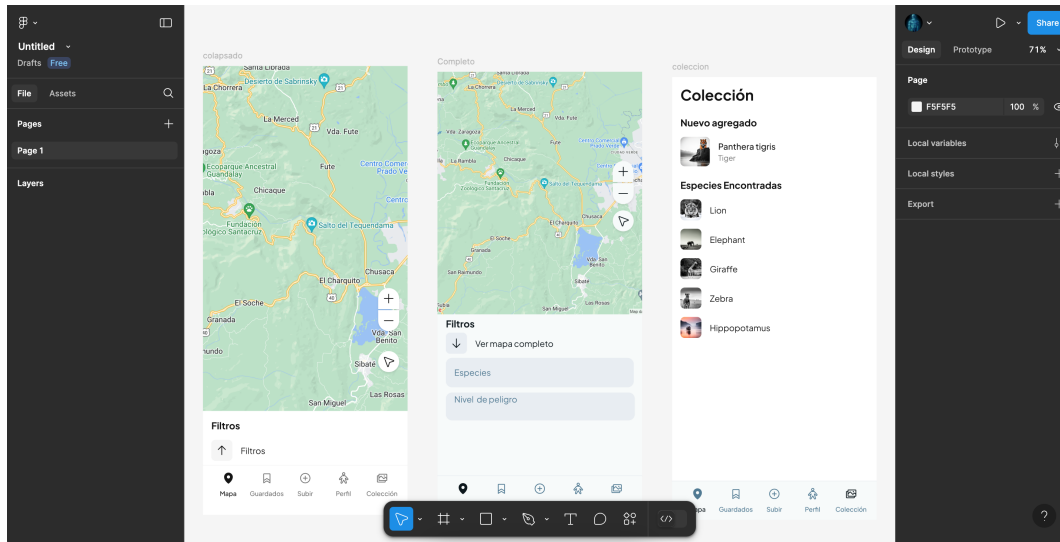


Figura 4.1: Mockup hecho en Figma.

La interfaz fue desarrollada utilizando ReactJS[3], una librería reconocida por su flexibilidad y eficiencia para construir aplicaciones web dinámicas. Se empleó JSX para la estructura básica de los componentes, mientras que TailwindCSS[28] permitió estilizar los elementos de forma coherente, garantizando una apariencia moderna y responsiva.

4.2.1. Integración con APIs y Base de Datos

Para obtener la información de las especies de la región del Pacífico, se integró la API de GBIF[25], una fuente confiable y consistente de datos sobre biodiversidad. Las coordenadas utilizadas para delimitar la región se obtuvieron mediante la herramienta GeoJSON[29] y abarcan los departamentos de Chocó, Valle del Cauca, CaEnuca y Nariño.

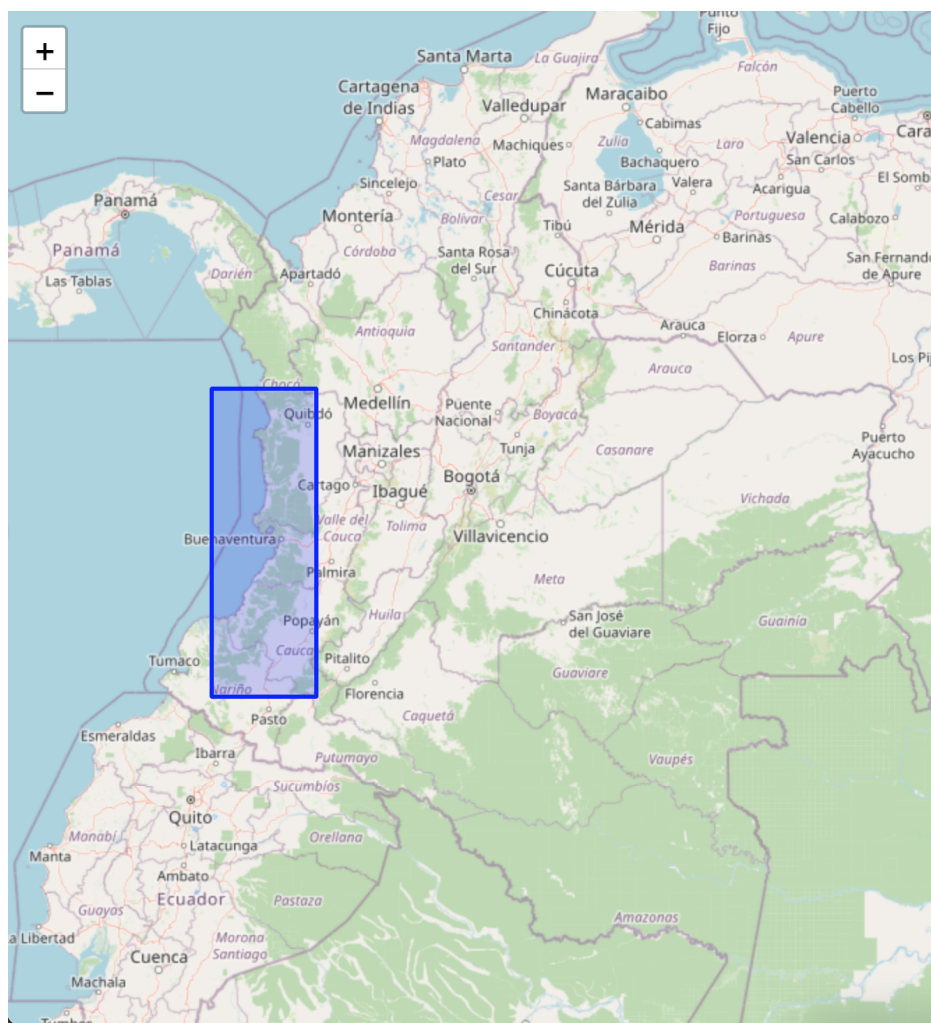


Figura 4.2: Zona delimitada para la región del Pacífico en el proyecto.

La API permite filtrar por ubicación geográfica y categoría taxonómica, obteniendo datos importantes como el nombre científico, coordenadas, categoría en la lista roja de la UICN[26], y enlaces a recursos multimedia. La información recopilada se almacena en una base de datos MongoDB para facilitar su gestión y consulta.

Para garantizar la integridad de los datos, se establecieron reglas para manejar posibles errores de la API:

- Si una especie no cuenta con una imagen, se utiliza por defecto el logotipo del proyecto.
- Si la especie no está evaluada en la lista roja de la UICN, se le asigna la categoría "no evaluada" ("NE").
- Las especies que no cumplen con los requisitos mínimos de información son descartadas.

En el siguiente endpoint de la aplicación, (`/test`), los usuarios pueden visualizar la zona delimitada en el mapa para el proyecto.

4.2.2. Prototipos e Interfaces de Usuario

Interfaz de Inicio

Mapa Interactivo y Búsqueda de Especies:

- Visualización de un mapa interactivo que permite explorar la distribución geográfica de las especies.
- Barra de búsqueda para filtrar especies por nombre, categoría o ubicación.

Listado de Especies Destacadas:

- Muestra una lista de especies destacadas con imágenes representativas.
- Permite acceder a detalles adicionales haciendo clic en cada especie.

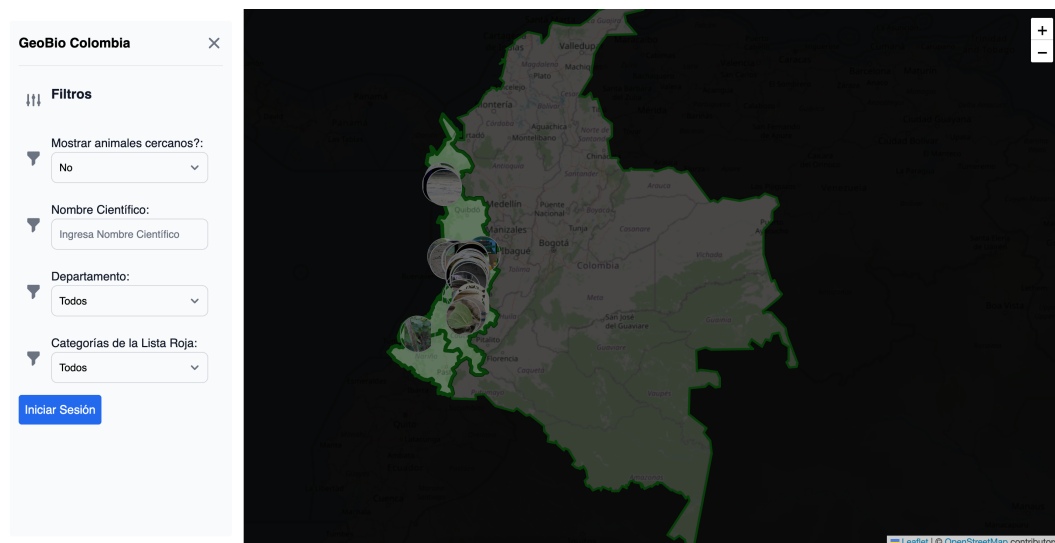
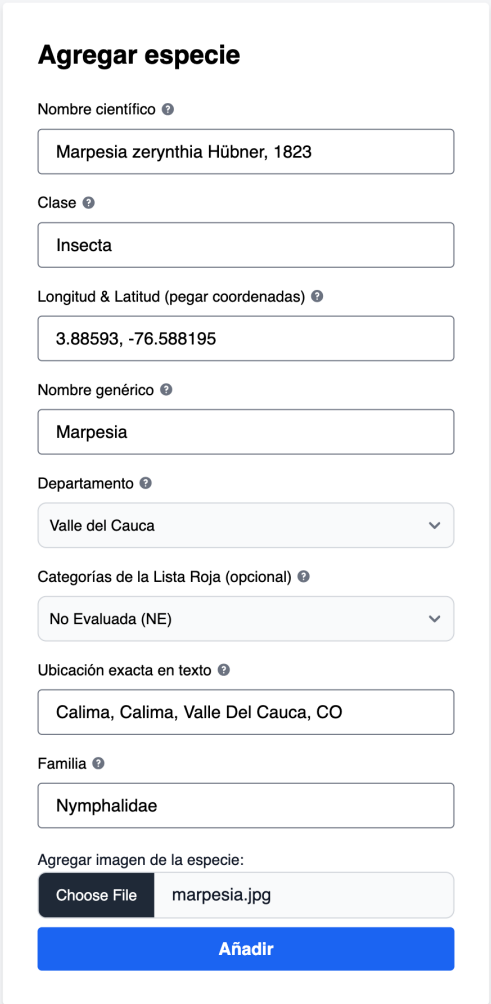


Figura 4.3: Mapa interactivo donde se muestran las especies y sidebar de los filtros.

Interfaz de Formulario

Formulario de Contribución:

- Permite a los usuarios añadir datos sobre nuevas especies.
- Campos obligatorios como nombre científico, coordenadas geográficas y categoría.
- Validación de datos para evitar información incompleta o errónea.



Agregar especie

Nombre científico ⓘ

Marpesia zerynthia Hübner, 1823

Clase ⓘ

Insecta

Longitud & Latitud (pegar coordenadas) ⓘ

3.88593, -76.588195

Nombre genérico ⓘ

Marpesia

Departamento ⓘ

Valle del Cauca

Categorías de la Lista Roja (opcional) ⓘ

No Evaluada (NE)

Ubicación exacta en texto ⓘ

Calima, Calima, Valle Del Cauca, CO

Familia ⓘ

Nymphalidae

Agregar imagen de la especie:

Choose File marpesia.jpg

Añadir

Mi Cuenta Publicar Dashboard

Figura 4.4: Formulario para agregar datos de nuevas especies.

Interfaz de Solicitudes de Usuarios

Panel de Solicitudes:

- Los administradores pueden revisar solicitudes de contribuciones realizadas por los usuarios.
- Funciones para aprobar, rechazar o solicitar más información.

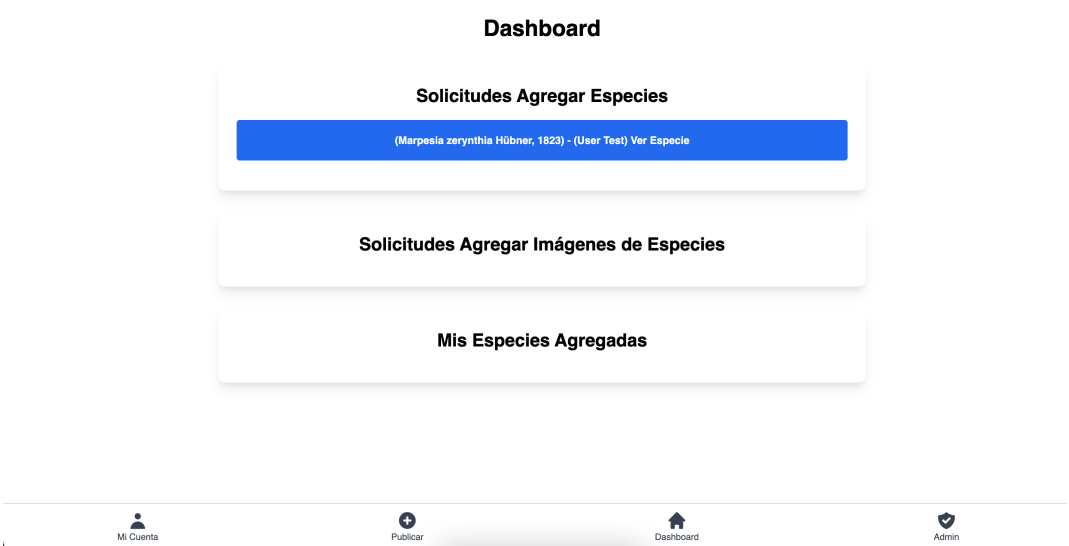


Figura 4.5: Panel de administración de solicitudes de usuarios.

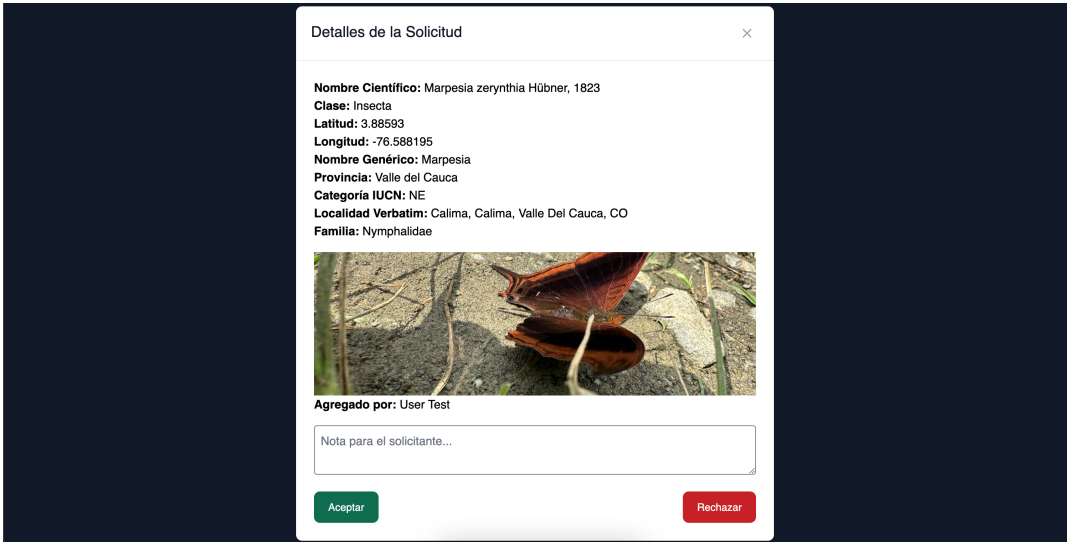


Figura 4.6: Visualización de especie para aceptar o rechazar solicitud.

Interfaz de Resultados de Búsqueda

Visualización de Resultados:

- Muestra una lista de especies que coinciden con los criterios de búsqueda.
- Incluye imágenes y nombres.

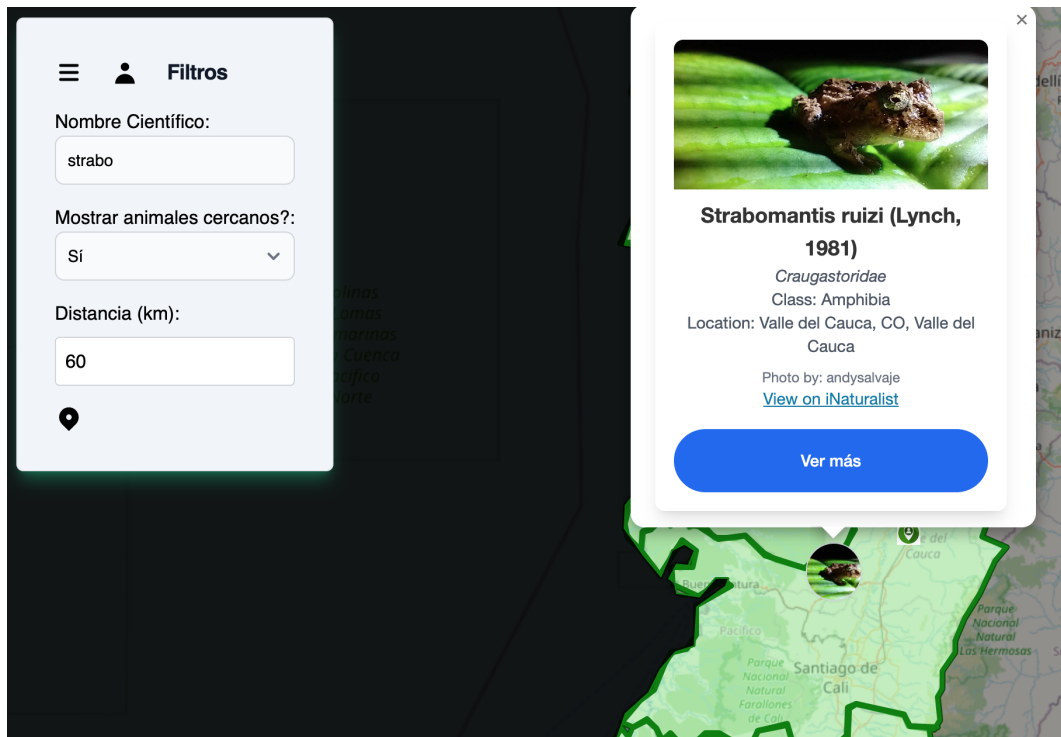


Figura 4.7: Página de resultados de búsqueda.

Interfaz de Detalles de Especies

Muestra Información Detallada de una Especie Seleccionada:

- Imagen representativa de la especie.
- Información clave como nombre científico, ubicación y características destacadas.
- Enlace para descargar datos adicionales de la especie.

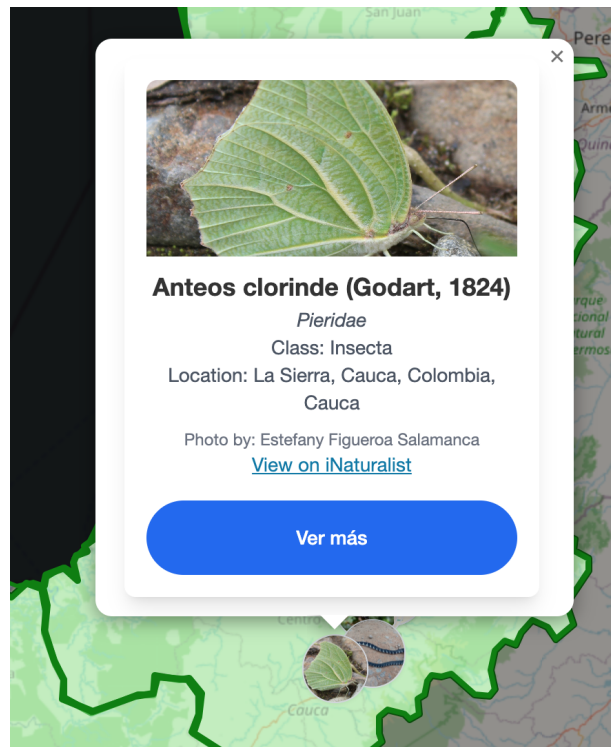


Figura 4.8: Página de detalles de especies.

Interfaz de Administración

Gestión de Especies:

- Permite al administrador añadir, editar o eliminar registros de especies en la base de datos.
- Campos editables: nombre, categoría, descripción y coordenadas geográficas.

Gestión de Usuarios:

- Muestra una lista de usuarios registrados.
- Opciones para editar permisos o eliminar cuentas.

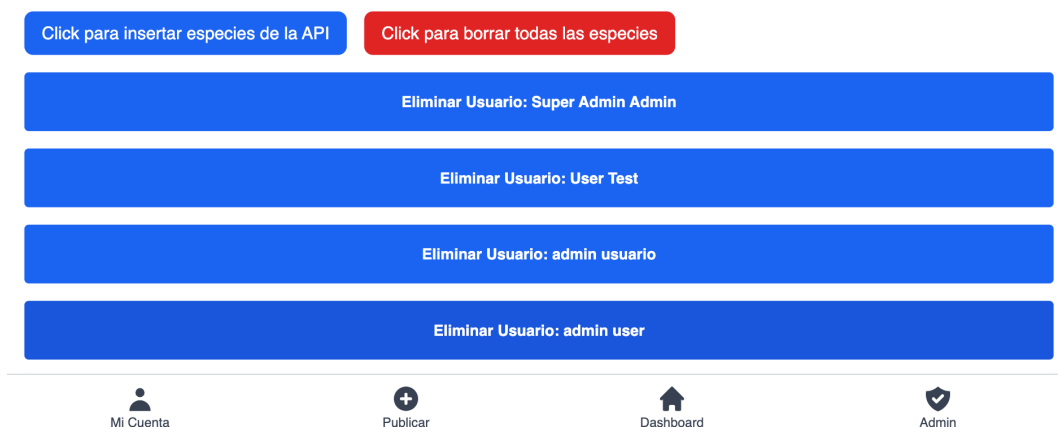


Figura 4.9: Panel de administración del sistema.

4.2.3. Conclusiones del Diseño y Desarrollo

El diseño y desarrollo de *GeoBio Colombia* se basaron en principios centrados en el usuario, integrando tecnologías modernas como ReactJS, TailwindCSS y MongoDB para garantizar una experiencia eficiente y visualmente atractiva.

La integración con la API de GBIF[25] permitió acceder a datos relevantes, esenciales para implementar funcionalidades como el mapa interactivo, los filtros, y la gestión eficiente de datos. Además, se implementaron formularios y paneles de administración que garantizan un manejo robusto de las contribuciones y solicitudes de los usuarios.

4.3. Factores Considerados en el Diseño de las Interfaces

El diseño de las interfaces de *GeoBio Colombia* se realizó teniendo en cuenta los siguientes factores clave:

1. Audiencia Objetivo:

- La aplicación está dirigida a estudiantes, investigadores y entusiastas de la biodiversidad. Por lo tanto, las interfaces fueron diseñadas para ser intuitivas y visualmente atractivas, adaptándose a usuarios con distintos niveles de experiencia tecnológica.

2. Accesibilidad y Responsividad:

- Se utilizó TailwindCSS para asegurar que la aplicación fuera completamente responsiva, permitiendo su correcto funcionamiento en dispositivos de distintos tamaños, desde teléfonos móviles hasta pantallas grandes.

3. Elementos Interactivos:

- Los botones, barras de búsqueda y mapas interactivos fueron diseñados para proporcionar retroalimentación inmediata, mejorando la interacción y reduciendo la curva de aprendizaje.
- Se incorporaron elementos como filtros y visualizaciones dinámicas para ofrecer una experiencia más personalizada.

4. Minimización de Pasos:

- La navegación entre diferentes secciones, como la búsqueda de especies o la consulta de información detallada, se optimizó para minimizar los clics necesarios, asegurando una experiencia fluida.

5. Conexión con el Contexto de Biodiversidad:

- Las interfaces destacan elementos visuales relacionados con la biodiversidad, como imágenes de especies y mapas geográficos, para conectar al usuario con el objetivo principal del proyecto.

4.3.1. Relación de las Interfaces con el Usuario Final

1. Interfaz de Inicio:

- Diseñada para proporcionar una visión general de las especies destacadas y permitir una búsqueda fácil mediante filtros como nombre, categoría o ubicación.
- Ideal para usuarios ocasionales o aquellos que buscan una experiencia de exploración.

2. Interfaz de Resultados de Búsqueda:

- Pensada para usuarios con un interés específico, esta interfaz permite visualizar los resultados de búsqueda de forma clara, con imágenes y nombres destacados.

3. Interfaz de Administración:

- Orientada a administradores del sistema, incluye funciones avanzadas como la gestión de especies y usuarios, con opciones para aprobar o rechazar contribuciones.

4. Interfaz de Contribución:

- Permite a los usuarios añadir datos sobre nuevas especies, promoviendo la colaboración y el enriquecimiento de la base de datos.

5. Interfaz de Detalles de Especies:

- Proporciona información completa de una especie seleccionada, conectando a los usuarios con datos geográficos, taxonómicos y multimedia.

4.3.2. Razonamiento Detrás del Esquema de Navegación

El esquema de navegación fue diseñado para ser intuitivo y directo, con los siguientes principios:

- **Agrupación Lógica:** Las funciones principales (búsqueda, contribuciones, administración) están claramente separadas en pestañas o secciones visibles desde el menú principal.
- **Accesibilidad Rápida:** Se priorizó que todas las acciones principales estén accesibles con un máximo de dos clics desde la pantalla de inicio.
- **Retroalimentación Visual:** Elementos interactivos como botones y filtros incluyen animaciones sutiles para indicar que una acción ha sido ejecutada exitosamente.

4.3.3. Características de Experiencia de Usuario Basadas en Resultados Esperados

1. Facilidad de Uso:

- Las interfaces se diseñaron para guiar al usuario de manera natural, reduciendo la necesidad de tutoriales extensos.
- Ejemplo: La barra de búsqueda incluye sugerencias automáticas para facilitar la interacción.

2. Velocidad y Eficiencia:

- Las solicitudes al servidor son rápidas, gracias a la integración con MongoDB y MeteorJS, reduciendo tiempos de espera y mejorando la experiencia general.

3. Engagement Visual:

- Las imágenes y mapas interactivos atraen al usuario y lo mantienen interesado, alineándose con el objetivo de educar sobre la biodiversidad.

4. Colaboración Activa:

- La interfaz de contribución fomenta la participación activa de los usuarios, fortaleciendo el propósito colaborativo de la plataforma.

4.4. Arquitectura de la Aplicación

4.4.1. Introducción

La arquitectura de *GeoBio Colombia* se diseñó bajo un enfoque integrado, en el que todos los componentes de la aplicación están cohesionados en un único sistema. Este diseño prioriza la simplicidad, la eficiencia y el tiempo de desarrollo, asegurando una experiencia consistente para los usuarios finales.

- **Simplicidad:** La integración en un solo sistema reduce la complejidad de gestión y despliegue.
- **Flexibilidad Inicial:** Este enfoque permite incorporar nuevas funcionalidades de manera ágil, ideal para proyectos en etapas tempranas de desarrollo.
- **Integración Directa:** Todos los componentes, desde la interfaz de usuario hasta la lógica del servidor y la base de datos, están conectados de manera directa.
- **Optimización de Recursos:** Al no requerir múltiples instancias ni servicios independientes, se logra un uso eficiente de los recursos en la instancia de AWS EC2.

4.4.2. Esquema y Organización de la Base de Datos

El sistema utiliza MongoDB como base de datos principal, con tres colecciones clave que reflejan la naturaleza semiestructurada y flexible de los datos manejados:

- **Usuarios (*Users*):** Contiene información de perfiles, roles y credenciales.
- **Especies (*Species*):** Almacena datos taxonómicos, geográficos y multimedia de las especies.
- **Configuraciones (*Settings*):** Incluye configuraciones globales del sistema, como la definición de roles.

Las características principales de MongoDB, como su capacidad para manejar datos jerárquicos y flexibles, lo convierten en la elección natural para MeteorJS, donde su compatibilidad nativa con MiniMongo facilita la integración con el frontend.

Esquema de la Colección *Species*

La colección *Species* se diseñó para optimizar el manejo de datos geolocalizados y taxonómicos. Los datos clave incluyen:

- **Datos geográficos:** Coordenadas de latitud y longitud para visualización en mapas.
- **Estado:** Indicador de la validación de datos (*accepted*, *pending*, etc.).
- **Multimedia:** Enlaces a imágenes asociadas a cada especie.

Esquema de la Colección *Users*

Esta colección almacena la información de los usuarios, incluyendo:

- **Roles:** Relacionados con la colección *Settings* para asignar permisos específicos.
- **Credenciales:** Almacenadas de manera segura utilizando bcrypt.
- **Tokens:** Para gestionar sesiones activas de manera eficiente.

4.4.3. Patrones de Diseño e Integración

El sistema sigue el patrón de diseño *Modelo-Vista-Controlador* (MVC), donde:

- El **modelo** está representado por las colecciones de MongoDB y su interacción a través de publicaciones y suscripciones.
- La **vista** se implementa con ReactJS para manejar la interacción con el usuario.
- El **controlador** reside en MeteorJS, que facilita la comunicación entre la vista y el modelo.

MeteorJS proporciona una integración nativa entre el backend y el frontend, eliminando la necesidad de capas intermedias. Esto simplifica la gestión de datos en tiempo real mediante publicaciones en el servidor y suscripciones en el cliente.

4.4.4. Proceso de Funcionamiento de la Aplicación

El flujo operativo del sistema abarca las siguientes etapas:

1. **Acceso al Sistema:** Los usuarios acceden mediante el dominio *geobiocolombia.online*.
2. **Publicaciones y Suscripciones:** El servidor publica datos relevantes según la vista solicitada; el cliente se suscribe y consume solo lo necesario.
3. **Interacción con la Base de Datos:** Las operaciones de inserción, actualización y eliminación se realizan a través de métodos de MeteorJS.
4. **Visualización:** Los datos geolocalizados se renderizan mediante Leaflet y React-Leaflet.

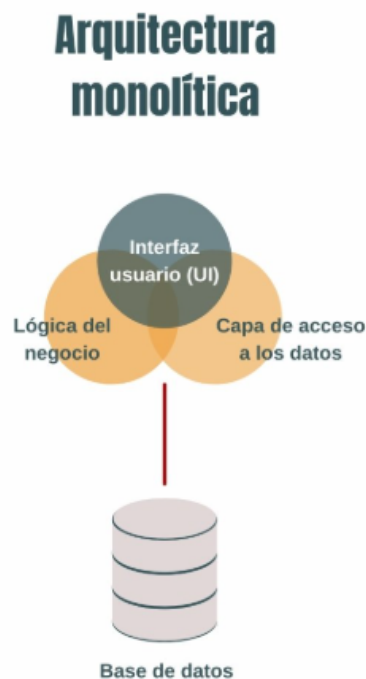


Figura 4.10: Arquitectura monolítica[1].

4.4.5. Estructura del Código y Función de las Carpetas

La estructura del código de *GeoBio Colombia* está diseñada para organizar de manera eficiente los diferentes componentes de la aplicación, facilitando su desarrollo, mantenimiento y escalabilidad futura. Aunque el proyecto sigue un enfoque cohesivo, la convención de organizar la lógica en la carpeta **imports** es una práctica común en MeteorJS. Esto permite mantener el código modular y fácilmente extensible.

Función de Cada Carpeta

- **client:** Contiene los archivos responsables de la inicialización del cliente, como hojas de estilo, scripts principales y configuraciones iniciales. Se relaciona con la **Vista** (*Frontend*), utilizando ReactJS para las interfaces de usuario.
- **server:** Incluye la inicialización principal del servidor, como las configuraciones de MeteorJS. Se relaciona con la **Lógica del Servidor** (*Backend*), donde se gestionan las publicaciones y la lógica general.
- **imports:** Es la carpeta principal donde reside toda la lógica del negocio y los componentes esenciales de la aplicación.
 - **Modelo:** Gestión de datos y comunicación con la base de datos mediante métodos y publicaciones.
 - **Vista:** Componentes y páginas desarrolladas con ReactJS.

MeteorJS utiliza la carpeta **imports** para diferenciar el código que debe cargarse explícitamente mediante **importaciones** del que se carga automáticamente. Esto permite mantener el proyecto modular y tener mayor control sobre qué partes del código se ejecutan en cada momento.

- **public:** Almacena recursos públicos como imágenes, íconos y configuraciones compartidas, accesibles directamente desde el cliente. Se relaciona con la **Vista**, proporcionando recursos visuales y configuraciones estáticas.
- **Archivos de configuración** (*Dockerfile* y *docker-compose.yml*): Son esenciales para la contenerización y despliegue de la aplicación utilizando Docker. Garantizan la consistencia del entorno de desarrollo y producción.

¿Por qué toda la lógica queda en imports?

MeteorJS adopta la convención de mantener toda la lógica en **imports** para permitir un control explícito sobre qué partes del código se ejecutan en cada contexto. Aunque el sistema no es distribuido, esta estructura:

- **Facilita la modularidad:** Los desarrolladores pueden importar únicamente los módulos necesarios, evitando cargar código innecesario.
- **Mejora la organización:** Divide la lógica del negocio (*api*) y la interfaz (*ui*) en subcarpetas bien definidas.
- **Sienta las bases para la escalabilidad:** Aunque el sistema es cohesivo, esta práctica prepara el proyecto para una eventual transición a arquitecturas más complejas, como microservicios.

Relación entre Carpetas y Componentes del Sistema

Cada carpeta cumple un papel específico en el modelo *Modelo-Vista-Controlador* (MVC):

- **Modelo:** Reside en `imports/api`, donde se define la interacción con la base de datos y la lógica del negocio.
- **Vista:** Se encuentra en `imports/ui`, que incluye los componentes de ReactJS responsables de la interacción con el usuario.
- **Controlador:** Se implementa en `imports/startup/server` y `imports/startup/client`, donde se inicializan las publicaciones, suscripciones y rutas.

Con esta organización, *GeoBio Colombia* garantiza un código limpio, estructurado y preparado para cumplir con los objetivos planteados en el proyecto.

4.4.6. Conclusión de la Arquitectura

La arquitectura de *GeoBio Colombia* ofrece una solución integrada y eficiente para un sistema que prioriza el manejo de datos geolocalizados en tiempo real. Aunque inicialmente se optó por un diseño cohesivo, el uso de buenas prácticas y tecnologías modernas asegura que el sistema pueda evolucionar hacia arquitecturas más complejas, como microservicios, si se requiere en futuras iteraciones.

4.5. Implementación de Algoritmos

En el desarrollo de *GeoBio Colombia*, se implementó un algoritmo para calcular distancias geográficas entre coordenadas, esencial para las funcionalidades relacionadas con la geolocalización de especies. El algoritmo utilizado es el de Haversine, ampliamente reconocido por su precisión en cálculos geográficos.

4.5.1. Algoritmo de Haversine

El algoritmo de Haversine[30] permite calcular la distancia entre dos puntos en la superficie de una esfera, en este caso, la Tierra. Este cálculo es fundamental para determinar la distancia entre la ubicación del usuario y las especies registradas en el sistema.

4.5.2. Descripción del Algoritmo

El algoritmo toma como entrada las coordenadas geográficas (latitud y longitud) de dos puntos y retorna la distancia en metros. Los pasos principales del cálculo incluyen:

- Convertir las coordenadas de grados a radianes.
- Calcular la diferencia de latitudes y longitudes.
- Utilizar la fórmula de Haversine para obtener la distancia en kilómetros.
- Convertir la distancia a metros.

Algorithm 1 Algoritmo de Haversine**Require:** Coordenadas del punto 1 (lat_1, lon_1), Coordenadas del punto 2 (lat_2, lon_2)**Ensure:** Distancia en metros d

```

1: function TORAD( $x$ ): return  $(x \times \pi)/180$ 
2: end function
3:  $R \leftarrow 6371$  ▷ Radio de la Tierra en kilómetros
4:  $dLat \leftarrow \text{toRad}(lat_2 - lat_1)$ 
5:  $dLon \leftarrow \text{toRad}(lon_2 - lon_1)$ 
6:  $a \leftarrow \sin^2(dLat/2) + \cos(\text{toRad}(lat_1)) \times \cos(\text{toRad}(lat_2)) \times \sin^2(dLon/2)$ 
7:  $c \leftarrow 2 \times \text{atan2}(\sqrt{a}, \sqrt{1-a})$ 
8:  $d \leftarrow R \times c \times 1000$  ▷ Distancia en metros return  $d$ 

```

4.5.3. Aplicación del Algoritmo

El algoritmo de Haversine[30] se integra en el sistema para calcular distancias entre:

- La ubicación actual del usuario (obtenida mediante el navegador) y las especies registradas en la base de datos.
- Dos puntos específicos en el mapa interactivo, permitiendo a los usuarios medir distancias geográficas con precisión.

4.5.4. Ventajas del Algoritmo

- Precisión en cálculos geográficos incluso para distancias pequeñas.
- Eficiencia computacional, ideal para aplicaciones en tiempo real.
- Fácil integración con tecnologías web modernas como JavaScript.

4.5.5. Conclusión

El uso del algoritmo de Haversine en *GeoBio Colombia* asegura una experiencia precisa y eficiente para los usuarios, mejorando la funcionalidad de geolocalización y permitiendo una exploración más intuitiva de la biodiversidad en la región.

Capítulo 5

Despliegue

5.1. Proceso de Despliegue de la Aplicación Web

El despliegue de la aplicación *GeoBio Colombia* se realizó utilizando una instancia de AWS EC2, Docker[5] para la contenedorización y Cloudflare para la gestión de seguridad y accesibilidad del dominio. Este despliegue garantiza una experiencia fluida y segura para los usuarios finales. A continuación, se describen las etapas principales del proceso de despliegue.

5.1.1. Planeación y Selección de Herramientas

Antes de iniciar el despliegue, se definieron los siguientes objetivos:

- Asegurar una infraestructura escalable y de alta disponibilidad.
- Garantizar la seguridad de los datos mediante HTTPS y medidas contra ataques.
- Facilitar futuras actualizaciones y mantenimiento con un entorno contenerizado.

Para cumplir con estos objetivos, se seleccionaron herramientas modernas y confiables:

- **AWS EC2:** Proporciona una plataforma escalable y personalizable para alojar la aplicación.
- **Docker:** Facilita la contenedorización de la aplicación, asegurando consistencia entre entornos.
- **Cloudflare:** Mejora la seguridad del sistema y la accesibilidad del dominio.

5.1.2. Configuración de la Instancia EC2

Se utilizó una instancia de AWS EC2 con las siguientes especificaciones:

- **Imagen:** Ubuntu 22.04
- **Tipo de instancia:** t3.medium
- **Memoria RAM:** 4 GB
- **Almacenamiento:** 20 GB

Pasos para la Configuración Inicial:

1. Creación de la instancia EC2 en la consola de AWS.

2. Configuración del grupo de seguridad para permitir tráfico HTTP (puerto 80) y HTTPS (puerto 443).
3. Instalación de paquetes básicos, como actualizaciones del sistema y Docker.

Además, se realizó la configuración del firewall de Ubuntu para reforzar la seguridad del sistema operativo.

5.1.3. Contenerización con Docker

Docker fue empleado para crear un entorno aislado y consistente para la aplicación. Las etapas incluyeron:

- Creación de un archivo `Dockerfile` para definir las dependencias y configuración de la aplicación.
- Construcción de la imagen Docker con el siguiente comando:

```
docker build -t geobio-colombia .
```

- Ejecución del contenedor con las configuraciones de red adecuadas:

```
docker run -d -p 80:3000 --name geobio-app geobio-colombia
```

Esta configuración permitió encapsular todas las dependencias del proyecto, asegurando su portabilidad y minimizando conflictos entre entornos.

5.1.4. Configuración de Cloudflare

El dominio `geobiocolombia.online` fue configurado en Cloudflare para mejorar la seguridad y accesibilidad. Las configuraciones clave incluyeron:

- **Configuración de DNS:** Asignación de la dirección IP pública de la instancia EC2 al dominio mediante un registro A.
- **Certificado HTTPS:** Activación de HTTPS para cifrar las comunicaciones entre los usuarios y el servidor.
- **Firewall:** Restricción de acceso a usuarios fuera de Colombia, bloqueando tráfico no autorizado.
- **Protección contra DDoS:** Activación de medidas avanzadas contra ataques de denegación de servicio.

5.1.5. Flujo de Despliegue

El flujo de despliegue se organizó en las siguientes etapas:

- **Configuración de la Instancia:** Instalación de dependencias y configuración inicial de la instancia EC2.
- **Construcción y Ejecución de Contenedores:** Contenerización de la aplicación utilizando Docker y su ejecución en la instancia.

- **Configuración de Dominio y Seguridad:** Configuración de DNS, HTTPS y medidas de seguridad mediante Cloudflare.
- **Pruebas de Validación:** Verificación del acceso al dominio, estabilidad del contenedor y funcionamiento de las configuraciones de seguridad.

5.1.6. Pruebas y Validación del Despliegue

Después del despliegue, se realizaron pruebas exhaustivas para validar el funcionamiento del sistema:

- **Acceso al dominio:** Validación de la accesibilidad de `geobiocolombia.online` desde diferentes dispositivos y navegadores.
- **Rendimiento:** Evaluación del desempeño del contenedor Docker bajo cargas simuladas.
- **Seguridad:** Verificación de la implementación de HTTPS y las reglas del firewall.

5.1.7. Dificultades y Soluciones

Durante el proceso de despliegue, se presentaron varios desafíos, entre ellos:

- **Configuración del Firewall:** Hubo dificultades iniciales para restringir el acceso a usuarios fuera de Colombia, que se resolvieron ajustando las reglas de Cloudflare y el firewall de Ubuntu.
- **Errores en la Contenerización:** Se detectaron problemas en la construcción del contenedor debido a dependencias faltantes, solucionados mediante ajustes en el archivo `Dockerfile`.
- **Integración de HTTPS:** La activación del certificado HTTPS presentó conflictos con configuraciones previas del dominio, resueltos revisando las políticas de seguridad en Cloudflare.

5.1.8. Diagrama de Despliegue

El siguiente diagrama UML muestra cómo se integran los diferentes componentes del sistema en el proceso de despliegue de *GeoBio Colombia*. La instancia EC2, configurada en AWS, ejecuta un contenedor Docker que integra tanto el frontend como el backend de la aplicación. Cloudflare se utiliza para gestionar el tráfico, aplicar medidas de seguridad y garantizar el acceso seguro a través de HTTPS.

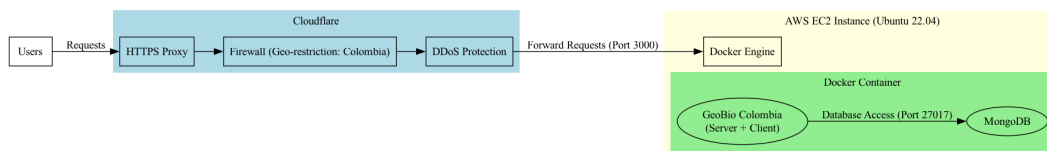


Figura 5.1: Diagrama de despliegue que ilustra la relación entre los componentes principales.

5.1.9. Conclusión

El despliegue en AWS EC2, utilizando Docker y Cloudflare, asegura un entorno robusto y seguro para la operación de *GeoBio Colombia*. Este enfoque facilita la gestión, escalabilidad y protección del sistema, ofreciendo una experiencia confiable y eficiente para los usuarios.

Capítulo 6

Pruebas

6.1. Pruebas de Usabilidad y Funcionalidad

6.1.1. Introducción

El objetivo de las pruebas de usabilidad y funcionalidad fue evaluar la facilidad de uso, la efectividad y la satisfacción general de los usuarios con respecto a la aplicación web *GeoBio Colombia*. Estas pruebas se realizaron mediante encuestas distribuidas a un grupo representativo de usuarios, recopilando sus opiniones y experiencias sobre diversos aspectos de la aplicación.

6.1.2. Metodología

Para llevar a cabo las pruebas, se diseñó un cuestionario usando Google Forms[31], estructurado en cuatro categorías principales, enfocadas en evaluar diferentes aspectos de la aplicación:

- **Usabilidad General:** Preguntas diseñadas para todos los usuarios, enfocadas en evaluar la facilidad de navegación y el diseño intuitivo de la interfaz.
- **Funcionalidad General de la Aplicación:** Consultas sobre el rendimiento de las funcionalidades principales, como la precisión de la información presentada y la rapidez de carga.
- **Específicas para Usuarios Regulares:** Preguntas dirigidas a evaluar aspectos como la experiencia con el mapa de geolocalización y el uso de filtros para encontrar especies.
- **Específicas para Administradores:** Evaluación de funcionalidades administrativas relacionadas con la gestión y actualización de información sobre especies.

El diseño de las preguntas se basó en las funcionalidades establecidas como objetivos en el anteproyecto, como el correcto funcionamiento de los filtros de búsqueda y la visualización geográfica precisa de las especies en el mapa. Además, para garantizar una evaluación sólida de la usabilidad, las preguntas se alinearon con los principios establecidos en las 10 heurísticas de usabilidad de Jacob Nielsen[32], como la consistencia, el control del usuario y la prevención de errores.



Figura 6.1: Principios de usabilidad de Jacob Nielsen[32].

El cuestionario fue distribuido entre estudiantes de la Universidad del Valle, sede Tuluá, pertenecientes a las carreras de Desarrollo de Software e Ingeniería de Sistemas. Este público objetivo fue seleccionado debido a su familiaridad con tecnologías digitales, lo que facilitó una evaluación informada y detallada de la aplicación. En total, participaron 41 estudiantes, superando el mínimo requerido de 25 participantes para garantizar una muestra representativa.

6.1.3. Pruebas de Usabilidad

Facilidad de Navegación

Pregunta: ¿Qué tan fácil te resultó navegar por la aplicación?
Resultados: La mayoría de los usuarios encontró la navegación fácil o muy fácil, indicando que la interfaz es intuitiva y adecuada para explorar las funcionalidades de la aplicación.

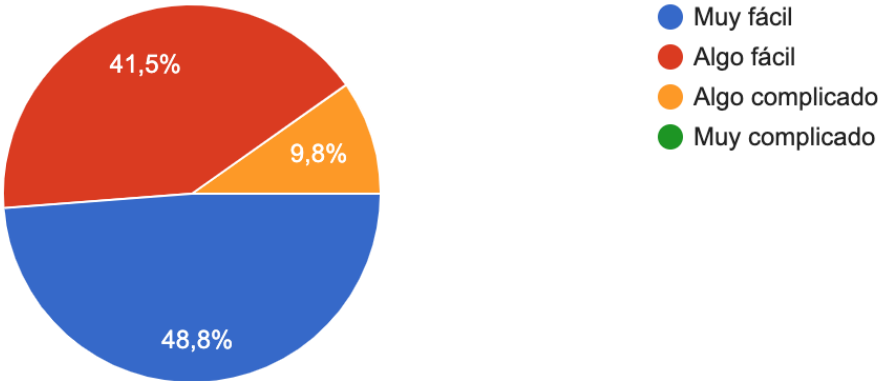


Figura 6.2: Gráfica sobre la facilidad de navegación en la aplicación

Carga y Rendimiento

Pregunta: ¿La aplicación se carga y funciona con rapidez en tu dispositivo?

Resultados: Un alto porcentaje de usuarios consideró que la aplicación carga rápidamente, aunque algunos reportaron ligeros retrasos, lo que sugiere un buen desempeño general pero con espacio para optimización.

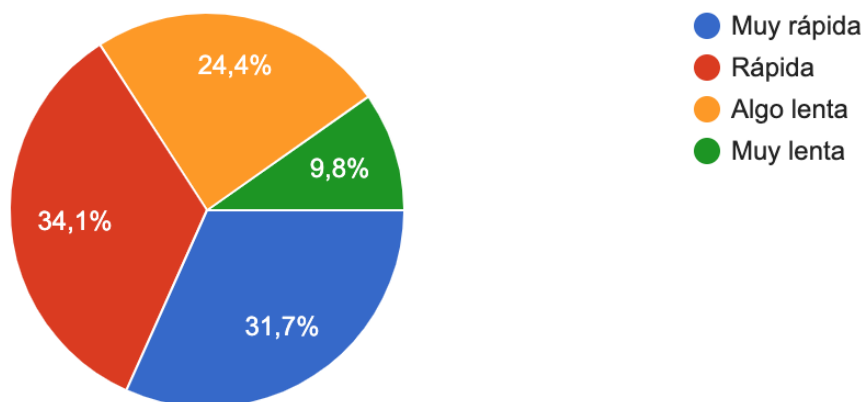


Figura 6.3: Gráfica sobre el tiempo de carga y rendimiento

6.1.4. Pruebas de Funcionalidad

Precisión de la Información de las Especies

Pregunta: ¿Qué tan precisa y útil consideras la información sobre las especies?

Resultados: Los usuarios calificaron la información como muy precisa y útil en su mayoría, validando la calidad de los datos presentados.

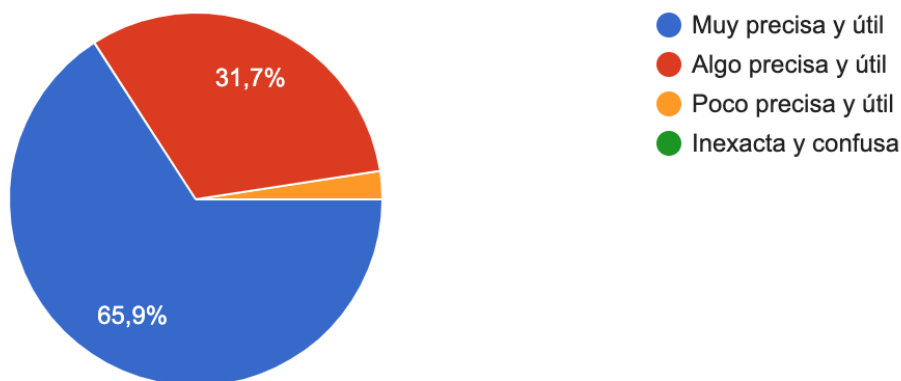


Figura 6.4: Gráfica sobre la precisión y utilidad de la información de las especies

Función de Geolocalización

Pregunta: ¿La función de geolocalización mejoró tu experiencia al usar la aplicación?

Resultados: La mayoría de los usuarios destacó que la función de geolocalización mejoró significativamente su experiencia, facilitando la interacción con la información geográfica.

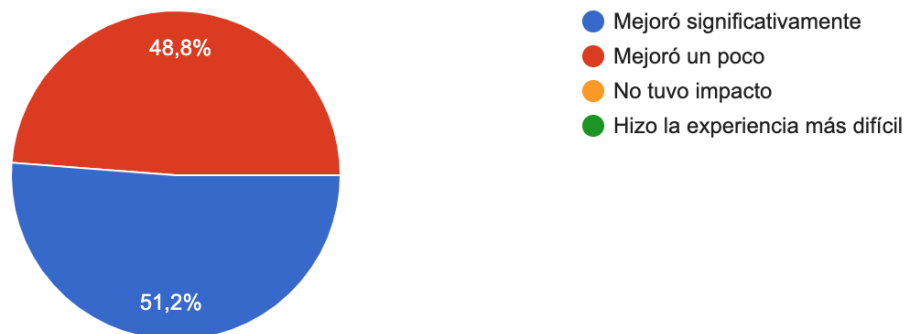


Figura 6.5: Gráfica sobre la mejora en la experiencia gracias a la geolocalización

6.1.5. Análisis de los Resultados Más Bajos

Se analizaron las preguntas con puntajes más bajos para identificar áreas de mejora en la aplicación. Entre las principales observaciones:

- Algunos usuarios reportaron ligeros retrasos en la carga en dispositivos más antiguos.
- Aunque la mayoría encontró útil la información sobre las especies, se sugirió mejorar la organización de algunos datos específicos.
- La navegación, si bien fue en general intuitiva, podría beneficiarse de una guía más interactiva.

A pesar de los puntos señalados, la aplicación logró resultados satisfactorios en usabilidad y funcionalidad. Se recomienda monitorear continuamente el feedback de los usuarios y realizar iteraciones para optimizar la experiencia y asegurar el cumplimiento de los objetivos educativos de la plataforma.

Capítulo 7

Conclusiones

7.1. Conclusiones

Este proyecto aborda la creación de una aplicación web para registrar y consultar información sobre la fauna de la región del Pacífico colombiano. Integrando funcionalidades como geolocalización y registros de especies. La aplicación permite a los usuarios acceder a información sobre especies de fauna, realizar búsquedas personalizadas y visualizar datos mediante mapas interactivos, todo esto a través de una interfaz web intuitiva. Al estar optimizada para múltiples dispositivos, se asegura una experiencia accesible tanto para expertos en biodiversidad como para ciudadanos interesados en contribuir al conocimiento y la conservación de la naturaleza.

El sistema desarrollado incluye un algoritmo de distancia de Haversine, que permite calcular distancias geográficas de manera precisa. Esta funcionalidad ha demostrado ser importante para identificar la proximidad de los usuarios a las ubicaciones de interés, mejorando significativamente la experiencia de exploración y aprendizaje. Además, la implementación de mapas interactivos utilizando Leaflet potencia la visualización de datos espaciales, facilitando la identificación de áreas ricas en biodiversidad y fomentando una conexión más cercana con el entorno natural.

El despliegue de la aplicación, realizado en un entorno de producción utilizando tecnologías como Docker y AWS EC2, ha permitido garantizar la escalabilidad, seguridad y disponibilidad del sistema. La integración con Cloudflare proporciona medidas adicionales de protección contra ataques DDoS, mientras que la implementación de HTTPS asegura conexiones cifradas, generando confianza entre los usuarios al interactuar con la plataforma.

Una de las fortalezas del proyecto es su arquitectura integrada, diseñada para facilitar el mantenimiento y la implementación de nuevas funcionalidades. Aunque se optó por una solución centralizada, el uso de tecnologías modernas y estándares como React para el frontend y Meteor.js para el backend asegura un desarrollo eficiente y una comunicación fluida entre las distintas capas de la aplicación. Asimismo, el uso de MongoDB como base de datos permite una gestión eficiente de los datos, asegurando consultas rápidas y almacenamiento escalable.

El desarrollo de este proyecto ha representado un importante desafío técnico y personal. A lo largo del proceso, se adquirieron valiosas habilidades en tecnologías web, gestión de servidores y despliegue en entornos en la nube. La experiencia de trabajar con geolocalización y visualización de datos geográficos ha fortalecido mi capacidad para resolver problemas complejos y desarrollar soluciones tecnológicas con un

impacto positivo en la sociedad. Además, el manejo de variables de entorno en Docker, la configuración de servidores y la integración de dominios con servicios como Cloudflare han sido aprendizajes significativos que enriquecen mi formación como ingeniero de sistemas.

Aunque la aplicación ha logrado cumplir con sus objetivos principales, como proporcionar una herramienta educativa y funcional para la exploración de la biodiversidad, existen áreas que pueden mejorarse. Por ejemplo, se pueden implementar algoritmos de inteligencia artificial para ofrecer recomendaciones personalizadas basadas en los intereses del usuario, así como optimizar el rendimiento de la aplicación para manejar un mayor volumen de usuarios simultáneamente. También sería valioso incluir funcionalidades adicionales, como la posibilidad de realizar análisis estadísticos sobre los datos recopilados.

Esta aplicación web sienta un precedente para futuros proyectos orientados a la conservación y educación ambiental. Su capacidad para involucrar a los usuarios en el descubrimiento y aprendizaje sobre la biodiversidad colombiana tiene el potencial de generar un impacto significativo, no solo en términos educativos, sino también en la concienciación sobre la importancia de proteger y preservar el medio ambiente. Las bases tecnológicas establecidas permiten vislumbrar un futuro prometedor para la evolución de esta herramienta y su contribución al conocimiento y la conservación de la biodiversidad en Colombia.

Capítulo 8

Trabajos futuros

8.1. Trabajos Futuros

La aplicación web desarrollada para la fauna de la región del Pacífico colombiano ha sido diseñada con una arquitectura modular, permitiendo futuras expansiones y mejoras sin comprometer la estructura existente. A continuación, se presentan posibles áreas de desarrollo y mejoras futuras:

8.1.1. Mejoramiento del Sistema de Recomendaciones

El sistema de recomendaciones podría beneficiarse significativamente de las siguientes mejoras:

- **Integración de Algoritmos de Machine Learning:** Incorporar técnicas avanzadas de aprendizaje automático, como redes neuronales profundas, para mejorar la precisión y personalización de las recomendaciones.
- **Variables Adicionales:** Agregar más variables, como patrones de comportamiento de los usuarios y condiciones geográficas, para enriquecer el sistema de recomendaciones.
- **Adaptación Basada en Temporada:** Implementar lógicas adaptativas que prioricen especies según la estación del año o cambios en las poblaciones de fauna.

8.1.2. Optimización del Sistema de Búsqueda

Para mejorar la experiencia del usuario en la búsqueda de especies:

- **Sugerencias Predictivas:** Integrar funciones de autocompletado que sugieran especies mientras el usuario escribe, optimizando el tiempo de búsqueda.
- **Pruebas de Desempeño:** Realizar evaluaciones regulares para garantizar que el motor de búsqueda responda rápidamente y sin errores.

8.1.3. Migración a la Nube

Para garantizar escalabilidad y disponibilidad:

- **Servicios en la Nube:** Migrar la infraestructura actual a plataformas como AWS o Azure, aprovechando herramientas avanzadas de monitoreo y redundancia.

8.1.4. Mejora de la Interfaz de Usuario

La experiencia del usuario puede optimizarse mediante:

- **Retroalimentación Constante:** Analizar comentarios de los usuarios para ajustar el diseño y mejorar la navegación.
- **Pruebas de Usabilidad:** Implementar ciclos continuos de pruebas para identificar problemas y adaptar la interfaz a las necesidades de los usuarios.

8.1.5. Integración de Nuevas Funcionalidades

Desarrollar herramientas adicionales para enriquecer la aplicación:

- **Mapeo Avanzado:** Introducir capas geográficas adicionales que permitan a los usuarios identificar especies en mapas interactivos con mayor detalle.
- **Reportes Personalizados:** Habilitar la generación de informes detallados basados en los datos consultados.

8.1.6. Fortalecimiento del Panel de Administradores

Mejorar las capacidades de gestión para los administradores:

- **Análisis de Datos:** Incorporar gráficos y paneles interactivos que faciliten el monitoreo de contribuciones y datos.
- **Herramientas de Moderación:** Desarrollar controles más robustos para la validación de datos aportados por usuarios.

8.1.7. Actualización y Evaluación Continua

El campo de la tecnología evoluciona rápidamente, por lo que es fundamental:

- **Incorporar Nuevas Tecnologías:** Evaluar periódicamente nuevas herramientas y métodos para garantizar la relevancia del sistema.
- **Capacitación Constante:** Asegurar que el equipo de desarrollo esté preparado para implementar los cambios necesarios.

Referencias

- [1] Martin Fowler. *Patterns of Enterprise Application Architecture*. Sección sobre arquitectura monolítica. Addison-Wesley Professional, 2002. URL: <https://martinfowler.com/eaCatalog/>.
- [2] Meteor Development Group. *MeteorJS Documentation*. 2024. URL: <https://docs.meteor.com/>.
- [3] Meta Platforms. *ReactJS Documentation*. 2024. URL: <https://reactjs.org/docs/getting-started.html>.
- [4] Amazon Web Services. *Amazon EC2 Documentation*. 2024. URL: <https://docs.aws.amazon.com/ec2/>.
- [5] Docker. *Docker Documentation*. 2024. URL: <https://docs.docker.com/>.
- [6] Cloudflare. *DDoS Protection: Cloudflare Documentation*. 2024. URL: <https://www.cloudflare.com/ddos/>.
- [7] GoDaddy. *GoDaddy Documentation: Domain Management*. 2024. URL: <https://www.godaddy.com/help/>.
- [8] Axios. *Axios Documentation*. 2024. URL: <https://axios-http.com/docs/intro>.
- [9] Flowbite. *Flowbite Documentation*. 2024. URL: <https://flowbite.com/docs/>.
- [10] L. VÉLEZ. *Desconocimiento sobre biodiversidad en Colombia: motor de su deterioro*. 2013. URL: https://www.elcolombiano.com/historico/desconocimiento_sobre_biodiversidad_en_colombia_motor_de_su_deterioro-eyec_265414.
- [11] Constanza Barrera y Romina Bahamondes. “Turismo Sostenible: Importancia en el cuidado del medio ambiente”. En: *Revista Interamericana de Ambiente y Turismo-RIAT* 8.1 (2012), págs. 50-56. URL: http://handbook.usfx.bo/nueva/vicerrectorado/citas/ECONOMICAS_6/Ingenieria_Comercial/58.pdf.
- [12] Miguel Ángel Beltrán Aguirre. “Las TIC como herramientas para el reconocimiento y apropiación de la biodiversidad”. En: (2023). URL: <https://hdl.handle.net/10495/35936>.
- [13] Giovanni Orlando Cancino Escalante y Alejandro Chaparro Giraldo. “Estado de la investigación científica y el acceso a los recursos genéticos por grupos de investigación colombianos”. En: *Revista Colombiana de Biotecnología* 22.1 (2020), págs. 70-78. URL: <https://www.redalyc.org/journal/776/77664304008/html/>.
- [14] Lady Johanna Conde Nieto y Martha Johanna Vargas Obando. “Uso y apropiación de recursos educativos digitales en el aula: una aproximación pedagógica para el reconocimiento de la biodiversidad del entorno de los estudiantes de grado 601 en la Institución Educativa Manuel Uribe Ángel Medellín, Antioquia”. Tesis doct. Universidad de Cartagena, 2022. URL: <https://hdl.handle.net/11227/16452>.
- [15] React-Leaflet. *React-Leaflet Documentation*. 2024. URL: <https://react-leaflet.js.org/>.

- [16] Cloudflare. *Cloudflare Documentation*. 2024. URL: <https://developers.cloudflare.com/>.
- [17] Paul A. Longley et al. *Geographic Information Science and Systems*. John Wiley & Sons, 2015.
- [18] Kang-Tsung Chang. *Introduction to Geographic Information Systems*. 4.^a ed. Boston: McGraw-hill, 2008.
- [19] Edward O. Wilson. “Biodiversity”. En: (1988).
- [20] Rick Bonney et al. “Citizen science: a developing tool for expanding science knowledge and scientific literacy”. En: *BioScience* 59.11 (2009), págs. 977-984.
- [21] Richard B. Primack. *A primer of conservation biology*. 1995.
- [22] Kevin J. Gaston y John I. Spicer. *Biodiversity: an introduction*. John Wiley & Sons, 2013.
- [23] Don Norman. *The Design of Everyday Things: Revised and Expanded Edition*. Basic books, 2013.
- [24] Bruce Lawson y Remy Sharp. *Introducing HTML5*. New Riders, 2011.
- [25] GBIF. *GBIF API Documentation*. 2024. URL: <https://www.gbif.org/developer/summary>.
- [26] IUCN Red List of Threatened Species. *IUCN Red List Documentation*. 2024. URL: <https://www.iucnredlist.org/>.
- [27] Figma. *Figma Documentation*. 2024. URL: <https://help.figma.com/>.
- [28] TailwindCSS. *TailwindCSS Documentation*. 2024. URL: <https://tailwindcss.com/docs>.
- [29] GeoJSON. *GeoJSON Specification*. 2024. URL: <https://geojson.org/>.
- [30] Chris Veness. *Calculate Distance Using the Haversine Formula*. 2024. URL: <https://www.movable-type.co.uk/scripts/latlong.html>.
- [31] Google. *Google Forms Documentation*. 2024. URL: <https://support.google.com/forms/>.
- [32] Jakob Nielsen. *10 Usability Heuristics for User Interface Design*. 1994. URL: <https://www.nngroup.com/articles/ten-usability-heuristics/>.

Capítulo 9

Anexos

El código fuente del proyecto puede encontrarse en el siguiente repositorio de GitHub:

<https://github.com/gengisKhanDev/pdg-geobio-colombia/tree/developer>

Enlace para ver el video que muestra el funcionamiento del proyecto de grado.

<https://drive.google.com/file/d/1phMno3vZW5qmdshGYgrbjfyL0ZsBITaj/view?t=19>