



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

TRABAJO DE FIN DE GRADO

INGENIERÍA INFORMÁTICA – INGENIERÍA DEL SOFTWARE

AvIAry

Aplicación multiplataforma para clasificación e identificación de especies de aves

Realizado por

JOSE MARÍA GARCÍA QUIJADA

Dirigido por

JUAN PEDRO DOMÍNGUEZ MORALES

Departamento

ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES

Sevilla, junio de 2023

AGRADECIMIENTOS

Quisiera expresar mi profundo agradecimiento a todos aquellos que contribuyeron de manera significativa a la finalización de mi último proyecto en la carrera de Ingeniería Informática. Este viaje ha sido una travesía llena de desafíos y aprendizajes, y no podría haber alcanzado este hito sin el apoyo y la colaboración de diversas personas.

En primer lugar, quiero expresar mi gratitud a mis profesores a lo largo de este camino, tanto aquellos que me han enseñado lo bueno, como aquellos que me han enseñado lo duro y lo difícil del camino. Sus consejos y conocimientos han sido pilares fundamentales para el éxito de este proyecto y mi crecimiento como profesional.

Un agradecimiento especial a mis compañeros de clase, quienes han compartido este trayecto académico conmigo. El apoyo incondicional entre nosotros, el hambre por competir para vacilar quien era el mejor posicionado en cada momento y sobre todo el querer avanzar juntos en cada asignatura, han creado recuerdos inolvidables.

No puedo dejar de reconocer el apoyo incondicional de mi familia y amigos. Su paciencia infinita han sido el pilar emocional que necesitaba para superar los desafíos y mantenerme enfocado en mis objetivos.

Finalmente, quiero agradecer a dos de mis pilares fundamentales que soñaban con verme estar en la posición en la que hoy me encuentro, con todos los logros que he conseguido hasta ahora bajo mis brazos, que han luchado y que confiaban plenamente en mí desde el principio, incluyo cuando ni yo mismo confiaba y que han permitido que hoy sea la persona que soy, abuelo y papá, desde donde quiera que estéis, espero que ambos estéis orgullosos de mí.

Este último trabajo representa el cierre de un capítulo significativo en mi carrera, y cada uno de ustedes habéis sido una parte esencial de este viaje. Con gratitud en mi corazón, miro hacia el futuro con entusiasmo, miedo y coraje. ¡Gracias a todos por ser parte de este logro!

RESUMEN

AvIAry combina la pasión por las aves con tecnología de última generación para ofrecer una experiencia única de observación de aves, impulsada por la inteligencia artificial (IA). Diseñada para entusiastas de las aves y fotógrafos de vida silvestre, la aplicación destaca por su enfoque innovador en la identificación de aves mediante tecnologías avanzadas.

En el núcleo de AvIAry se encuentra un método de IA robusto que utiliza modelos de redes neuronales, incluyendo *Convolutional Neural Networks* (CNN) y *Transformers*. Esta implementación se basa en una exhaustiva investigación que ha comparado y perfeccionado modelos para garantizar identificaciones precisas y rápidas a partir de fotografías de aves.

La aplicación brinda a los usuarios una interfaz accesible y fácil de usar, permitiéndoles cargar imágenes de aves y recibir identificaciones instantáneas respaldadas por la sofisticada tecnología de IA. AvIAry no solo es una herramienta de identificación; es una ventana digital a la diversidad aviar potenciada por tecnologías de última generación.

En resumen, AvIAry es mucho más que una aplicación de identificación de aves. Es un testimonio de cómo la tecnología avanzada puede potenciar nuestra conexión con la naturaleza. Bienvenido a AvIAry, donde la inteligencia artificial y la observación de aves se encuentran para ofrecer una experiencia diferente a lo común a los apasionados de la ornitología.

PALABRAS CLAVE

Identificación de Aves, Inteligencia Artificial, Ornitología, Redes Neuronales, Aplicación Web, Modelos Avanzados de IA, Comunidad de Observadores de Aves, Interfaz Intuitiva, Identificación Instantánea, Colaboración en Línea, Experiencia Educativa, Fotografía de Aves, Conservación Aviar, Innovación Tecnológica.

ABSTRACT

AvIAry represents the convergence of a love for birds and cutting-edge technology, offering a unique bird-watching experience driven by artificial intelligence (AI). Designed for bird enthusiasts and wildlife photographers, the application stands out for its innovative approach to bird identification using advanced technologies.

At the core of AvIAry is a robust AI system utilizing neural network models, including *Convolutional Neural Networks* (CNN) and *Transformers*. This implementation is based on exhaustive research that has compared and refined models to ensure precise and rapid bird identifications from photographs.

The application provides users with an accessible and user-friendly interface, allowing them to upload bird images and receive instant identifications backed by sophisticated AI technology. AvIAry is not just an identification tool; it is a digital window into avian diversity powered by state-of-the-art technologies.

In summary, AvIAry is much more than a bird identification app. It is a testament to how advanced technology can enhance our connection with nature. Welcome to AvIAry, where artificial intelligence and bird watching come together to offer a different experience to ornithology enthusiasts.

KEYWORDS

Bird Identification, Artificial Intelligence, Ornithology, Neural Networks, Web Application, Advanced AI Models, Community of Bird Observers, Intuitive Interface, Instant Identification, Online Collaboration, Educational Experience, Bird Photography, Avian Conservation, Technological Innovation

ÍNDICE DE CONTENIDOS

A) DESCRIPCIÓN DEL PROYECTO	10
1. INTRODUCCIÓN Y MOTIVACIÓN	10
2. OBJETIVOS DEL PROYECTO	11
2.1 OBJETIVOS A NIVEL PROFESIONAL	11
2.2 OBJETIVOS A NIVEL EDUCACIONAL	12
2.3 OBJETIVOS A NIVEL PERSONAL	13
3. ESTADO DEL ARTE	14
3.1 MERLIN BIRD ID (CORNELL LAB)	14
3.2 eBIRD (CORNELL LAB Y AUDUBON)	14
3.3 iBIRD	15
3.4 BIRDNET	16
3.5 AUDUBON BIRD GUIDE	16
4. ELICITACIÓN DE REQUISITOS	17
4.1 REQUISITOS	17
4.2 RIESGOS	21
B) EJECUCIÓN DEL PROYECTO	23
5. DISEÑO Y ARQUITECTURA DEL SISTEMA Y TECNOLOGÍAS EMPLEADAS EN ÉL	23
5.1 FRONT-END: INTERACTIVIDAD Y ESTILO UNIFICADOS	24
5.2 SERVIDOR DE IA: PODER DE APRENDIZAJE PROFUNDO	24
5.3 BACK-END: POTENCIA NOSQL DE MONGODB	24
5.4 FLUJO DETALLADO DEL PROCESO: UN BALLET TECNOLÓGICO	25
5.5 RAZONES DETRÁS DE LAS ELECCIONES TECNOLÓGICAS	26
5.6 CONCLUSIÓN: UNA SINFONÍA TECNOLÓGICA EQUILIBRADA	27
6. IMPLEMENTACIÓN Y DESARROLLO	27
6.1 FRONT-END: DISEÑO INTERACTIVO CON REACT Y BOOTSTRAP	27
❖ Componente 'Inicio'	28
❖ Componente 'Nuestra Tecnología'	30
❖ Componente 'IA'	31
❖ Componente 'Características'	33
❖ Componente 'Imagen Cortante'	35
❖ Componente 'Equipo'	36
❖ Componente 'Preguntas frecuentes'	38
❖ Componente 'Marcas'	39
❖ Componente 'Pie de página'	41

❖	Componente ' <i>Biodiversidad</i> '	42
❖	Componente ' <i>Artículos</i> '	43
❖	Componente ' <i>Artículo</i> '	44
6.2	SERVIDOR IA: LA CREACIÓN DE UNA NUEVA FORMA DE PENSAR	46
6.3	BACK-END: EL ALMACÉN DE INFORMACIÓN DE AVIARY	69
6.4	¿QUÉ ES REALMENTE AVIARY? ¿UNA APP O UNA WEB?	72
7.	PRUEBAS DEL SISTEMA	74
7.1	PRUEBAS UNITARIAS CON JEST	75
7.2	PRUEBAS DE INTERFAZ DE USUARIO (UI)	75
7.3	PRUEBAS DE NAVEGADOR CRUZADO	75
7.3	PRUEBAS DE VELOCIDAD CON WEBPAGETEST.COM	76
7.4	PRUEBAS DE RENDIMIENTO CON K6 Y GRAFANA LABS	76
7.5	REPORTE GENERAL CON LIGHTHOUSE DE GOOGLE	78
C)	PLANIFICACIÓN DEL PROYECTO	79
8.	PLANIFICACIÓN INICIAL	79
8.1	PLANIFICACIÓN TEMPORAL	79
8.2	PLANIFICACIÓN FINANCIERA	82
9.	PLANIFICACIÓN FINAL	82
9.1	PLANIFICACIÓN TEMPORAL	82
9.2	PLANIFICACIÓN FINANCIERA	83
10.	ESTUDIO DE MERCADO	83
10.1	CLIENTES POTENCIALES	83
10.2	PLAN DE COMERCIALIZACIÓN	84
10.3	ESTRATEGIAS DE MARKETING	85
11.	CONCLUSIONES	85
12.	TRABAJO FUTURO	86
12.1	EXPANSIÓN DE LA BASE DE DATOS DE ESPECIES	86
12.2	DESARROLLO DE COLABORACIONES EDUCATIVAS	86
12.3	IMPLEMENTACIÓN DE FUNCIONALIDADES DE REGISTRO OPCIONAL	86
12.4	INTEGRACIÓN DE REDES SOCIALES	86
12.5	DIVERSIFICACIÓN DE MODELOS DE IA	87
12.6	USO DE PLATAFORMAS DE PAGO PARA LA ARQUITECTURA	87
13.	BIBLIOGRAFÍA Y ANEXOS	87

ÍNDICE DE ILUSTRACIONES Y TABLAS

ILUSTRACIONES

Imagen 1: Flujo de la aplicación	25
Imagen 2: Mock up componente Inicio	28
Imagen 3: Componente Inicio final	29
Imagen 4: Mock up componente Inicio (móvil)	29
Imagen 5: Componente Inicio final (móvil)	29
Imagen 6: Componente Inicio con predicción	29
Imagen 7: PDF con los datos obtenidos	30
Imagen 8: Mock up componente Nuestra tecnología	30
Imagen 9: Componente Nuestra tecnología final	31
Imagen 10: Mock up componente Nuestra tecnología (móvil)	31
Imagen 11: Componente Nuestra tecnología final (móvil)	31
Imagen 12: Mock up componente IA	31
Imagen 13: Componente IA final	32
Imagen 15: Componente IA final (móvil)	32
Imagen 14: Mock up componente IA (móvil)	32
Imagen 16: Mock up componente Características	33
Imagen 17: Componente Características final	34
Imagen 18: Mock up componente Características (móvil)	34
Imagen 19: Componente Características final (móvil)	34
Imagen 20: Mock up componente Imagen Cortante	35
Imagen 21: Componente Imagen Cortante final	35
Imagen 22: Mock up componente Imagen Cortante (móvil)	36
Imagen 23: Componente Imagen Cortante final (móvil)	36
Imagen 24: Mock up componente Equipo	36
Imagen 25: Componente Equipo final	37
Imagen 26: Mock up componente Equipo (móvil)	37
Imagen 27: Componente Equipo final (móvil)	37
Imagen 28: Mock up componente Preguntas Frecuentes	38
Imagen 29: Componente Preguntas Frecuentes final	39
Imagen 30: Mock up componente Preguntas Frecuentes (móvil)	39
Imagen 31: Componente Preguntas Frecuentes final (móvil)	39
Imagen 32: Mock up componente Marcas	39
Imagen 33: Componente Preguntas Frecuentes final	40
Imagen 34: Mock up componente Preguntas Frecuentes (móvil)	40
Imagen 35: Componente Preguntas Frecuentes final (móvil)	40
Imagen 36: Mock up componente Pie de Página	41
Imagen 37: Componente Pie de Página final	41
Imagen 38: Mock up componente Pie de Página (móvil)	41
Imagen 39: Componente Pie de Página final (móvil)	41
Imagen 40: Mock up componente Biodiversidad	42
Imagen 41: Componente Biodiversidad final	42
Imagen 42: Componente Biodiversidad final (móvil)	42
Imagen 43: Mock up componente Biodiversidad (móvil)	42
Imagen 44: Mock up componente Artículos	43
Imagen 45: Componente Artículos final	43

Imagen 46: Componente Biodiversidad final (móvil)	44
Imagen 47: Componente Biodiversidad final (móvil)	44
Imagen 48: Mock up componente Artículo	44
Imagen 49: Componente Artículo final	45
Imagen 50: Mock up componente Artículo (móvil)	45
Imagen 51: Componente Artículo final (móvil)	45
Imagen 52: Endpoint que une al Front-end con la IA	45
Imagen 53: Endpint que une al Front-end con la API REST	46
Imagen 54: Estructura de la arquitectura Transformer	47
Imagen 55: Paso 1 de la acción Encoder-Decoder	48
Imagen 56: Paso 2 de la acción Encoder-Decoder	48
Imagen 57: Paso 3 de la acción Encoder-Decoder	48
Imagen 58: Bloques Encoder y Decoder en Transformer	49
Imagen 59: Esquema sobre la atención	51
Imagen 60: Esquema sobre la atención múltiple	52
Imagen 61: Representación de la atención con oraciones en inglés y francés	52
Imagen 62: Esquema de la estructura ViT	54
Imagen 63: Representación del orden de parches	54
Imagen 64: Relaciones entre capas en una CNN	55
Imagen 65: Mapa conceptual sobre las IA's a crear	57
Imagen 66: Valores en consola durante el entreno con CNN	59
Imagen 67: Gráfica exactitud CNN sin preentrenamiento	59
Imagen 68: Gráfica precisión CNN sin preentrenamiento	59
Imagen 69: Gráfica recall CNN sin preentrenamiento	59
Imagen 70: Gráfica puntuación F1 CNN sin preentrenamiento	59
Imagen 71: Gráfica pérdida en épocas CNN sin preentrenamiento	60
Imagen 72: Gráfica todas las métricas CNN sin preentrenamiento	60
Imagen 73: Gráfica exactitud CNN con preentrenamiento	60
Imagen 74: Gráfica precisión CNN con preentrenamiento	61
Imagen 75: Gráfica recall CNN con preentrenamiento	61
Imagen 76: Gráfica F1 score CNN con preentrenamiento	61
Imagen 77: Gráfica pérdida en épocas CNN con preentrenamiento	62
Imagen 78: Gráfica todas las métricas CNN con preentrenamiento	62
Imagen 79: Gráfica exactitud VIT sin preentrenamiento	64
Imagen 80: Gráfica precisión VIT sin preentrenamiento	64
Imagen 81: Gráfica recall VIT sin preentrenamiento	64
Imagen 82: Gráfica F1 Score VIT sin preentrenamiento	64
Imagen 83: Gráfica pérdidas en épocas VIT sin preentrenamiento	64
Imagen 84: Gráfica todas las métricas VIT sin preentrenamiento	65
Imagen 85: Gráfica exactitud VIT con preentrenamiento	65
Imagen 86: Gráfica precisión VIT con preentrenamiento	66
Imagen 87: Gráfica Recall VIT con preentrenamiento	66
Imagen 88: Gráfica F1 Score VIT con preentrenamiento	66
Imagen 89: Gráfica pérdidas en épocas VIT con preentrenamiento	66
Imagen 90: Gráfica todas las métricas VIT con preentrenamiento	67
Imagen 91: Método POST para las predicciones de la IA	69
Imagen 92: Formato del cada Objeto en el archivo .JSON	70
Imagen 93: Vinculación de MongoDB con la API	71
Imagen 94: Servidor Express con la API	71
Imagen 95: Resultado de la API REST en línea	72

Imagen 96: Descarga de AvIAry en la web	73
Imagen 97: Aplicación e icono en el escritorio	74
Imagen 98: Descargar AvIAry en móviles	74
Imagen 99: AvIAry como una aplicación	74
Imagen 100: Prueba Unitaria	75
Imagen 101: Prueba de AvIAry en Safari	76
Imagen 102: Métricas de la aplicación	76
Imagen 103: Gráfica de prueba de estrés	77
Imagen 104: Gráfica de prueba de soak	77
Imagen 105: Gráfica de prueba de spike	78
Imagen 106: Resultados en Light House	78
Imagen 107: Diagrama de Gantt inicial	81
Imagen 108: Diagrama de Gantt final	83

TABLAS

Tabla 1: Tabla de requisitos	17
Tabla 2: Tabla del requisito RF-1	18
Tabla 3: Tabla del requisito RF-2	18
Tabla 4: Tabla del requisito RF-3	18
Tabla 5: Tabla del requisito RF-4	18
Tabla 6: Tabla del requisito RF-5	18
Tabla 7: Tabla del requisito RF-6	19
Tabla 8: Tabla del requisito RF-7	19
Tabla 9: Tabla del requisito RF-8	19
Tabla 10: Tabla del requisito RF-9	19
Tabla 11: Tabla del requisito RF-10	19
Tabla 12: Tabla del requisito RF-11	20
Tabla 13: Tabla del requisito RF-12	20
Tabla 14: Tabla del requisito RF-13	20
Tabla 15: Tabla del requisito RF-14	20
Tabla 16: Tabla del requisito RF-15	20
Tabla 17: Tabla del requisito RF-16	20
Tabla 18: Tabla del requisito RF-17	20
Tabla 19: Tabla de riesgos	21
Tabla 20: Tabla del riesgo RSK-1	21
Tabla 21: Tabla del riesgo RSK-2	21
Tabla 22: Tabla del riesgo RSK-3	22
Tabla 23: Tabla del riesgo RSK-4	22
Tabla 24: Tabla del riesgo RSK-5	22
Tabla 25: Tabla del riesgo RSK-6	22
Tabla 26: Tabla del riesgo RSK-7	23
Tabla 27: Tabla del riesgo RSK-8	23
Tabla 28: Resultados CNN simple	60
Tabla 29: Resultados CNN con ResNet18	62
Tabla 30: Tabla comparativa CNN	63
Tabla 31: Comparativa ViT	68

A) DESCRIPCIÓN DEL PROYECTO

1. INTRODUCCIÓN Y MOTIVACIÓN

En el mundo de la biodiversidad, las aves han fascinado a la humanidad a lo largo de la historia. Su plumaje colorido y vuelos intrigantes han sido una fuente de inspiración para poetas, artistas y observadores de aves. Sin embargo, reconocer las numerosas especies de aves en nuestro planeta puede ser un desafío monumental, especialmente para quienes no son expertos en ornitología. Es aquí donde surge la necesidad de una solución innovadora que utilice tecnología avanzada para acercar el mundo de las aves a todos.

En respuesta a esta necesidad, presentamos AvIAry, una aplicación web que combina la emoción de observar aves con la potencia de la inteligencia artificial. Esta aplicación no solo permite identificar aves a partir de imágenes, sino que también proporciona información detallada sobre las especies, creando un vínculo entre la curiosidad y el conocimiento profundo.

El ámbito en el que se focaliza nuestro proyecto es el de la divulgación científica y la apreciación de la naturaleza. Queremos acercar la riqueza de la biodiversidad aviar a todos, desde el ornitólogo aficionado hasta el simple amante de la naturaleza. Con nuestra aplicación, buscamos derribar las barreras que separan a las personas de la comprensión y aprecio de las aves, contribuyendo así a la conciencia ambiental y al respeto por la fauna.

La problemática que abordamos es la dificultad inherente a la identificación de aves, un desafío que puede resultar abrumador debido a la diversidad de especies, los matices en el plumaje y la falta de conocimientos especializados. Aquí es donde entra en juego nuestra aplicación, aprovechando el auge y el avance en los últimos años de la inteligencia artificial. Estas tecnologías superan las barreras tradicionales y proporcionan a los usuarios una herramienta precisa y accesible para la identificación de aves.

El propósito último de este proyecto es contribuir a la preservación de la biodiversidad, promoviendo la comprensión y conexión con las aves. Al dotar a las personas de herramientas que les permitan apreciar y comprender mejor a las aves, aspiramos a despertar un sentido de responsabilidad hacia la conservación del medio ambiente y las especies que lo habitan.

Este proyecto no sólo es una manifestación de todo lo que he aprendido durante mi estancia en la universidad en su máxima expresión, sino también un puente entre la tecnología y la biodiversidad, construido con la intención de inspirar y educar a aquellos que buscan explorar el mundo aviar que nos rodea cada día.

2. OBJETIVOS DEL PROYECTO

Estos objetivos profesionales y educativos están diseñados para abordar de manera integral los aspectos técnicos y educativos del proyecto, contribuyendo así a la misión más amplia de acercar el mundo de las aves a personas de todas las edades y niveles de conocimiento.

2.1 OBJETIVOS A NIVEL PROFESIONAL

Desarrollar y desplegar una aplicación web innovadora:

- ❖ Diseñar una interfaz de usuario intuitiva y atractiva que permita a los usuarios cargar imágenes de aves para su identificación.
- ❖ Implementar algoritmos de inteligencia artificial, como una red neuronal convolucional y Visual Transformer (ViT), para garantizar la precisión en la identificación de especies.
- ❖ Utilizar tecnologías modernas, como PyTorch y Python, para la implementación eficiente de los modelos de inteligencia artificial.

Implementar y optimizar modelos de inteligencia artificial:

- ❖ Crear dos modelos de inteligencia artificial incluyendo una red neuronal convolucional y un Visual Transformer (ViT).
- ❖ Realizar ajustes y optimizaciones en los modelos de inteligencia artificial para mejorar la velocidad y la precisión de la identificación de aves.
- ❖ Evaluar y comparar el rendimiento de diferentes variantes de los modelos, incluyendo opciones con y sin preentrenamiento, para determinar la configuración más eficaz.

Diseñar e implementar una base de datos NoSQL en MongoDB:

- ❖ Definir la estructura de la base de datos para almacenar información detallada sobre las especies de aves, incluyendo características físicas, hábitats y datos geográficos.
- ❖ Garantizar la escalabilidad de la base de datos para manejar el crecimiento futuro de la información y las especies registradas.

Configurar un servidor Flask eficiente:

- ❖ Implementar un servidor Flask que gestione las solicitudes de los usuarios y la comunicación efectiva con los modelos de inteligencia artificial.
- ❖ Optimizar la arquitectura del servidor para garantizar tiempos de respuesta rápidos y una experiencia fluida para el usuario.

Desplegar en la nube con Netlify y Render:

- ❖ Utilizar servicios en la nube como Netlify para alojar la interfaz de usuario y Render para desplegar el servidor Flask y la base de datos MongoDB.
- ❖ Asegurar la disponibilidad global de la aplicación y proporcionar escalabilidad para manejar un aumento en la carga de usuarios.

2.2 OBJETIVOS A NIVEL EDUCACIONAL

Facilitar el acceso y la comprensión de la biodiversidad aviar:

- ❖ Diseñar la aplicación con una interfaz intuitiva que permita a personas de todas las edades y niveles de experiencia explorar y aprender sobre las aves.
- ❖ Incorporar funcionalidades educativas, como descripciones detalladas, comportamientos y curiosidades sobre cada especie identificada.

Proporcionar una herramienta educativa integral:

- ❖ Integrar recursos multimedia, como imágenes de aves, para enriquecer la experiencia educativa de los usuarios.
- ❖ Ofrecer contenido educativo relacionado con la conservación de aves y su importancia en los ecosistemas.

Sensibilizar sobre la conservación de la biodiversidad:

- ❖ Incluir secciones educativas que destaquen los desafíos que enfrentan las aves en la actualidad, como la pérdida de hábitat y el cambio climático.
- ❖ Facilitar la participación de los usuarios en actividades relacionadas con el mundo aviar, como rutas para avistar nuevas especies o como construir un lugar donde un ave pueda vivir.

Inspirar el interés en la tecnología y la IA:

- ❖ Mostrar cómo la inteligencia artificial puede utilizarse de manera práctica para abordar problemas del mundo real, fomentando el interés en campos STEM.
- ❖ Proporcionar recursos educativos que guíen a los usuarios hacia una comprensión más profunda de los conceptos relacionados con la inteligencia artificial.

2.3 OBJETIVOS A NIVEL PERSONAL

Estos objetivos personales reflejan el deseo de crecimiento profesional, el compromiso con la causa ambiental y la búsqueda de una satisfacción personal a través de la contribución a un proyecto significativo.

Desarrollo de habilidades técnicas:

- ❖ Mejorar y ampliar mis habilidades en el diseño y desarrollo de aplicaciones web, especialmente aquellas que integran inteligencia artificial.
- ❖ Adquirir experiencia en la implementación y optimización de algoritmos de aprendizaje automático, explorando tanto redes neuronales convolucionales como Visual Transformer (ViT).

Profundización en la gestión de bases de datos NoSQL:

- ❖ Ampliar mis conocimientos en el diseño y gestión de bases de datos NoSQL, específicamente en MongoDB, explorando estrategias eficientes de almacenamiento y consulta de datos.

Aprendizaje en configuración de servidores y despliegue en la nube:

- ❖ Adquirir experiencia práctica en la configuración de servidores utilizando Flask, comprendiendo los principios de una comunicación eficiente entre la interfaz de usuario y la inteligencia artificial.
- ❖ Explorar el despliegue en la nube utilizando servicios como Netlify y Render, entendiendo los aspectos relacionados con la escalabilidad y la disponibilidad.

Desarrollo de perspectiva ambiental y de conservación:

- ❖ Profundizar mi comprensión sobre la importancia de la biodiversidad aviar y su impacto en los ecosistemas, desarrollando una perspectiva más amplia sobre la conservación del medio ambiente.

Inspiración y motivación personal:

- ❖ Encontrar satisfacción y motivación personal al contribuir a un proyecto que combina mi pasión por la tecnología con el amor por la naturaleza.
- ❖ Inspirar a otros y compartir el entusiasmo por la exploración y comprensión de la biodiversidad aviar, generando un impacto positivo en la conexión entre las personas y el mundo natural.

3. ESTADO DEL ARTE

En el universo digital de la ornitología, diversas aplicaciones compiten por acercar a los entusiastas de las aves a la riqueza de la biodiversidad alada. En este análisis, exploraremos cinco destacadas aplicaciones, evaluando sus características, ventajas y desventajas. Este panorama detallado contextualizará nuestro proyecto, que busca fusionar identificación instantánea, educación y conciencia ambiental. A través de esta comparativa, destacaremos las fortalezas únicas que distinguen a nuestra aplicación en este vibrante campo.

3.1 MERLIN BIRD ID (CORNELL LAB)

Merlin Bird ID, desarrollada por el Cornell Lab of Ornithology, destaca como una aplicación móvil que capitaliza la colaboración de la comunidad ornitológica. Su funcionalidad principal reside en la identificación precisa de aves a través de imágenes, aprovechando algoritmos avanzados respaldados por una extensa base de datos. La colaboración comunitaria en proyectos de investigación es un rasgo distintivo, permitiendo a los usuarios contribuir activamente y participar en la expansión del conocimiento ornitológico.

Pros:

- ❖ **Colaboración comunitaria:** Merlin Bird ID fomenta la participación de la comunidad ornitológica, permitiendo contribuciones valiosas a proyectos de investigación.
- ❖ **Interfaz amigable:** Su interfaz intuitiva facilita el uso, siendo accesible para usuarios con diversos niveles de experiencia.

Contras:

- ❖ **Limitaciones educativas:** Aunque proporciona información básica sobre las especies, carece de detalles educativos más extensos.
- ❖ **Enfoque en identificación:** Su enfoque primario en la identificación podría limitar la profundidad educativa y de conciencia ambiental.

Comparación con nuestro proyecto:

- ❖ **Colaboración y educación:** Mientras Merlin Bird ID destaca en la colaboración comunitaria, nuestro proyecto busca fusionar la identificación precisa con una experiencia educativa y de conciencia ambiental más rica. Nuestra aplicación aspira a no solo identificar aves, sino también inspirar el aprendizaje profundo y la apreciación de la biodiversidad aviar.

3.2 eBIRD (CORNELL LAB Y AUDUBON)

eBird, una colaboración entre el Cornell Lab of Ornithology y la National Audubon Society, se destaca como una plataforma en línea dedicada al registro y compartir de observaciones de aves. Su enfoque se centra en la colaboración comunitaria y la construcción de una base de datos exhaustiva sobre la distribución y abundancia de especies aviares.

Pros:

- ❖ **Colaboración y comunidad:** eBird facilita la colaboración entre observadores de aves, permitiendo contribuciones a proyectos científicos y la construcción de una base de datos rica.
- ❖ **Datos detallados:** Ofrece información detallada sobre la distribución y abundancia de las especies.

Contras:

- ❖ **Falta de identificación visual instantánea:** La aplicación no proporciona una función de identificación visual instantánea, lo que puede limitar su utilidad para usuarios que buscan respuestas inmediatas.
- ❖ **Complejidad técnica:** Puede resultar más técnica y menos accesible para principiantes en comparación con aplicaciones de identificación visual.

Comparación con nuestro proyecto:

- ❖ **Identificación instantánea y educación:** A diferencia de eBird, nuestro proyecto se enfoca en la identificación instantánea a través de algoritmos de inteligencia artificial y busca integrar elementos educativos y de conciencia ambiental. Nuestra aplicación aspira a brindar una experiencia más completa para los amantes de las aves y la naturaleza.

3.3 iBIRD

iBird se destaca como una aplicación móvil enfocada en la identificación de aves mediante imágenes y sonidos. Su plataforma busca ofrecer una experiencia de usuario detallada, permitiendo búsquedas por atributos y proporcionando una amplia base de datos de aves con imágenes y grabaciones de sonido.

Pros:

- ❖ **Búsqueda por atributos:** Permite a los usuarios buscar aves utilizando una variedad de atributos como color, tamaño y ubicación.
- ❖ **Base de datos amplia:** Contiene una extensa base de datos de aves con múltiples imágenes y grabaciones de sonido.

Contras:

- ❖ **Interfaz compleja para algunos usuarios:** La interfaz puede resultar compleja para algunos usuarios, especialmente aquellos que buscan una experiencia más simple.
- ❖ **Limitaciones en colaboración comunitaria:** No se centra en la colaboración de la comunidad ni en proyectos de investigación.

Comparación con nuestro proyecto:

- ❖ **Enfoque educativo y de colaboración:** A diferencia de iBird, nuestro proyecto no solo se centra en la identificación, sino que también integra aspectos educativos y de conciencia ambiental, además de fomentar la colaboración comunitaria. Nuestra aplicación aspira a proporcionar una experiencia más completa y educativa para los usuarios.

3.4 BIRDNET

BirdNET destaca como una aplicación que utiliza inteligencia artificial para la identificación de aves mediante el reconocimiento de cantos. Su diseño está orientado a la identificación basada en vocalizaciones, siendo capaz de funcionar en tiempo real para identificar aves por sus sonidos.

Pros:

- ❖ **Identificación por sonido:** Ofrece una alternativa para la identificación basada en vocalizaciones, especialmente útil en entornos donde la identificación visual es difícil.
- ❖ **Enfoque en especies singulares:** Puede destacar especies menos comunes que son conocidas por sus llamadas distintivas.

Contras:

- ❖ **Limitado a sonidos:** Se enfoca principalmente en la identificación por sonido, dejando de lado la identificación visual.
- ❖ **Menos información educativa:** Carece de información educativa extensa sobre las especies identificadas.

Comparación con nuestro proyecto:

- ❖ **Enfoque integral:** A diferencia de BirdNET, nuestro proyecto aborda tanto la identificación visual como la educación extensa sobre las especies de aves. Buscamos proporcionar una experiencia completa que integre tanto la identificación como la educación para una comprensión más profunda de la biodiversidad aviar.

3.5 AUDUBON BIRD GUIDE

La Audubon Bird Guide es una aplicación desarrollada por la National Audubon Society y se destaca como una herramienta integral para la identificación y exploración de aves. Su enfoque abarca desde la identificación mediante imágenes hasta la información detallada sobre hábitats, comportamientos y migraciones.

Pros:

- ❖ **Identificación visual completa:** Ofrece una sólida función de identificación visual mediante imágenes de alta calidad y descripciones detalladas.
- ❖ **Amplia base de datos:** Contiene información exhaustiva sobre una amplia variedad de especies, proporcionando detalles sobre su biología y conservación.

Contras:

- ❖ **Interfaz puede ser abrumadora:** La abundancia de información podría resultar abrumadora para algunos usuarios, especialmente aquellos que buscan respuestas rápidas.
- ❖ **Menos enfocada en colaboración:** Aunque permite la contribución de avistamientos, no se centra tanto en la colaboración comunitaria como algunas otras aplicaciones.

Comparación con nuestro proyecto:

- ❖ **Enfoque integral:** Similar a nuestra aplicación, la Audubon Bird Guide busca proporcionar una experiencia completa que incluye identificación, información educativa y conservacionista. Sin embargo, nuestra aplicación se diferencia al centrarse en la identificación instantánea a través de algoritmos de inteligencia artificial.

Como podemos observar, cada competidor aporta su propia perspectiva y enfoque, pero nuestro proyecto se distingue al buscar una fusión integral de identificación, educación y conciencia ambiental para brindar una experiencia única y completa a los amantes de las aves y la naturaleza.

4. ELICITACIÓN DE REQUISITOS

En esta sección crucial del proyecto, nos sumergimos en la tarea de definir claramente los requisitos funcionales y no funcionales de nuestra aplicación, que no solo busca identificar aves, sino también ofrecer una experiencia completa y enriquecedora a los usuarios. Además, exploramos los posibles riesgos que podrían surgir durante el desarrollo y la implementación de la aplicación, anticipando desafíos y estableciendo estrategias para mitigarlos.

A través de tablas organizadas, abordaremos requisitos funcionales y no funcionales clave, detallando su importancia y proporcionando una base estructurada para el desarrollo. Asimismo, identificamos riesgos potenciales, evaluamos su probabilidad y describimos las medidas preventivas y correctivas que implementaremos para garantizar la estabilidad y el éxito continuo de nuestra aplicación. Este proceso es esencial para sentar las bases sólidas que guiarán el desarrollo, asegurando que nuestra aplicación cumpla con las expectativas y supere los posibles obstáculos en su camino desde su diseño hasta su finalización completa.

4.1 REQUISITOS

Cada uno de los requisitos vienen representados en una tabla, teniendo dicha tabla el siguiente formato:

Identificador	Tipo	Nombre	Descripción	Prioridad

Tabla 1: Tabla de requisitos

- ❖ **Identificador:** Es un código corto que se emplea para saber de forma rápida de que requisito estamos hablando. Tiene la siguiente estructura; **RF-x**, siendo **RF**

Requisito Funcional y x un número que se incrementa a cada nuevo requisito o **RNF-x** funcionando de la misma forma, pero para Requisitos No Funcionales.

- ❖ **Tipo:** Detalla si es funcional o no de forma más explícita.
- ❖ **Nombre:** Da vida de forma resumida al requisito.
- ❖ **Descripción:** Ofrece información más detallada sobre el requisito en cuestión.
- ❖ **Prioridad:** Establece un nivel de prioridad al requisito, pudiendo esta ser **Baja, Media o Alta**.

Identificador	Tipo	Nombre	Descripción	Prioridad
RF-1	Funcional	Identificación de aves por imágenes	La aplicación debe ser capaz de identificar aves a partir de imágenes proporcionadas por los usuarios.	Alta

Tabla 2: Tabla del requisito RF-1

Identificador	Tipo	Nombre	Descripción	Prioridad
RF-2	Funcional	Interfaz de usuario (UI) intuitiva	La aplicación debe contar con una interfaz de usuario intuitiva y fácil de usar para garantizar una experiencia positiva para usuarios de todos los niveles de habilidad.	Alta

Tabla 3: Tabla del requisito RF-2

Identificador	Tipo	Nombre	Descripción	Prioridad
RF-3	Funcional	Compatibilidad Multiplataforma	La aplicación debe ser compatible con varios dispositivos y sistemas operativos para llegar a un público más amplio.	Medio

Tabla 4: Tabla del requisito RF-3

Identificador	Tipo	Nombre	Descripción	Prioridad
RF-4	Funcional	Base de datos actualizable	La base de datos de especies de aves debe ser actualizable para incluir nuevas especies y mantenerse al día con la evolución de la información ornitológica.	Alta

Tabla 5: Tabla del requisito RF-4

Identificador	Tipo	Nombre	Descripción	Prioridad
RF-5	Funcional	Información de calidad sobre cada especie de ave	La aplicación debe proporcionar información verídica sobre los atributos de cada especie de ave.	Alta

Tabla 6: Tabla del requisito RF-5

Identificador	Tipo	Nombre	Descripción	Prioridad
RF-6	Funcional	PDF con el resultado obtenido	La aplicación debe ser capaz de mostrar un PDF donde se refleje un informe con el resultado obtenido.	Baja

Tabla 7: Tabla del requisito RF-6

Identificador	Tipo	Nombre	Descripción	Prioridad
RF-7	Funcional	Buscador de especies	La aplicación debe contar con una barra de búsqueda para poder buscar cualquier especie y conocer sus datos	Alta

Tabla 8: Tabla del requisito RF-7

Identificador	Tipo	Nombre	Descripción	Prioridad
RF-8	Funcional	Aproximación de precisión	La aplicación debe mostrar un índice de confianza tras dar un resultado al usuario.	Alta

Tabla 9: Tabla del requisito RF-8

Identificador	Tipo	Nombre	Descripción	Prioridad
RF-9	Funcional	Mostrar abanico de resultados	La aplicación debe ser capaz de además de mostrar un índice de confianza, que pueda dar la información de las cinco aves que más se aproximan al resultado.	Media

Tabla 10: Tabla del requisito RF-9

Identificador	Tipo	Nombre	Descripción	Prioridad
RNF-10	No Funcional	Eficiencia de carga	La aplicación debe tener tiempos de carga eficientes para garantizar una experiencia de usuario rápida y sin inconvenientes.	Alta

Tabla 11: Tabla del requisito RF-10

Identificador	Tipo	Nombre	Descripción	Prioridad
RNF-11	No Funcional	Eficiencia de carga	La aplicación debe tener tiempos de carga eficientes para garantizar una experiencia de usuario rápida y sin inconvenientes.	Alta

Tabla 12: Tabla del requisito RF-11

Identificador	Tipo	Nombre	Descripción	Prioridad
RNF-12	No Funcional	Seguridad de datos mostrados	La información, incluidas las imágenes cargadas, debe almacenarse y gestionarse de forma segura y privada.	Alta

Tabla 13: Tabla del requisito RF-12

Identificador	Tipo	Nombre	Descripción	Prioridad
RNF-13	No Funcional	Escalabilidad del sistema	La aplicación debe ser escalable para manejar un aumento sustancial de usuarios y datos a medida que crece la comunidad.	Alta

Tabla 14: Tabla del requisito RF-13

Identificador	Tipo	Nombre	Descripción	Prioridad
RNF-14	No Funcional	Accesibilidad	La aplicación debe cumplir con estándares de accesibilidad para garantizar que sea utilizable por personas con discapacidades.	Alta

Tabla 15: Tabla del requisito RF-14

Identificador	Tipo	Nombre	Descripción	Prioridad
RNF-15	No Funcional	Eficiencia en consumo de batería	La aplicación debe optimizar el consumo de batería para garantizar una duración adecuada en dispositivos móviles.	Media

Tabla 16: Tabla del requisito RF-15

Identificador	Tipo	Nombre	Descripción	Prioridad
RNF-16	No Funcional	Tiempo de respuesta del sistema	La aplicación debe ofrecer tiempos de respuesta rápidos (dentro de lo posible) para la identificación y consulta de información.	Alta

Tabla 17: Tabla del requisito RF-16

Identificador	Tipo	Nombre	Descripción	Prioridad
RNF-17	No Funcional	Compatibilidad con navegadores	La aplicación debe ser compatible con una variedad de navegadores web para garantizar la accesibilidad a través de diferentes plataformas.	Media

Tabla 18: Tabla del requisito RF-17

4.2 RIESGOS

Cada uno de los riesgos vienen representados en una tabla, teniendo dicha tabla el siguiente formato:

Identificador	Nombre del riesgo	Probabilidad	Descripción	Mitigación del riesgo

Tabla 19: Tabla de riesgos

- ❖ **Identificador:** Es un código corto que se emplea para saber de forma rápida de que riesgo estamos hablando. Tiene la siguiente estructura; **RSK-x**, siendo x un número que se incrementa a cada nuevo riesgo.
- ❖ **Probabilidad / Impacto:** Que tan probable es de que ocurra el riesgo, puede ser Baja, Media y Alta. Además, el Impacto mide cuanto dañaría ese riesgo al proyecto.
- ❖ **Nombre:** Da vida de forma resumida al riesgo.
- ❖ **Descripción:** Ofrece información más detallada sobre el riesgo en cuestión.
- ❖ **Mitigación del riesgo:** Medida preventiva para evitar que el riesgo ocurra o para solucionarlo en el caso de que ocurra.

Identificador	Nombre del riesgo	Probabilidad / Impacto	Descripción	Mitigación del riesgo
RSK-1	Resolución baja de imágenes	Media / Medio	Las imágenes de baja resolución pueden afectar la precisión de la identificación.	Utilizar algoritmos de procesamiento de imágenes avanzados y establecer requisitos mínimos de resolución para las imágenes cargadas.

Tabla 20: Tabla del riesgo RSK-1

Identificador	Nombre del riesgo	Probabilidad / Impacto	Descripción	Mitigación del riesgo
RSK-2	Fallos en el reconocimiento de imágenes	Alta / Alto	Pueden ocurrir fallos en la identificación de aves debido a imágenes de baja calidad o patrones de plumaje inusuales.	Implementar algoritmos robustos de procesamiento de imágenes y realizar pruebas extensivas con una variedad de imágenes.

Tabla 21: Tabla del riesgo RSK-2

Identificador	Nombre del riesgo	Probabilidad / Impacto	Descripción	Mitigación del riesgo
RSK-3	Actualizaciones de sistema operativo	Baja / Bajo	Actualizaciones de los sistemas operativos pueden afectar la compatibilidad de la aplicación.	Mantener un equipo de desarrollo activo para realizar adaptaciones rápidas a nuevas actualizaciones de sistemas operativos.

Tabla 22: Tabla del riesgo RSK-3

Identificador	Nombre del riesgo	Probabilidad / Impacto	Descripción	Mitigación del riesgo
RSK-4	Carga masiva de datos	Media / Alto	La carga simultánea de grandes cantidades de datos puede afectar el rendimiento del sistema.	Implementar límites de carga y optimizar la gestión de datos para manejar volúmenes masivos de manera eficiente.

Tabla 23: Tabla del riesgo RSK-4

Identificador	Nombre del riesgo	Probabilidad / Impacto	Descripción	Mitigación del riesgo
RSK-5	Dependencia de servicios en la nube	Media / Alto	Problemas en los servicios en la nube utilizados para la aplicación pueden afectar la disponibilidad y funcionalidad.	Implementar estrategias de respaldo y tener planes de contingencia para cambiar rápidamente a servicios alternativos si es necesario.

Tabla 24: Tabla del riesgo RSK-5

Identificador	Nombre del riesgo	Probabilidad / Impacto	Descripción	Mitigación del riesgo
RSK-6	Dificultades en la actualización de la base de datos	Media / Medio	Problemas técnicos o logísticos que podrían surgir durante la actualización de la base de datos de especies de aves.	Realizar pruebas exhaustivas antes de las actualizaciones y tener un proceso de reversión efectivo en caso de problemas.

Tabla 25: Tabla del riesgo RSK-6

Identificador	Nombre del riesgo	Probabilidad / Impacto	Descripción	Mitigación del riesgo
RSK-7	Problemas de Cache en	Media	Posibles desafíos en la gestión de caché	Realizar pruebas exhaustivas en entornos offline para

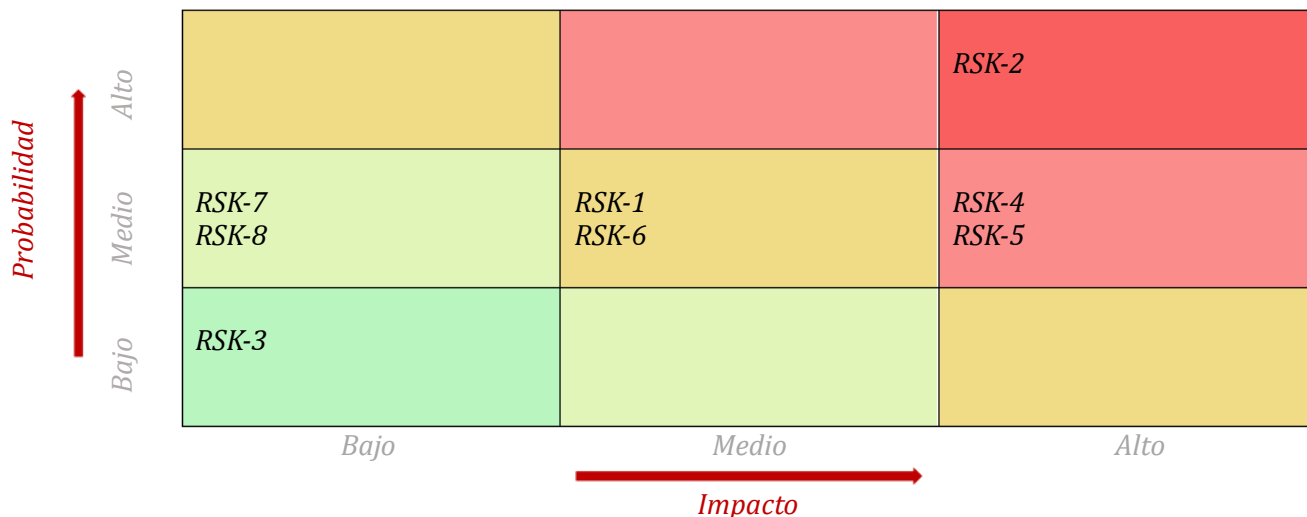
	Modo Offline	/ Bajo	cuando la aplicación opera en modo offline.	garantizar la funcionalidad adecuada de la caché.
--	--------------	-----------	---	---

Tabla 26: Tabla del riesgo RSK-7

Identificador	Nombre del riesgo	Probabilidad / Impacto	Descripción	Mitigación del riesgo
RSK-8	Cambios en Estándares Web	Media / Bajo	Cambios en los estándares web pueden afectar la compatibilidad de la aplicación con navegadores específicos.	Mantenerse informado sobre los cambios en los estándares web y adaptar la aplicación según sea necesario.

Tabla 27: Tabla del riesgo RSK-8

Una vez vistos todos los riesgos, podemos agruparlos en una matriz de riesgos gracias a sus valores de impacto y probabilidad para ver qué tan arriesgado sería la producción de nuestra aplicación.



B) EJECUCIÓN DEL PROYECTO

5. DISEÑO Y ARQUITECTURA DEL SISTEMA Y TECNOLOGÍAS EMPLEADAS EN ÉL

El diseño integral de nuestro sistema se estructura en tres componentes fundamentales: el Frontend (Parte visual de la aplicación), el servidor de IA y el Backend (parte donde ocurre la lógica y se almacenan los datos de la aplicación). Cada uno desempeña un papel crucial, trabajando en sinergia para brindar una experiencia excepcional al usuario interesado en la identificación de aves. Vamos a adentrarnos en cada componente con un enfoque detallado y extenso, destacando las razones detrás de las elecciones tecnológicas y proporcionando una visión más profunda para principiantes.

5.1 FRONT-END: INTERACTIVIDAD Y ESTILO UNIFICADOS

El Front-end es la cara visible de nuestra aplicación, donde el usuario interactúa directamente. La elección de utilizar React, una biblioteca de JavaScript desarrollada por Facebook es estratégica. React simplifica el desarrollo de interfaces de usuario interactivas al descomponer la interfaz en componentes reutilizables. Este enfoque modular facilita la mantenibilidad del código y fomenta la escalabilidad del sistema a medida que nuevas características se agregan. Además, Bootstrap, un marco de diseño CSS, se integra para gestionar los estilos. Este enfoque garantiza una apariencia moderna y una experiencia de usuario consistente en una variedad de dispositivos.

Para principiantes: React es una biblioteca de JavaScript utilizada para construir interfaces de usuario reactivas. Bootstrap es un marco de diseño que proporciona estilos predefinidos y componentes para facilitar el diseño de interfaces de usuario atractivas.

5.2 SERVIDOR DE IA: PODER DE APRENDIZAJE PROFUNDO

El corazón de nuestro sistema reside en el Servidor de IA, impulsado por una inteligencia artificial desarrollada en PyTorch. PyTorch es una biblioteca de aprendizaje profundo ampliamente utilizada, conocida por su flexibilidad y facilidad de uso. La elección de una GPU Nvidia RTX 3050Ti para el entrenamiento del método de IA proporciona un impulso significativo en el rendimiento, acelerando el proceso de aprendizaje. Flask, un marco web ligero para Python, se elige para la implementación del servidor debido a su simplicidad y eficiencia.

Para principiantes: PyTorch es una biblioteca de aprendizaje profundo para Python. Flask es un marco web que facilita la creación de servidores web en Python.

5.3 BACK-END: POTENCIA NOSQL DE MONGODB

El Back-end actúa como la columna vertebral de nuestra aplicación, almacenando y gestionando los datos. Se ha optado por una base de datos NoSQL, MongoDB, y se utiliza MongoDB Atlas y MongoDB Compass para la administración de datos. MongoDB, una base de datos orientada a documentos, es preferida en entornos donde la estructura de datos puede variar y expandirse con el tiempo. A diferencia de las bases de datos SQL, que son estructuradas y requieren un esquema fijo, MongoDB permite una flexibilidad significativa y es altamente escalable.

Para principiantes: MongoDB es una base de datos NoSQL que almacena datos en formato de documentos JSON. MongoDB Atlas es un servicio de base de datos en la nube y MongoDB Compass es una interfaz gráfica de usuario para gestionar bases de datos MongoDB.

5.4 FLUJO DETALLADO DEL PROCESO: UN BALLET TECNOLÓGICO

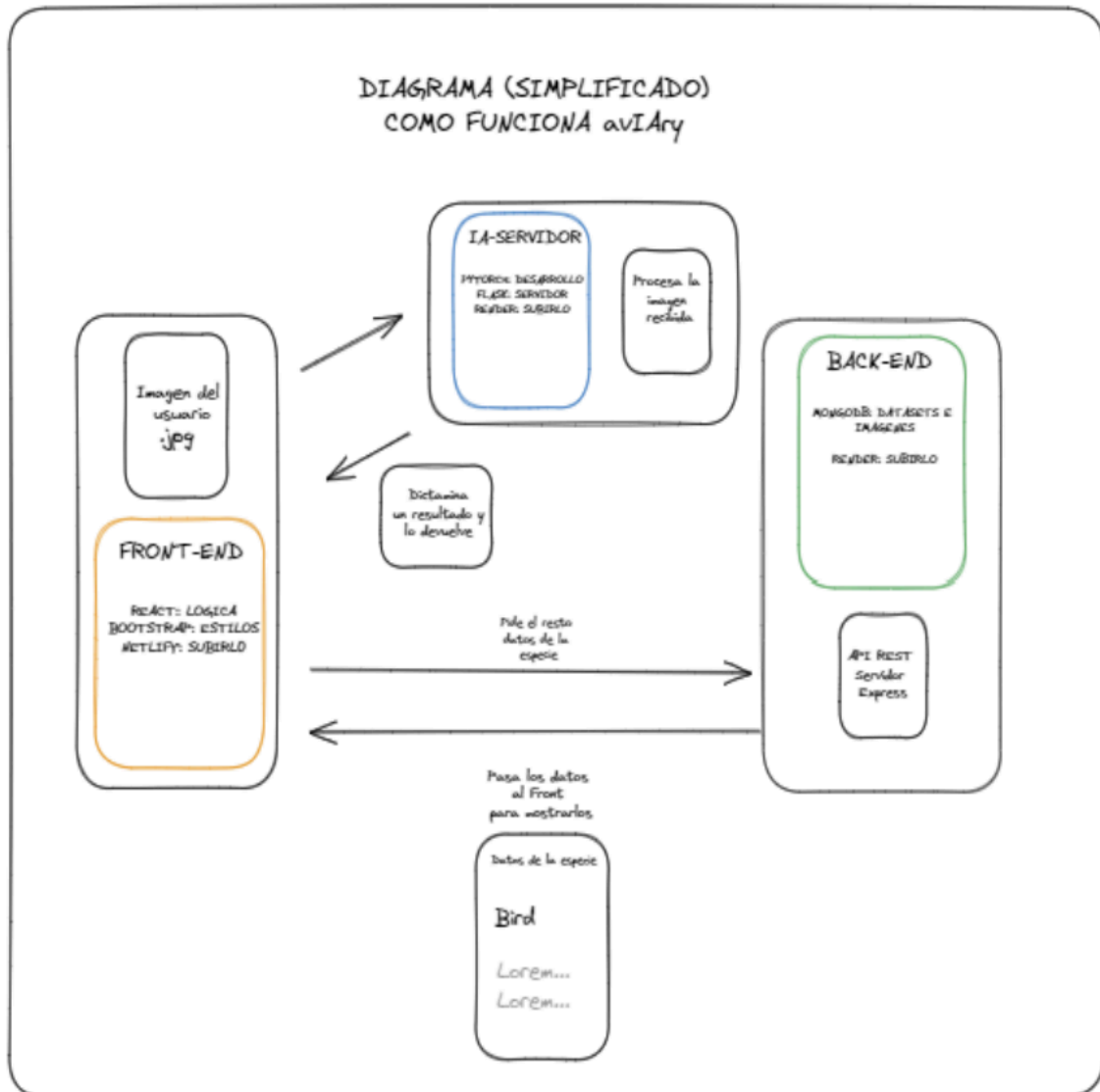


Imagen 1: Flujo de la aplicación

- ❖ El usuario carga una imagen en el Front-end.
- ❖ El Front-end utiliza React para gestionar la interfaz interactiva y Bootstrap para estilos modernos.
- ❖ La imagen se envía al Servidor de IA, construido con Flask y alimentado por la potencia de PyTorch.
- ❖ El método de IA procesa la imagen y devuelve una predicción al Front-end.
- ❖ El Front-end, ahora con una predicción, envía una solicitud al Back-end para obtener datos adicionales sobre la especie predicha.
- ❖ El Back-end, basado en MongoDB, consulta y recupera datos de la especie mediante una API REST.

- ❖ La información se devuelve al Front-end, que presenta al usuario la especie identificada y sus datos.

5.5 RAZONES DETRÁS DE LAS ELECCIONES TECNOLÓGICAS

La elección de **React** como marco de desarrollo para el Front-end se basa en varias consideraciones estratégicas.

- ❖ **Popularidad y comunidad activa:** React es ampliamente adoptado y cuenta con una comunidad activa. Esta popularidad se traduce en una abundancia de recursos, bibliotecas y soluciones a desafíos comunes, facilitando el desarrollo y la resolución de problemas.
- ❖ **Arquitectura basada en componentes:** La arquitectura modular basada en componentes de React permite la creación de interfaces de usuario reutilizables y mantenibles. Esto es esencial para un proyecto de este tipo, donde la escalabilidad y la eficiencia del desarrollo son primordiales.
- ❖ **Virtual DOM para eficiencia:** El uso del Virtual DOM de React optimiza las actualizaciones de la interfaz de usuario, mejorando la eficiencia en comparación con otras tecnologías. Esta característica es crucial para proporcionar una experiencia de usuario fluida, especialmente al cargar y actualizar imágenes.

¿Por qué no otras bibliotecas populares como Vue o Angular?

Si bien Vue y Angular son opciones sólidas, React se destacó por su equilibrio entre flexibilidad, rendimiento y una curva de aprendizaje relativamente suave. Además, la popularidad y la amplia adopción de React facilitan la colaboración y la resolución de problemas.

La elección de **PyTorch** para el desarrollo de la inteligencia artificial se basa en las siguientes consideraciones.

- ❖ **Flexibilidad y dinamismo:** PyTorch se destaca por su estructura dinámica y flexible, lo que facilita la experimentación y la iteración durante el desarrollo del modelo de aprendizaje profundo. Esto es especialmente relevante cuando se exploran diferentes arquitecturas y enfoques.
- ❖ **Comunidad activa y documentación clara:** Cuenta con una comunidad activa y una documentación clara. Esto facilita la resolución de problemas y la adopción rápida de nuevas técnicas y enfoques en el campo del aprendizaje profundo.

¿Por qué no apoyarse de otros como TensorFlow y Keras?

Si bien TensorFlow y Keras son opciones brillantes y ampliamente utilizadas, PyTorch se ha destacado por gran flexibilidad y polivalencia, simplificando el desarrollo y la experimentación en el contexto específico de este proyecto.

La elección de **MongoDB** y la tecnología **NoSQL** se basa en la flexibilidad y escalabilidad inherentes a este enfoque.

- ❖ **Esquema dinámico de NoSQL:** MongoDB, como base de datos NoSQL, permite un esquema dinámico. Esto es esencial en un proyecto donde la estructura de los datos puede evolucionar con el tiempo, proporcionando flexibilidad sin comprometer el rendimiento.
- ❖ **Escalabilidad horizontal:** MongoDB es conocido por su capacidad de escalabilidad horizontal, permitiendo manejar grandes volúmenes de datos y proporcionando un rendimiento constante incluso a medida que la aplicación crece.

¿Por qué no usar SQL tradicional?

Aunque las bases de datos SQL ofrecen relaciones estructuradas y soporte transaccional, su rigidez en el esquema y la escalabilidad vertical hacen que NoSQL sea una opción más adecuada para proyectos que priorizan la flexibilidad y la capacidad de escalar horizontalmente.

Este diseño, donde cada componente desempeña un papel específico, proporciona una base sólida para la identificación de aves. La selección de tecnologías se basa en la eficiencia, la escalabilidad y la facilidad de mantenimiento, garantizando un sistema robusto y preparado para futuros cambios.

5.6 CONCLUSIÓN: UNA SINFONÍA TECNOLÓGICA EQUILIBRADA

La elección de tecnologías en cada componente del sistema se basa en un equilibrio cuidadoso entre eficiencia, escalabilidad y facilidad de desarrollo. **React** para el Front-end, **MongoDB** para el Back-end y **PyTorch** para el método de IA forman una sinfonía tecnológica que potencia la identificación de aves con fluidez, flexibilidad y potencia de aprendizaje profundo. Estas decisiones, respaldadas por la solidez y la popularidad de cada tecnología, proporcionan una base robusta para nuestro proyecto.

6. IMPLEMENTACIÓN Y DESARROLLO

El desarrollo de nuestra aplicación de identificación de aves implica una cuidadosa implementación de cada componente: el Front-end, el servidor de IA y el Back-end. A continuación, exploraremos en detalle cómo se ha llevado a cabo cada parte del proceso, desde la concepción hasta la integración armoniosa de todas las piezas.

6.1 FRONT-END: DISEÑO INTERACTIVO CON REACT Y BOOTSTRAP

Para realizar la parte visible de la aplicación debemos comenzar por el diseño de la aplicación. Puesto que estamos usando React, la aplicación se divide en componentes por lo que el primer paso a realizar es saber en cuantos componentes vamos a dividir nuestra interfaz de usuario y que va a contener cada componente. En este caso vamos a dividir la aplicación en 12 componentes principales, por lo que la necesitamos hacer un boceto o mock-up de cada componente para poder plasmar las ideas en algún lugar.

En este caso, al igual que con el diagrama del funcionamiento de la aplicación, emplearemos la herramienta Excalidraw, una estupenda herramienta en línea que nos permite hacer los bocetos de forma simple rápida sin tener que entrar en detalles en el proceso de creación.

6.1.1 COMPONENTES QUE FORMAN EL FRONT-END

❖ Componente 'Inicio'

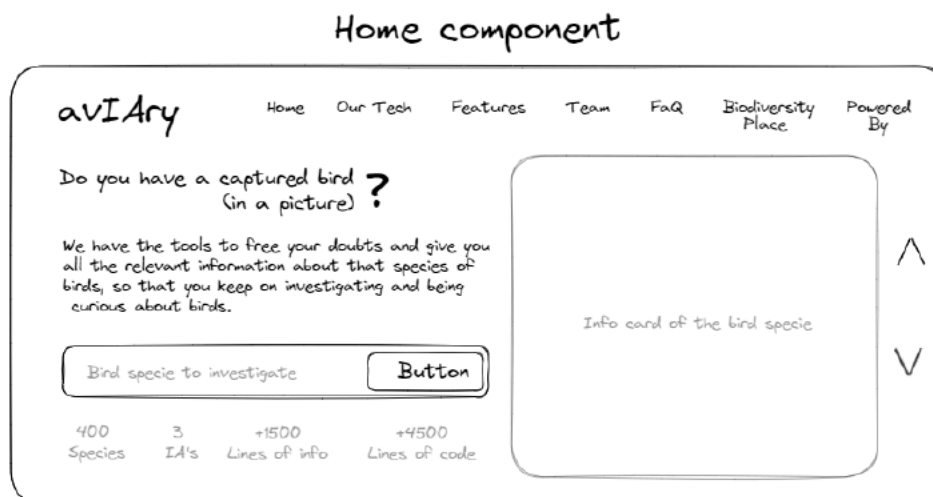


Imagen 2: Mock up componente Inicio

Este componente es uno de los dos más importantes de nuestra aplicación ya que incluye la mayor parte de los elementos clave de nuestra aplicación, siendo estos:

- ❖ **Barra de navegación:** En ella podemos encontrar un menú donde podemos desplazarnos a cualquier parte de nuestra aplicación de forma rápida y sencilla. Cuenta con siete diferentes opciones que nos desplazarán a los componentes que iremos detallando a continuación.
- ❖ **Barra de búsqueda:** Es nuestra principal herramienta para buscar por su nombre a cualquier ave de la que queramos obtener información al respecto. Es sencilla de usar y en ella empleamos todo el poder de React ya que funciona a tiempo real, es decir no necesitas pulsar un botón para dar la orden, sino que a medida que vas escribiendo automáticamente el sistema busca la especie con mayor índice de similitud.
- ❖ **Carta de la especie:** Es el lugar donde se mostrará toda la información de la especie que estemos buscando mediante la barra de búsqueda o donde aparecerá la especie resultado dada por la IA cuando usemos el buscador mediante imágenes.
- ❖ **Descarga PDF con datos de la especie:** También cuando aparezca el resultado de la IA, podremos descargar un PDF con los datos de dicha especie, así como de las predicciones de la IA favoreciendo la comodidad de los usuarios.

El funcionamiento es bastante innovador ya que cuenta con un pequeño efecto 3D cuando el ratón se arrastra por encima de la carta. En esta carta se muestra la imagen de la especie en cuestión e información detallada de la especie como el hábitat natural, el color más común, la esperanza de vida, su alimentación, su tamaño y su peso promedios. Además de diferentes botones a su alrededor para que el usuario pueda ver cómodamente el abanico de especies disponibles.

Con todo ello, tras tener plasmado el boceto de cómo sería el componente de inicio sólo queda ponerse manos a la obra para hacer que todo aquello que hemos detallado en el boceto esté en la versión final de este componente.

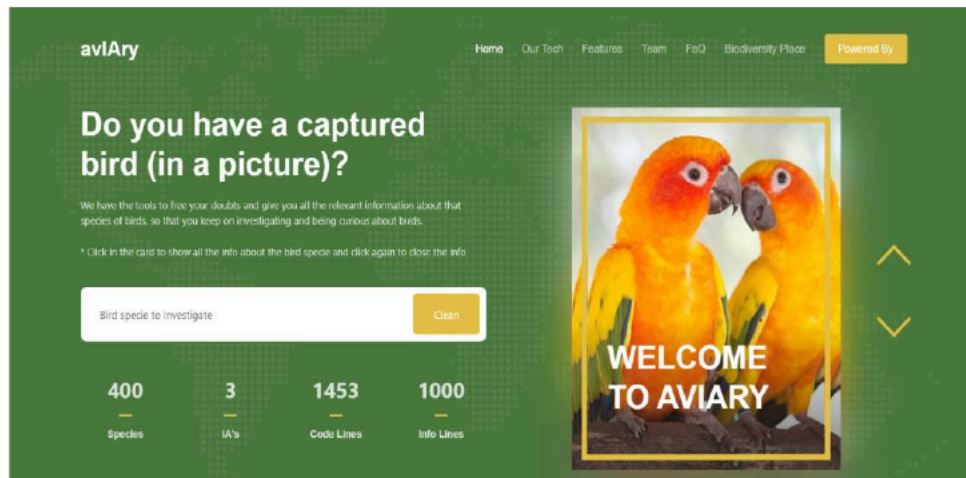


Imagen 3: Componente Inicio final

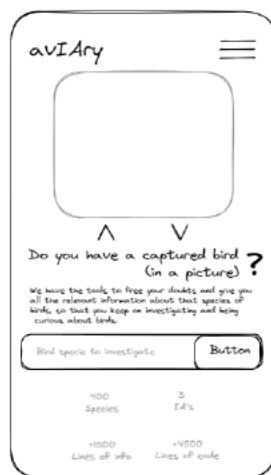


Imagen 4: Mock up componente Inicio (móvil)

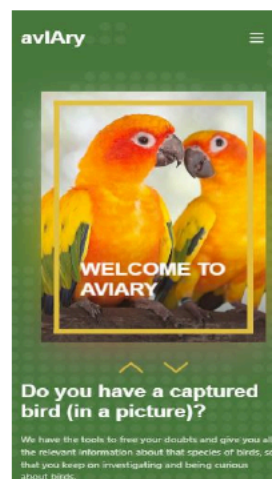


Imagen 5: Componente Inicio final (móvil)

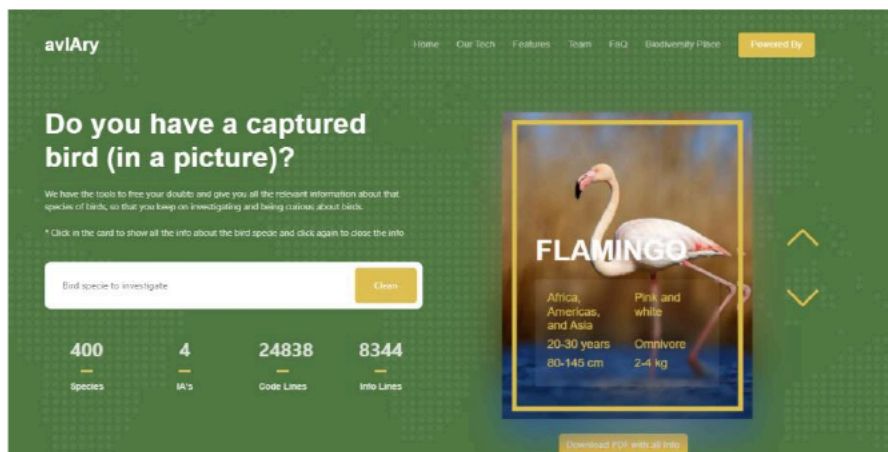


Imagen 6: Componente Inicio con predicción

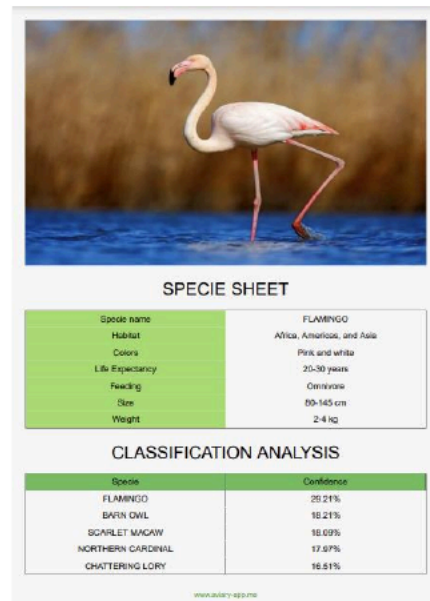


Imagen 7: PDF con los datos obtenidos

❖ Componente 'Nuestra Tecnología'

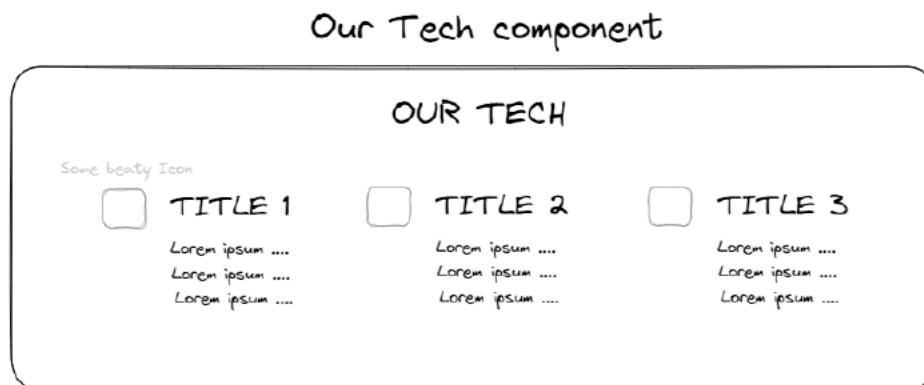


Imagen 8: Mock up componente Nuestra tecnología

Este componente tiene como objetivo mostrarle al público de forma muy sencilla, qué tecnologías usamos en AvlAry para hacer posible la magia de ofrecer un resultado, en este caso una especie, cuando nos llega una imagen.

Podemos ver que este se divide en tres partes, tratando la primera de ellas de explicar qué es una red neuronal y cómo funciona de forma muy simplificada para que cualquier persona que no tenga conocimientos sobre Deep Learning sepa a grandes rasgos qué función tiene. La segunda parte intenta explicar qué es el preentrenamiento y cómo afecta este al funcionamiento y al comportamiento de estas redes neuronales. Finalmente, la tercera y última parte trata sobre la estructura Transformer y cómo su potencial ha revolucionado el mundo de la inteligencia artificial desde su aparición.

Todos estos conceptos serán altamente detallados en el apartado 6.2 Servidor IA: La creación de una nueva forma de pensar, donde se detalla el proceso de creación de los modelos detrás de AvIAry.

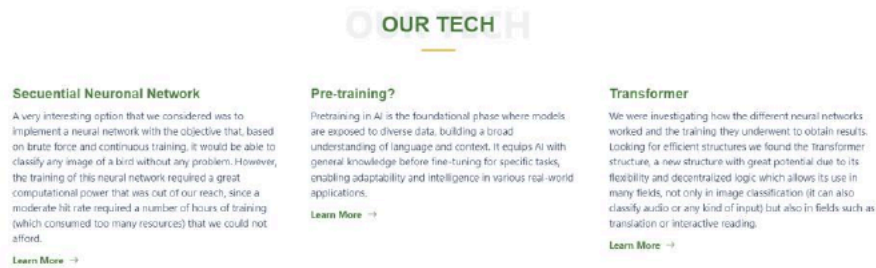


Imagen 9: Componente Nuestra tecnología final

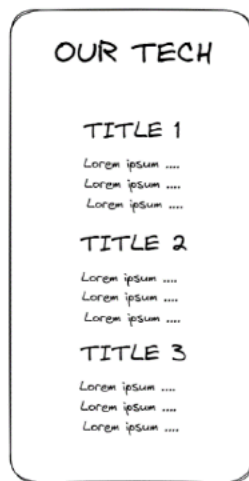


Imagen 10: Mock up componente Nuestra tecnología (móvil)

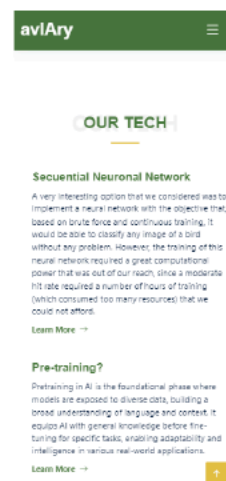


Imagen 11: Componente Nuestra tecnología final (móvil)

❖ Componente 'IA'

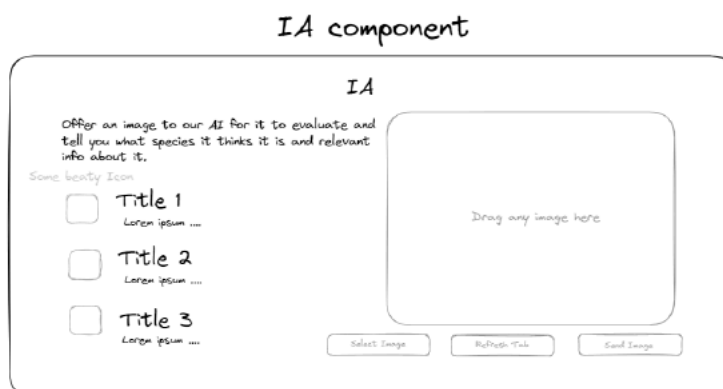


Imagen 12: Mock up componente IA

Antes hemos hablado de uno de los dos componentes más importantes de nuestra aplicación, ahora nos toca destacar al segundo, el componente de la 'IA', dividiéndose este en dos partes.

La primera de ellas ofrece al usuario información generalizada sobre la IA, pudiendo destacar un aviso que advierte al usuario de que el resultado de la predicción que la inteligencia artificial es directamente proporcional a la calidad de la imagen que el propio usuario le ofrezca.

La segunda parte es la más destacada del componente y una de las más destacadas de AvIAry, el lugar donde el usuario ofrece una imagen para que la inteligencia artificial dictamine un resultado. Esta imagen puede ser arrastrada hasta el recuadro o si el usuario pulsa el botón 'Seleccionar Imagen' aparecerá un selector de archivos del sistema.

Además esta parte del componente cuenta con fuertes medidas de seguridad y restricciones que impiden al usuario subir archivos que no sean imágenes, es decir que el usuario no puede arrastrar archivos que no terminen en .jpg .jpeg o .webp, además el selector de archivos del sistema de forma predeterminada no te mostrará ningún archivo que no sea una imagen. Además, también cuenta con una restricción de tamaño para un máximo de 2.5 mB para no sobrecargar la base de datos y ralentizar el funcionamiento de la aplicación.

Finalmente podemos ver un botón 'Refrescar Ventana' para, como su propio nombre indica, refrescar la ventana si fuere necesario y un botón 'Enviar Imagen' para que esta sea analizada y de forma automática se te muestre el resultado en el inicio (habiéndose automáticamente deslizado hacia arriba)

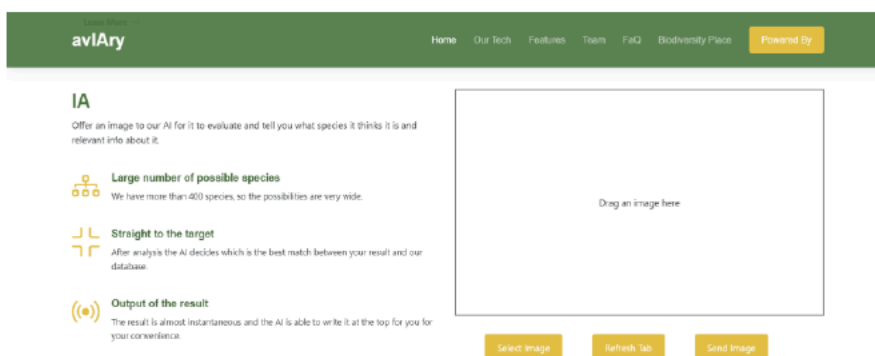


Imagen 13: Componente IA final



Imagen 14: Mock up componente IA (móvil)



Imagen 15: Componente IA final (móvil)

❖ Componente 'Características'

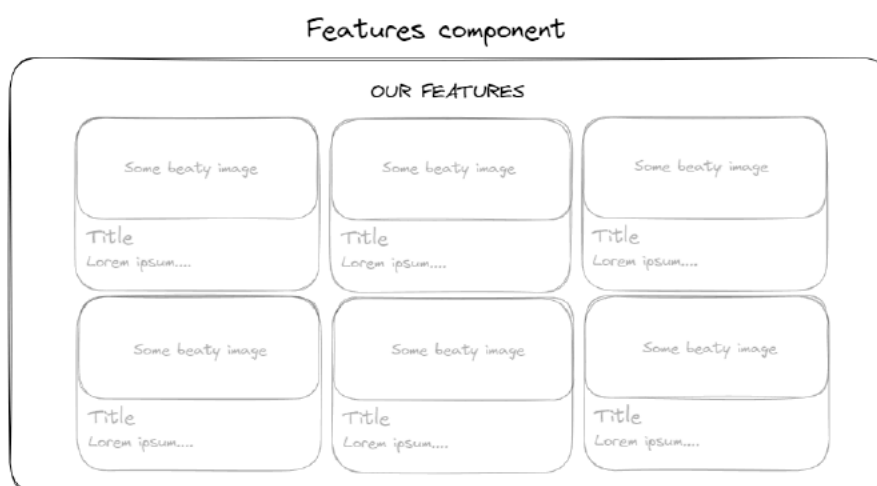


Imagen 16: Mock up componente Características

Con el objetivo de que el usuario conozca las principales características de AvIAry, este componente cobra vida con imágenes coloridas mencionando y explicando las principales características con las que nuestra aplicación cuenta, siendo estas:

- ❖ **Clasificación cromática de precisión:** Identifica aves por el espectro cromático de su plumaje, desde tonos vibrantes hasta variaciones sutiles, con una precisión excepcional.
- ❖ **Detección de morfología aviar:** Explora las diferencias en las formas de las alas, el pico y el cuerpo para una identificación precisa de las especies basada en la morfología.
- ❖ **Comparación de especies en tiempo real:** Compara varias especies de aves en tiempo real y obtenga información detallada sobre sus diferencias y similitudes visuales.
- ❖ **Interfaz fácil de usar y accesible:** Nuestra plataforma ofrece una interfaz fácil de usar y accesible para satisfacer tanto a los expertos en aves como a los aficionados principiantes, garantizando una experiencia de identificación agradable.
- ❖ **Infraestructura en nube escalable:** Desplegamos nuestros servicios en una infraestructura de nube escalable, lo que garantiza una alta disponibilidad, rendimiento y capacidad para gestionar cargas de trabajo variables.
- ❖ **Tratamiento de imágenes en tiempo real:** Nuestro sistema utiliza el procesamiento de imágenes en tiempo real para ofrecer resultados de clasificación de especies rápidos y precisos, por lo que resulta ideal para observadores de aves e investigadores.

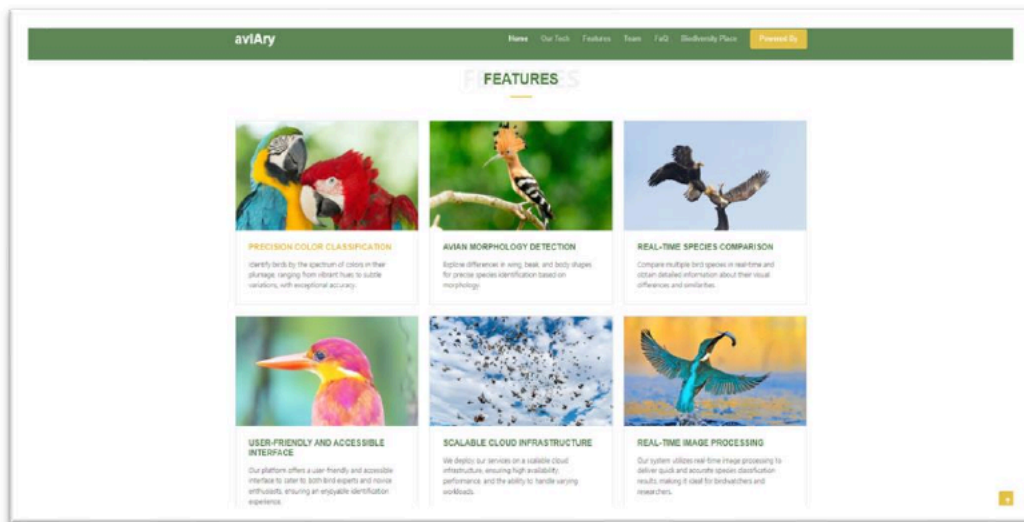


Imagen 17: Componente Características final



Imagen 18: Mock up componente Características (móvil)

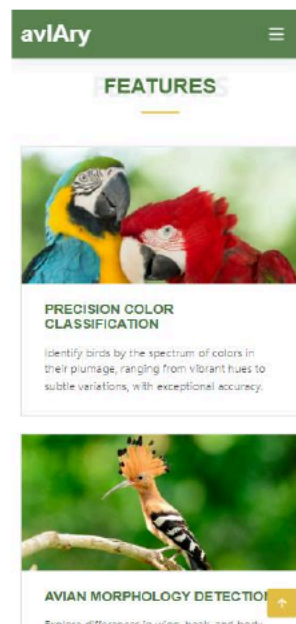


Imagen 19: Componente Características final (móvil)

❖ Componente 'Imagen Cortante'

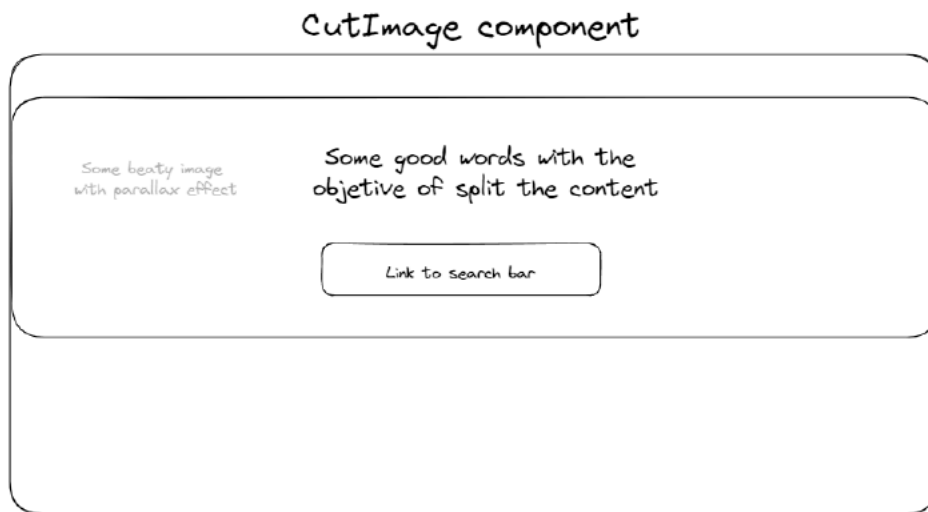


Imagen 20: Mock up componente Imagen Cortante

Este componente fue creado para dar un respiro visual al usuario y crear un contraste con el siguiente componente de forma creativa y diferente, para ello se ha usado de forma inteligente un efecto Parallax con una imagen que encaja perfectamente con el estilo y de la aplicación, además de incluir un mensaje de motivación al usuario para hago uso del buscador de especies del componente 'Inicio'.

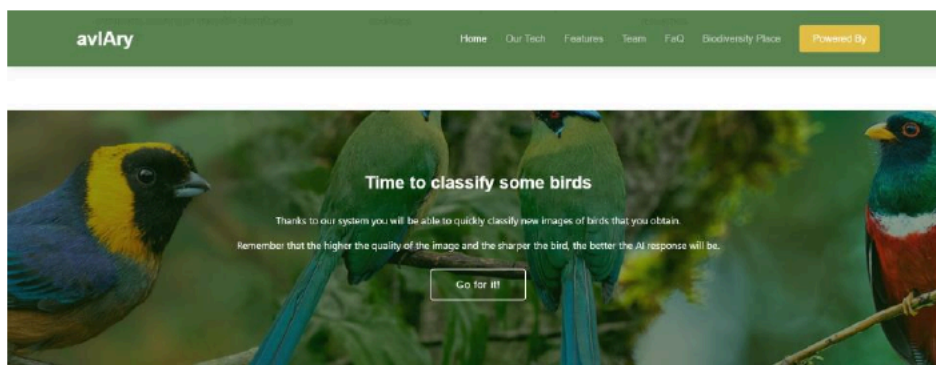


Imagen 21: Componente Imagen Cortante final



Imagen 22: Mock up componente Imagen Cortante (móvil)

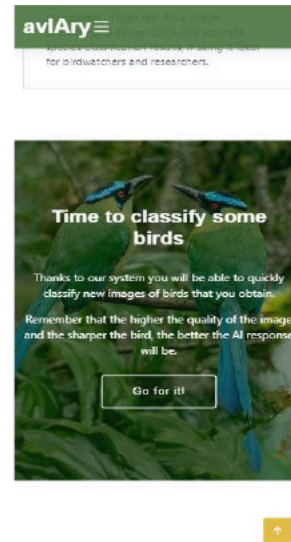


Imagen 23: Componente Imagen Cortante final (móvil)

❖ Componente 'Equipo'

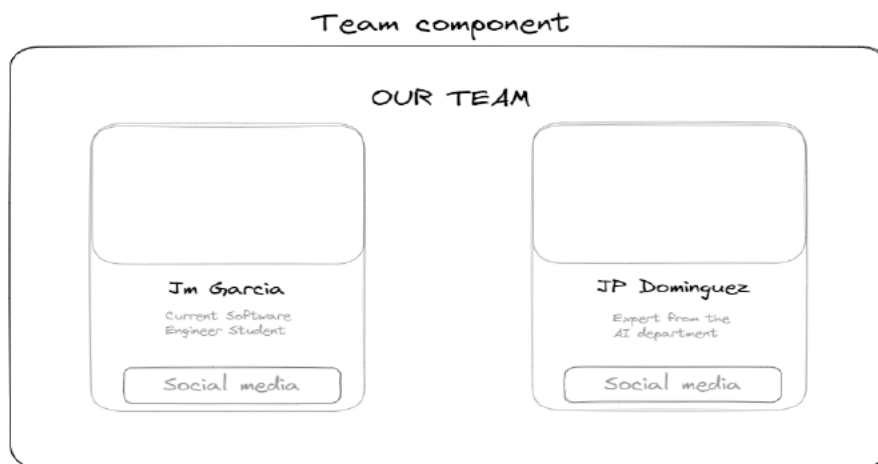


Imagen 24: Mock up componente Equipo

Dejar plasmado en un apartado de AvIAry, quienes han estado detrás de su creación y desarrollo era una prioridad por lo que este componente muestra dos tarjetas, una por cada participante, en este caso José M^a García Quijada y Juan Pedro Domínguez Morales donde cabe destacar una imagen de cada uno de ellos, así como su rol en el proyecto y sus respectivas redes sociales personales.

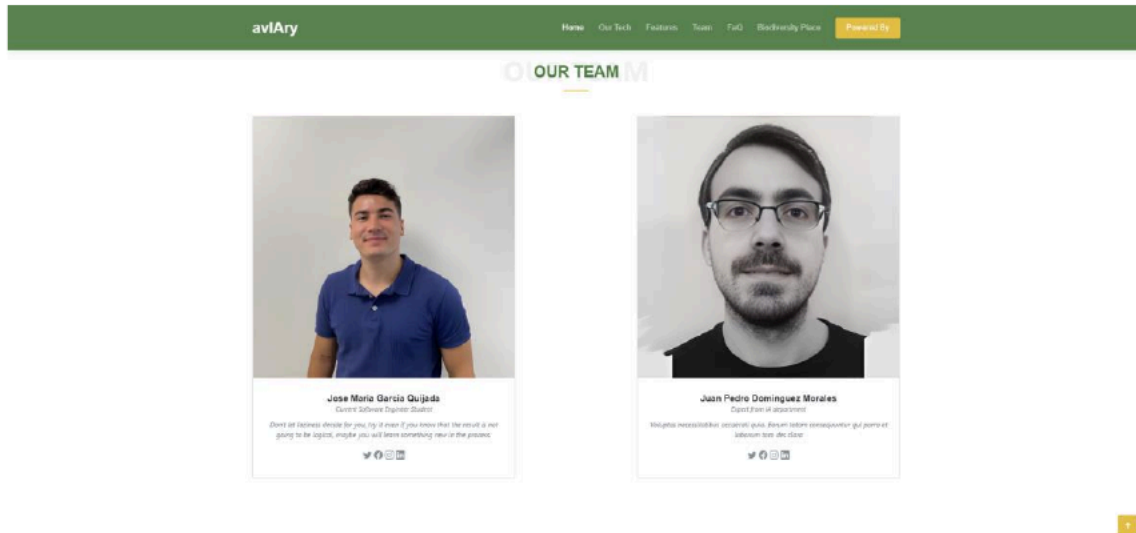


Imagen 25: Componente Equipo final

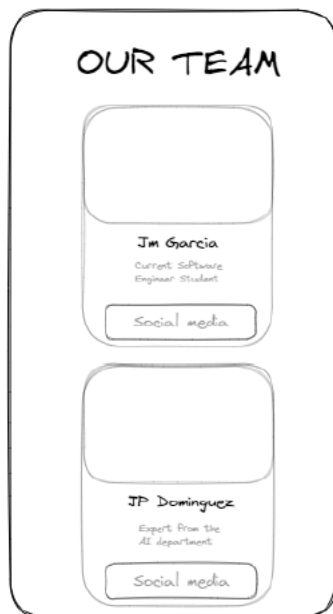


Imagen 26: Mock up componente Equipo (móvil)

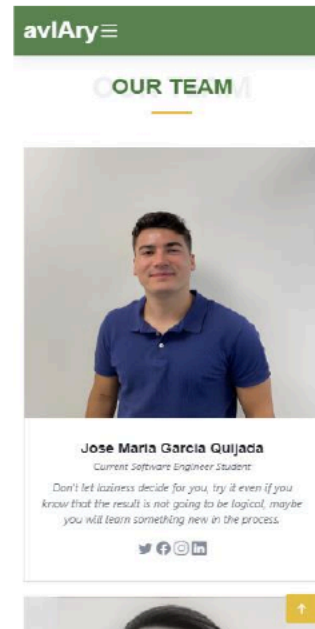


Imagen 27: Componente Equipo final (móvil)

❖ Componente 'Preguntas frecuentes'

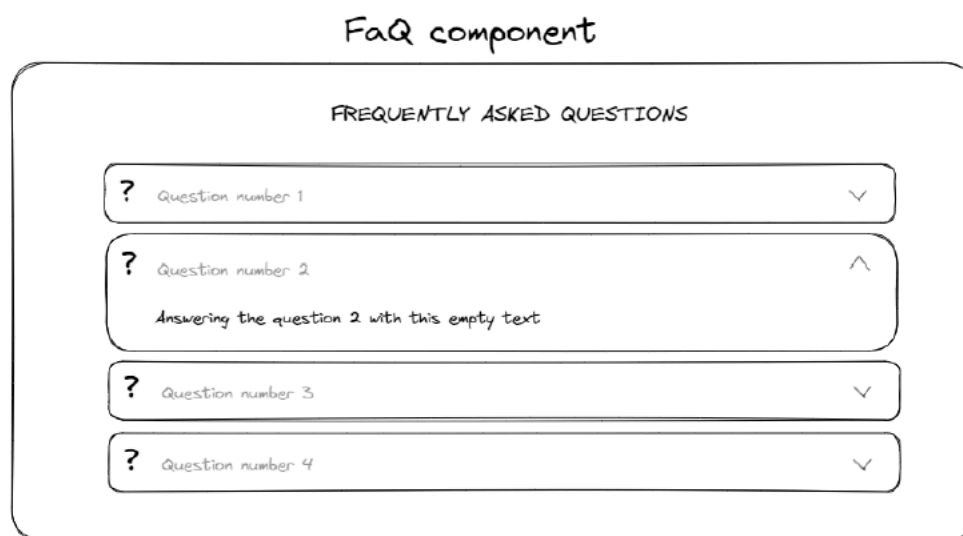


Imagen 28: Mock up componente Preguntas Frecuentes

En todas las aplicaciones donde el usuario que la consume es un apartado fundamental, cuenta con una sección de preguntas frecuentes, por lo que en AvIAry creemos que es una parte fundamental que no puede faltar.

Con todo ello en el componente se pueden observar diferentes preguntas frecuentes con su correspondiente respuesta asociada si activamos su desplegable, siendo estas.

❖ ¿Cómo funciona la clasificación de especies de aves en esta plataforma?

Utilizamos un algoritmo de aprendizaje profundo basado en PyTorch para analizar imágenes de aves y determinar su especie. Los usuarios pueden subir una imagen y nuestra IA la procesará y proporcionará la correspondiente clasificación de especies.

❖ ¿Qué debo hacer si no estoy seguro de la especie de un ave que he fotografiado?

Si tiene una foto de un ave, pero no está seguro de su especie, puede subir la imagen a nuestra plataforma. Nuestro método de IA intentará clasificarla. También puedes buscar una coincidencia visual en la base de datos de imágenes de aves.

❖ ¿Hasta qué punto es precisa la clasificación de las especies de aves?

Nuestro método de IA se ha entrenado con una amplia base de datos de imágenes de aves y es bastante precisa. Sin embargo, la precisión puede variar en función de la calidad de la imagen y de la dificultad de identificación. Siempre recomendamos cotejar la clasificación con otras fuentes si es necesario.

❖ ¿Es necesario registrarse para utilizar la plataforma?

No, no es necesario registrarse para realizar clasificaciones de especies de aves, basta con subir una imagen para ver los resultados.

❖ ¿Puedo utilizar la plataforma en mi dispositivo móvil?

Sí, nuestra plataforma es responsive y apta para móviles, por lo que puedes acceder a ella desde tu smartphone o tableta. Hemos diseñado la interfaz de usuario para que funcione a la perfección en distintos tamaños de pantalla.

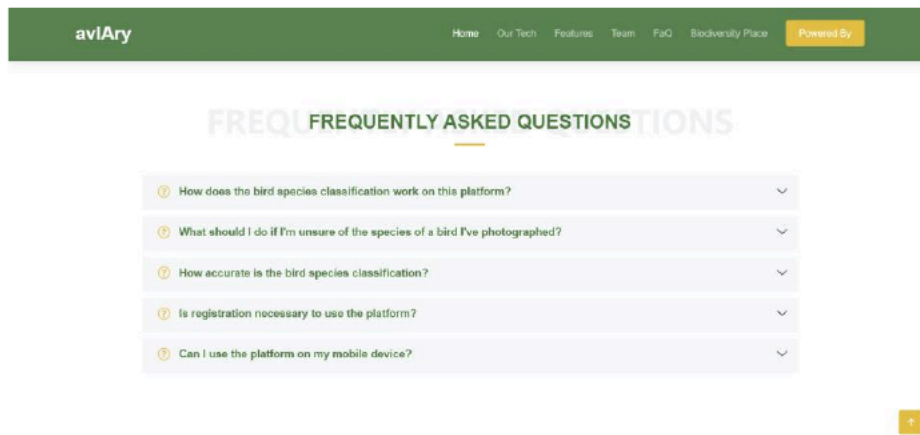


Imagen 29: Componente Preguntas Frecuentes final

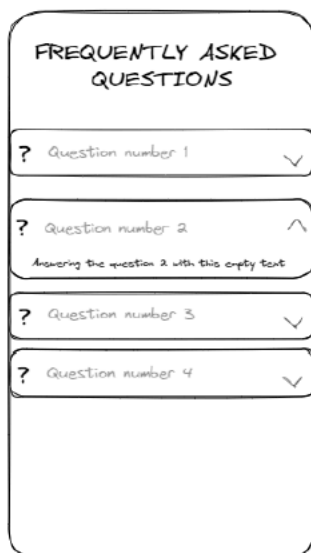


Imagen 30: Mock up componente Preguntas Frecuentes (móvil)

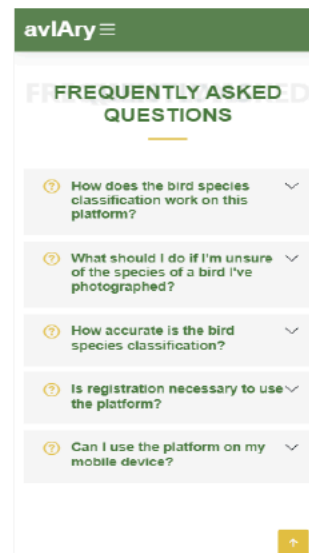


Imagen 31: Componente Preguntas Frecuentes final (móvil)

❖ Componente 'Marcas'

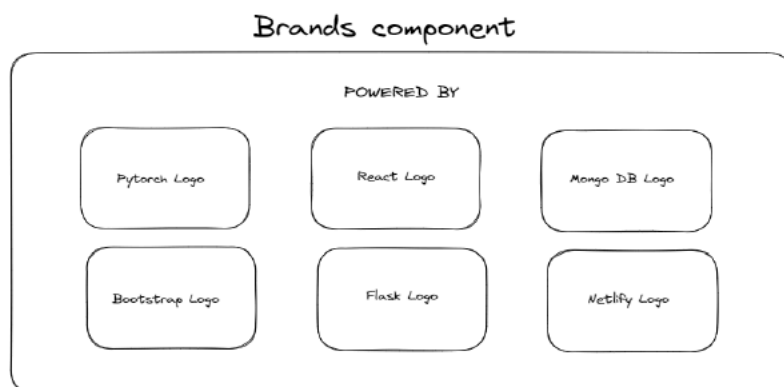


Imagen 32: Mock up componente Marcas

La tecnología que se ha utilizado para crear AvIAry debe tener su espacio, por lo que un componente mostrando todos los logos de las principales tecnologías usadas era algo necesario. Además, el usuario puede conocer que tecnologías están detrás de la aplicación que están usando.

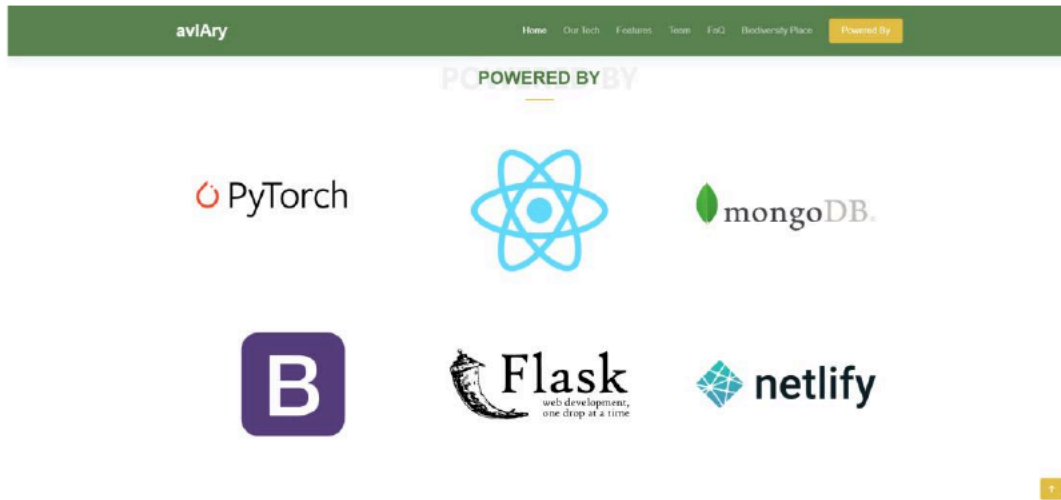


Imagen 33: Componente Preguntas Frecuentes final

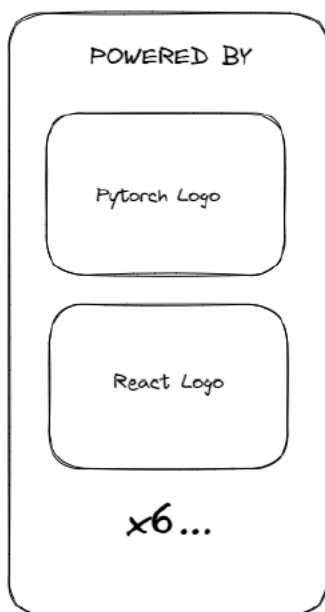


Imagen 34: Mock up componente Preguntas Frecuentes (móvil)



Imagen 35: Componente Preguntas Frecuentes final (móvil)

❖ Componente 'Pie de página'

Footer component

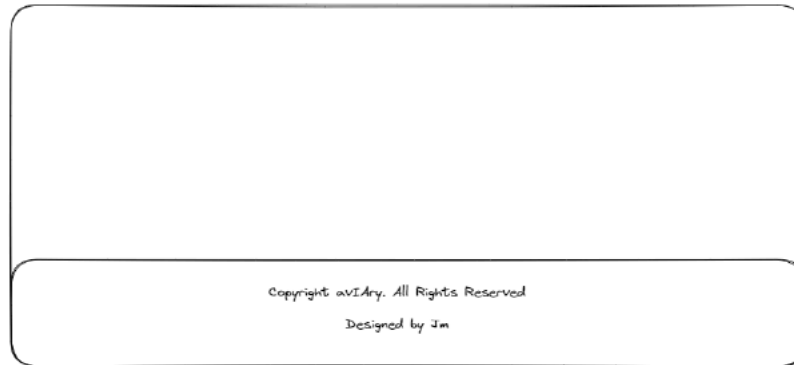


Imagen 36: Mock up componente Pie de Página

Para finalizar con la página principal, siempre es necesario incluir un pie de página resaltando al autor, José M^a García Quijada y los derechos reservados de AvlAry.

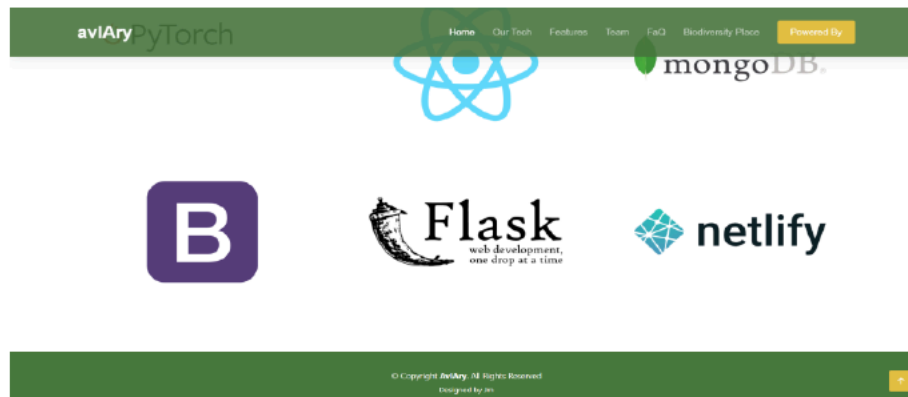


Imagen 37: Componente Pie de Página final



Imagen 39: Componente Pie de Página final (móvil)



Imagen 38: Mock up componente Pie de Página (móvil)

❖ Componente 'Biodiversidad'



Imagen 40: Mock up componente Biodiversidad

Este componente pertenece al apartado de 'Lugar de la biodiversidad' que se encuentra en la barra de navegación. Este apartado ha sido creado con el objetivo de fomentar la educación y la concienciación de temas de interés relacionados con el mundo aviar en nuestros usuarios. Creemos que es importante que nuestros usuarios dispongan de información útil y de fácil comprensión al alcance de la mano y con este apartado podemos hacerlo.

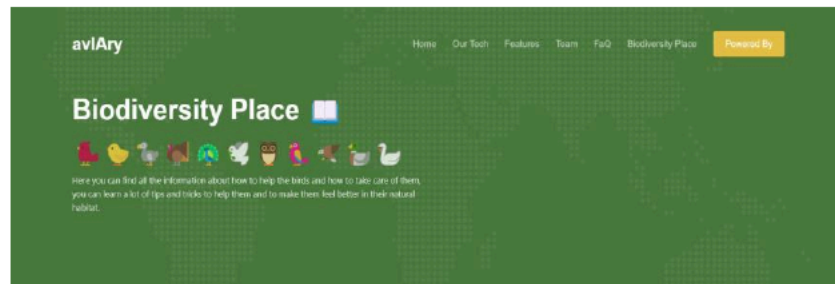


Imagen 41: Componente Biodiversidad final



Imagen 42: Componente Biodiversidad final (móvil)



Imagen 43: Mock up componente Biodiversidad (móvil)

❖ Componente 'Artículos'

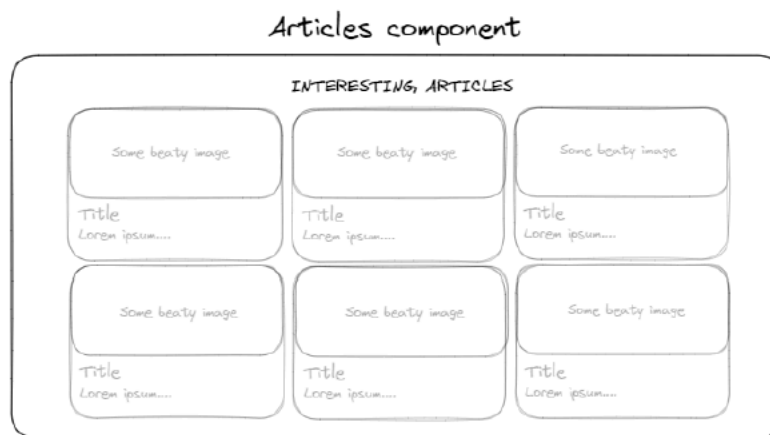


Imagen 44: Mock up componente Articulos

Para que el usuario pueda visualizar de forma rápida y sencilla cuales son los artículos disponibles, hemos creado esta interfaz simple que muestra los diferentes tópicos en cajas con diferentes imágenes y títulos relacionados con el tema en cuestión.

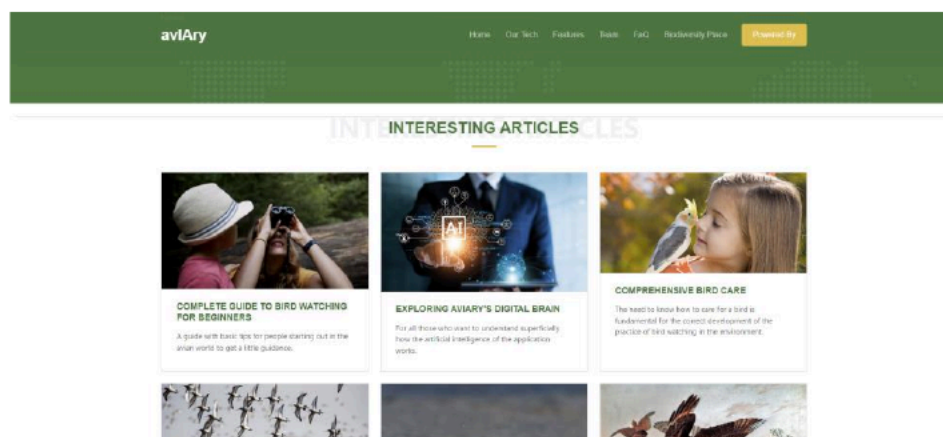


Imagen 45: Componente Articulos final



Imagen 46: Componente Biodiversidad final (móvil)

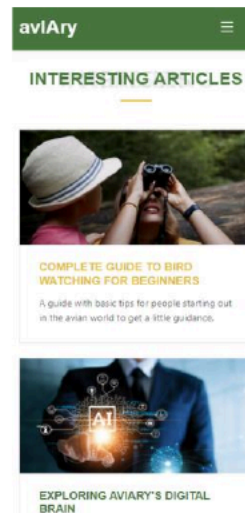


Imagen 47: Componente Biodiversidad final (móvil)

❖ Componente 'Artículo'

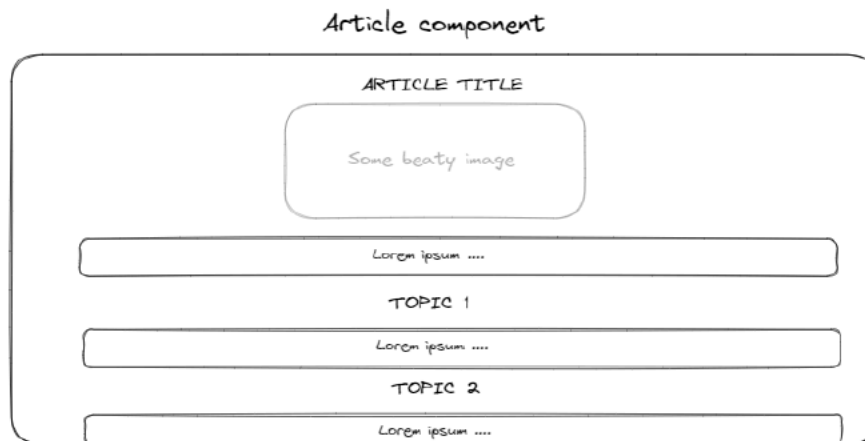


Imagen 48: Mock up componente Artículo

Finalmente, tenemos una gran diversidad de artículos muy interesantes que abarcan temas como el cuidado que debe de tener un pájaro, la migración de aves o el cambio climático y su impacto en estos seres vivos en otros muchos temas. Creemos que merecen mucho la pena disfrutar de estos artículos cortos que se dividen en diferentes tópicos para hacer que el lector no acumule fatiga visual al leerlos.

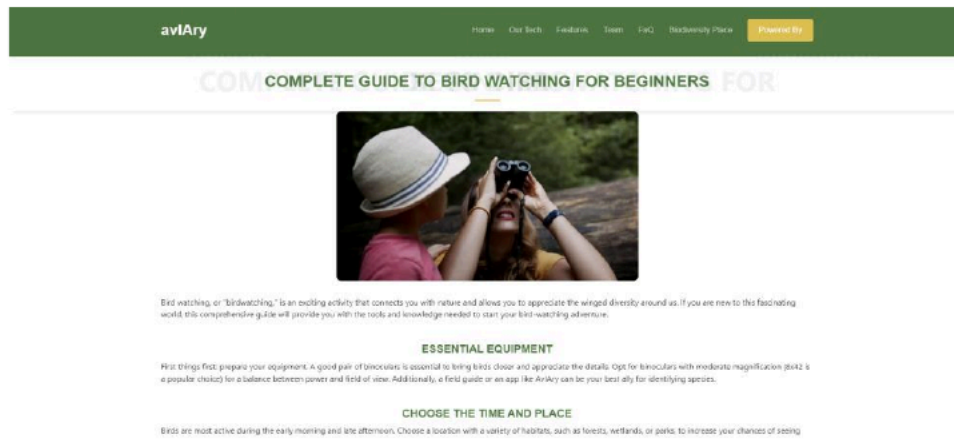


Imagen 49: Componente Artículo final



Imagen 50: Mock up componente Artículo (móvil)

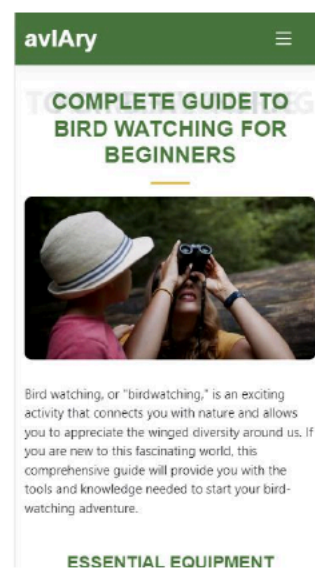


Imagen 51: Componente Artículo final (móvil)

6.1.2 SUBIDA A LA NUBE DEL FRONT-END

Ya tenemos una interfaz con la que el usuario puede navegar y disfrutar, no obstante, necesitamos conectarla con el resto de la aplicación, para ello necesitamos realizar una petición, en este caso con Axios, a un endpoint que aloje a la inteligencia artificial y toda su lógica.

```
const formData = new FormData();
formData.append('image', image);

axios.post('https://aviary-ia.onrender.com/upload', formData).then((res) => {
  setData(res.data[0].class);
  setClassificationResults(res.data);
});
```

Imagen 52: Endpoint que une al Front-end con la IA

Además, de la misma forma con Axios, debemos crear otro endpoint para conectarla con la API y todos los datos almacenados sobre las especies.

```
useEffect(() => {
  axios.get("https://aviary-backend.onrender.com/birds/get")
    .then(response => {
      setBackData(response.data);
    })
    .catch(error => {
      console.error("Error:", error);
    });
});
```

Imagen 53: Endpoint que une al Front-end con la API REST

Tras tener un acceso de entrada y salida de datos para la interfaz, con el objetivo de tener una versión estable de todo el Front-end creado en React, hemos usado los servicios de Netlify, una web que te permite tener de forma gratuita sitios web alojados. Para poder subir AvIAry a Netlify fue necesario crear una build (proceso que convierte toda la lógica de la aplicación en lenguaje HTML, CSS y JavaScript para que el navegador pueda entenderlo) y subir la carpeta build generada.

6.2 SERVIDOR IA: LA CREACIÓN DE UNA NUEVA FORMA DE PENSAR

Durante estos últimos años, el mundo de la tecnología ha sufrido grandes avances en la mayoría de sus ámbitos debido a los múltiples y revolucionarios descubrimientos obtenidos gracias al esfuerzo de miles de personas. En este artículo vamos a hablar de uno de ellos, el Transformer. Un avance que no solo ha revolucionado el mundo de la inteligencia artificial por completo, sino que también ha revolucionado a todos los ámbitos con los que este tipo de arquitectura de red neuronal ha sido vinculado.

6.2.1 INTRODUCCIÓN

Como brevemente se ha comentado, el Transformer, es un tipo de arquitectura de red neuronal que principalmente fue creada con el objetivo de mejorar la traducción, así como para tareas de procesamiento de lenguaje natural, no obstante, con el tiempo se ha ido demostrando que su límite no se encontraba ahí. Debido a que este cuenta con algoritmos de propósito general, posee una gran versatilidad para adaptarse a las diferentes tareas de análisis, pudiendo tener así un uso muy amplio siendo capaz de emplearse para, además de traducir; generación de textos, resumen de textos, interpretación de audio, identificación de entidades, responder a preguntas y sobre todo, siendo este el tema principal de este artículo, la clasificación de imágenes. Durante todo el artículo veremos cómo, con inteligencia artificial, ha sido posible clasificar más de un millar de imágenes de aves en sus correspondientes especies (llegando en torno a las cuatrocientas) de forma correcta, contando con una gran cantidad de imágenes (casi sesenta mil) con las que poder interactuar, llegando a dicha solución a través de diferentes caminos con resultados muy similares. La clasificación de imágenes de forma automatizada y precisa hasta hace relativamente poco tiempo fue siempre una gran incógnita. Sin embargo, la situación ahora es muy diferente ya que nunca ha sido posible de forma tan sencilla como es hoy en día. En la actualidad, para realizar con éxito esta tarea podemos crear redes neuronales de diversos tipos; convolucionales, densas, perceptrones, etc... Con esto puede parecer que la

arquitectura el Transformer no es ninguna novedad ya que no hemos descubierto como hacer dicha tarea con ella, pero si podemos afirmar que el método seguido con esta arquitectura ofrece porcentajes de acierto muy superiores a los demás teniendo en cuenta el tiempo de entrenamiento, además de contar con muchos menos parámetros que entrenar, traducándose esto en un menor coste para la GPU. Aun así, más adelante se expondrán a fondo diversas comparativas de rendimiento y optimización entre:

- ❖ Transformer
- ❖ Red neuronal convolucional o CNN

Siendo esta última una red neuronal soporte que nos ayudará a entender de forma acertada como de revolucionario es el Transformer en el mundo de la inteligencia artificial.

6.2.2 TRANSFORMER

Hemos estado hablando sobre la novedad de dicha estructura y hacia que ámbitos está enfocada, no obstante, no hemos entrado a definir qué es y cómo funciona por dentro. En primer lugar, empezaremos destacando a muy grandes rasgos cual es la idea principal de la estructura siendo esta gestionar completamente las dependencias entre la entrada y la salida con atención y recurrencia. Una vez que sabemos esa idea vamos a clarificarla definiendo su estructura y la función de cada una de las partes por la que está compuesto.

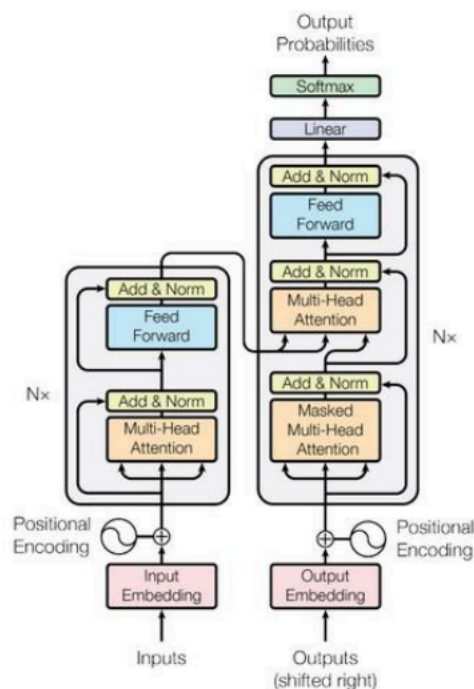


Imagen 54: Estructura de la arquitectura Transformer

A) BLOQUES ENCODER Y DECODER

La estructura básica y el esqueleto del Transformer se basa en un bloque Encoder y un bloque Decoder. La idea principal aquí es la retroalimentación de ambas partes ya que se emplea el bloque Encoder para analizar el contexto de la secuencia de entrada (a partir de

ahora será referenciada como "input" en el documento) y el bloque Decoder es el bloque encargado de generar la secuencia de la salida (será referenciada como "output") a partir de este contexto. Para entender esta idea de forma más clara podemos apreciar en la parte inferior cómo funciona con texto en francés y su correspondiente traducción al inglés.

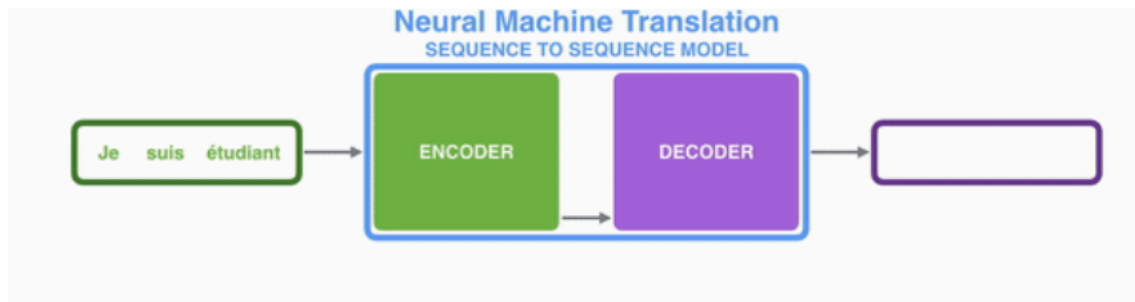


Imagen 55: Paso 1 de la acción Encoder-Decoder

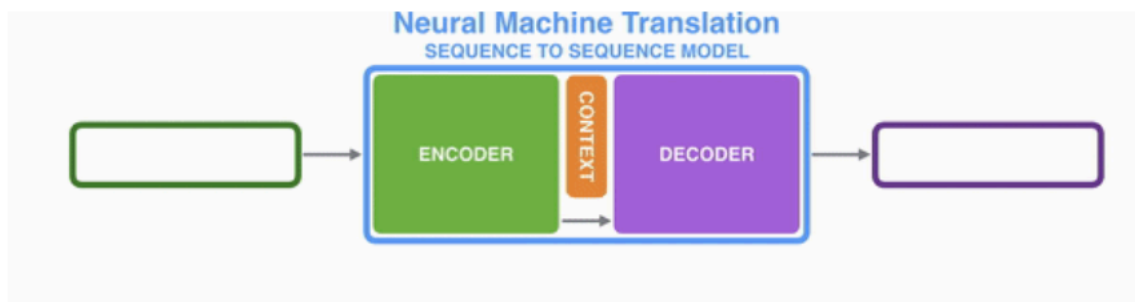


Imagen 56: Paso 2 de la acción Encoder-Decoder

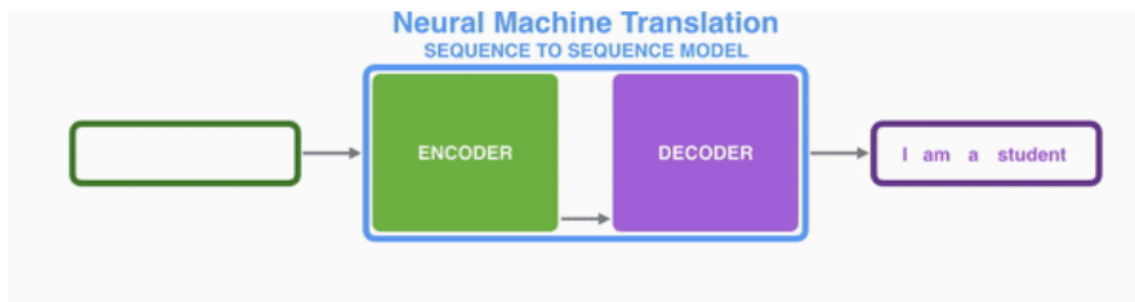


Imagen 57: Paso 3 de la acción Encoder-Decoder

Para clarificar aún más si cabe, los dos bloques Encoder y Decoder son los resaltados en la siguiente imagen.

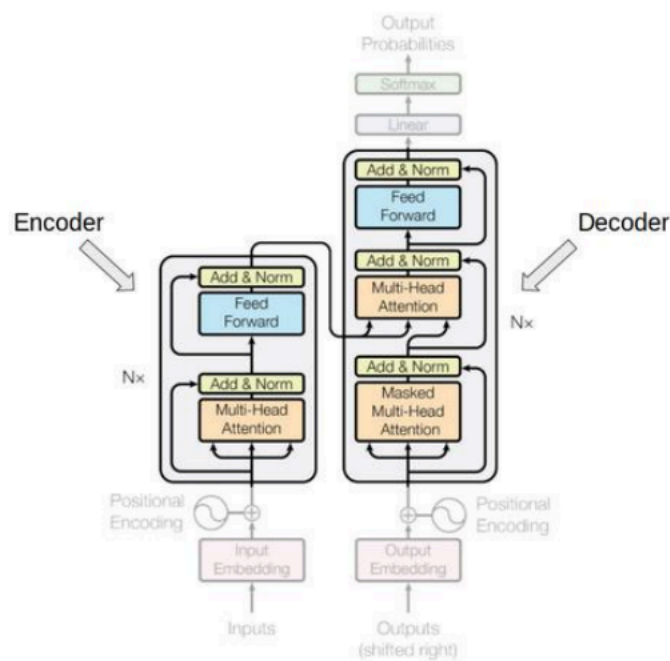


Imagen 58: Bloques Encoder y Decoder en Transformer

B) EMBEDDING

El objetivo principal de la arquitectura es que el encoder y el decoder puedan trabajar bien, para ello es necesario realizar un embedding del texto, imagen o audio (diversidad de tipos de inputs) según sea el uso que tenga El Transformer. La palabra Embedding, en este ámbito, significa una representación vectorial de cualquiera que sea su input. El objetivo de esta transformación a vectores es que conseguir una representación mediante número de dicho input y que, además, el input que tenga una relación similar estará cerca en el espacio vectorial.

C) POSITIONAL ENCODING

Pese a la pequeña posición que tiene esta parte en el diagrama de la arquitectura es una de las partes más importantes de esta, ya que juega un papel principal para que el modelo pueda entender la secuencia que le hemos pasado. El Transformer no cuenta con un mecanismo de recurrencia, como otras redes neuronales, ya que utiliza el mecanismo "Multi-Head Attention" siendo esta característica uno de los principales pilares del Transformer. Como en la introducción se ha resaltado, gracias a dicha característica esta arquitectura permite capturar dependencias mucho más largas y obtener mejores resultados en un menor período de tiempo. Esta característica se verá en mayor profundidad más adelante. Como se puede ver en el gráfico superior el input fluye de manera simultánea a entre el encoder y el decoder. Así pues, necesariamente debemos añadir información sobre la posición de cada input en el vector de secuencia. Para todo ello es necesario aplicar una función sinusoidal, siendo esta:

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{D_{model}}}}\right)$$

$$PE(pos, 2i + 1) = \cos\left(\frac{pos}{10000^{\frac{2i}{D_{model}}}}\right)$$

D) ATENCIÓN PROPIA (SELF-ATTENTION)

Como se ha comentado anteriormente, el Transformer cuenta con un mecanismo de auto atención que le permite saber con qué otro input de toda la secuencia está relacionado el input que se procesa en ese instante de tiempo. A diferencia del resto de redes, el Transformer gracias a esta cualidad no sufre de memoria a corto plazo a la hora de recordar un dato anteriormente procesado necesario a la hora de interpretar una situación en el presente. Con un ejemplo simple aplicado a texto debería verse de forma más clara esta idea.

Pepa y Manuel han trabajado en el proyecto de IA pero ella no espera aprobar

Para otros tipos de red neuronal, la asociación de "ella" con "Pepa" era un gran problema debido a su memoria a corto plazo y la gran distancia entre estas dos palabras en el contexto de la frase. No obstante, este mecanismo de autoatención resuelve por completo este problema, obteniéndose así una memoria prácticamente infinita (limitación de computación).

❖ ATENCIÓN MEDIANTE PRODUCTO ESCALAR

En primer lugar, antes de intentar entender cómo funciona el verdadero mecanismo de atención que emplea el Transformer, es necesario entender cómo funciona la versión en la que se basa dicho mecanismo. La atención mediante producto escalar se basa en tres valores de entrada:

- ❖ Valor Q: Representando a la Query que representa el vector del input.
- ❖ Valor K: Siendo este valor Keys el resto de inputs introducidos.
- ❖ Valor V: Value obtiene el valor vectorial de dicho input en el momento.

Este mecanismo se basa, en el producto de los valores Q y K para después tratar al resultado con un escalado, una máscara (opcional) y la función softmax para que devuelva los valores en rango [0,1]. Una vez finalizado ese proceso volvemos a realizar el producto entre el resultado obtenido y el valor V.

Scaled Dot-Product Attention

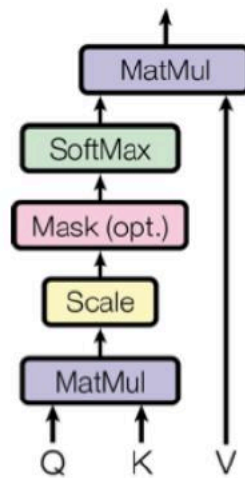


Imagen 59: Esquema sobre la atención

A través de la siguiente fórmula propuesta se puede calcular este índice de atención descrito e ilustrado anteriormente, destacando el producto entre los valores Q y T que intenta conseguir una aproximación de la alineación de los propios vectores para devolver un peso para cada input, la división entre la raíz del tamaño de K para evitar valores de gran tamaño perjudiciales para softmax y el producto con V para quedarnos solo con los inputs relevantes y desechar los inputs que no necesitemos, siendo esta una gran forma de optimizar recursos y no procesar datos de cierta irrelevancia .

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{D_k}})V$$

❖ ATENCIÓN MEDIANTE DIFERENTES CABEZAS

Una vez que hemos aprendido cómo funciona la atención mediante producto escalar, la atención mediante diferentes cabezas emplea un sistema muy parecido. No obstante, en este caso El Transformer emplea una proyección de Q, K y V en H espacios lineales. En este caso aparece una nueva variable H, que representa el número de cabezas con las que cuenta el sistema. Así pues, con un sistema formado por diversas cabezas podemos hacer que cada cabeza se centre en aspectos diferentes, para así después concatenar los diferentes resultados. Pudiendo tener múltiples representaciones de importancia de cada input, permitiendo que dicho input no sea el dominante en el contexto en el que se emplee dicho Transformer. Con todo ello podemos afirmar el gran potencial que tiene esta arquitectura ya que nos permite aprender dependencias mucho más complejas en el mismo tiempo de entrenamiento gracias a que la proyección lineal reduce el tamaño de cada vector.

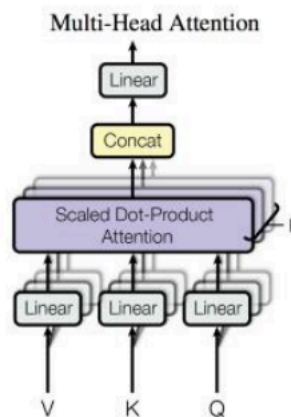


Imagen 60: Esquema sobre la atención múltiple

Para hacernos una pequeña idea y entender de forma completa el potencial final de este mecanismo de atención mediante diferentes cabezas, en la siguiente imagen podrá apreciarse una comparativa en la traducción para que se vea de forma intuitiva y gráfica los niveles de atención entre una oración en inglés y en francés (siendo el léxico de ambos idiomas completamente diferente y en orden distinto).

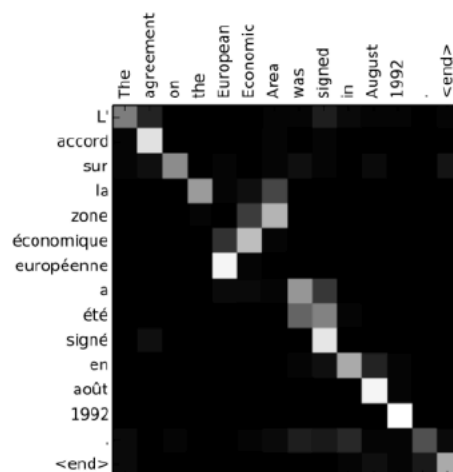


Imagen 61: Representación de la atención con oraciones en inglés y francés

E) FUNCIONAMIENTO FINAL

En primer lugar, se realiza la codificación en el bloque Encoder, estando este proceso formado por los siguientes pasos.

- ❖ Embedding del conjunto de inputs con el objetivo de transformar cada input para que el sistema pueda trabajar de forma numérica, en este caso en vectores. Gracias a esta propiedad es posible que acepte tanto texto, imágenes, audio, etc... Ya que cualquier input va a ser traducido a su forma vectorial.
- ❖ Pasamos por el componente Positional Encoding para agregarle la componente de la posición a cada vector.

- ❖ Aplicamos el mecanismo de atención de diferentes cabezas explicado anteriormente.
- ❖ Introducimos los datos que teníamos hasta ahora en la capa neuronal Feed Forward.

En segundo lugar, se realiza la decodificación en el bloque Decoder, estando este proceso formado por los siguientes pasos.

- ❖ Al igual que antes hacemos un embedding pero en este caso debemos hacerlo del resultado del punto justo anterior.
- ❖ Pasamos por el componente Positional Encoding para agregarle la componente de la posición a cada vector.
- ❖ Aplicamos el mecanismo de atención de diferentes cabezas en este caso al output del resultado del punto justo anterior.
- ❖ Volvemos a emplear el mecanismo de atención, no obstante, en este caso como tenemos los resultados del Encoder, empleamos dichos resultados como valor Q y K y el resultado obtenido en el mecanismo de atención anterior lo empleamos como V.
- ❖ Introducimos los datos que teníamos hasta ahora en la capa neuronal Feed Forward.
- ❖ Terminamos el proceso con una capa Lineal y Softmax para conseguir el objetivo final, obtener una probabilidad y devolver el input que tenga esta más alta como el siguiente.

6.2.3 MODELOS EMPLEADOS PARA RESOLVER EL PROBLEMA PROPUESTO

Una vez que hemos hablado sobre los beneficios del descubrimiento de la arquitectura el Transformer y sobre cómo son sus componentes y su funcionamiento a fondo, es hora de hablar de nuestras implementaciones.

A) VISION TRANSFORMER (ViT)

En primer lugar, para la propuesta en la que no se han utilizado librerías externas hemos utilizado un modelo de Vision Transformer (a partir de ahora será referenciado como "ViT" en el documento). No obstante, ¿cuál es la diferencia entre este ViT y la arquitectura el Transformer explicada anteriormente? Para entender esta arquitectura de forma correcta debemos dividirla en tres componentes.

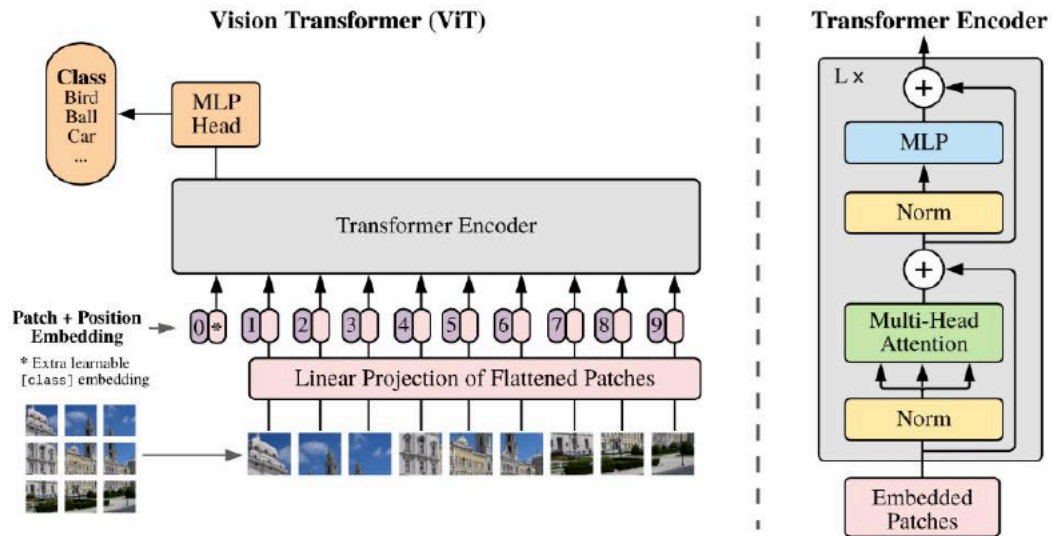


Imagen 62: Esquema de la estructura ViT

❖ EMBEDDING

Este modelo ViT está formado por un bloque Encoder que está minuciosamente pensado para que reciba input, en este caso la imagen del ave separada en diferentes pedacitos (parches) siendo estos proyectados. Tras este paso, deberemos calcular para cada uno de estos Patch y Position Embedding.

Para la clasificación, concatenamos una clase embedding aprendible con las otras proyecciones de parches, cuyo estado en la salida sirve como información de clase. Este token de clase adicional se añade al conjunto de tokens de la imagen que se encarga de agregar la información global de la imagen y la clasificación final. Es capaz de aprender esta agregación global mientras pasa y aprende a través de las capas de atención. También añadimos una posicional embedding a los parches, para establecer un cierto orden en los parches que hemos creado como entrada. No obstante, no debemos subestimar esto último ya que es muy necesario para que el modelo aprenda el orden de la imagen ya que los transformadores no son capaces de recordar el orden de las entradas. Si los parches de la imagen se reordenan, se pierde el significado de la imagen original. Por ello, añadimos ese posicional embedding a nuestros parches seguir la secuencia.



Imagen 63: Representación del orden de parches

❖ **ENCODER**

Este bloque Encoder es muy similar al explicado anteriormente procedente de la arquitectura de El Transformer. Los únicos cambios significativos en el cambio de orden entre la normalización y la capa de atención, siendo esta de tipo diferentes cabezas y la normalización y la capa Feed Forward o en este caso MLP (Perceptrón Multicapa) haciéndose siempre después la conexión residual.

❖ **MLP HEAD**

La arquitectura de este Transformer está diseñada para que el output del Bloque Encoder solo emplee la primera salida para conectarse a un MLP (Perceptron Multicapa) que hará la clasificación. Esta primera salida viene dada por la primera entrada la cual es una entrada vacía (creada en la primera posición de inputs antes que todos los parches) que solo sirve para realizar la conexión al clasificador MLP.

B) RED NEURONAL CONVOLUCIONAL (CNN)

Tras la construcción de nuestro propio ViT, pasamos ahora a construir una red neuronal sencilla que nos va a servir de gran ayuda para tener una comparativa entre ambas redes y ver sus respectivos resultados al final de este documento.

En primer lugar, debemos aclarar que el objetivo de esta segunda red neuronal es exactamente el mismo que el del ViT, por lo que para su creación decidimos que debía contener lo siguiente.

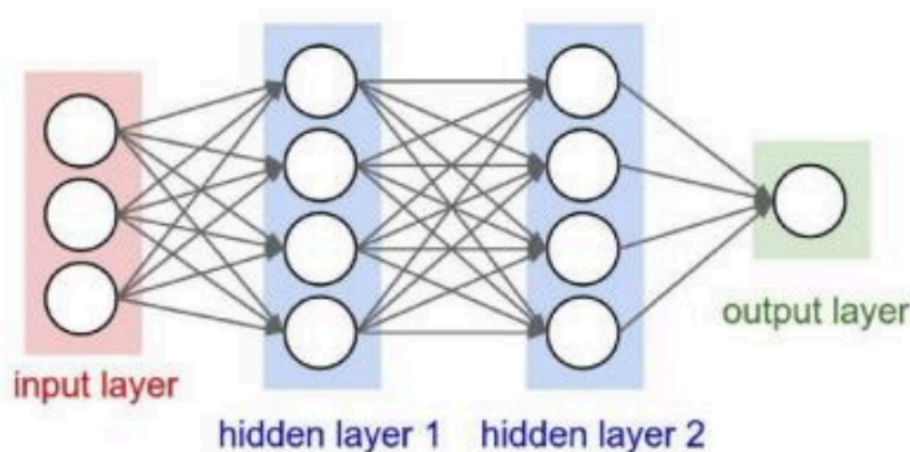


Imagen 64: Relaciones entre capas en una CNN

❖ **ALGORITMO DE DESCENSO DE GRADIENTE EN MINI LOTES**

La red neuronal debía seguir el algoritmo de descenso de gradiente en mini lotes, para entender que es este algoritmo y cómo funciona podemos decir que este algoritmo se encarga de dividir el conjunto de datos de entrenamiento en pequeños lotes que se utilizan para calcular el error del modelo y actualizar los coeficientes de este. Este algoritmo no es

único en su ámbito existen más como el gradiente estocástico o el gradiente por lotes, pero este trata de encontrar un equilibrio entre la robustez del descenso de gradiente estocástico y la eficiencia del descenso de gradiente por lotes. Suele ser la implementación más común del descenso de gradiente utilizada en el campo del deep learning.

❖ FUNCIONAMIENTO CONVOLUCIONAL

Puesto que es una red neuronal convolucional, su funcionamiento es realmente simple ya debido a su naturaleza convolucional es necesario que su entrenamiento también lo sea, por lo que el método de entrenamiento de dicha red en un bucle que finaliza cuando el número de épocas llega al número de épocas solicitado.

6.2.4 USO DE MODELOS PREENTRENADOS

La elección de un modelo subyacente desempeña un papel crucial en el rendimiento y la eficiencia de las tareas. En este contexto, la utilización de modelos preentrenados, como los de ResNet o Hugging Face, emerge como un catalizador para transformar radicalmente la forma en que estas arquitecturas procesan y comprenden información visual.

Antes de adentrarnos en la integración de ResNet18 o Base-patch-16 con nuestras creaciones, es esencial comprender la noción de "preentrenamiento". Un modelo preentrenado es aquel que ha sido previamente expuesto a vastos conjuntos de datos para aprender patrones y características generales. En el caso de ResNet18, este modelo ha pasado por una fase inicial de entrenamiento en conjuntos de datos masivos, lo que le confiere la capacidad de reconocer una amplia gama de características visuales, desde bordes simples hasta conceptos complejos, de igual forma con Base-patch-16 el cual es un punto de control para ViT creado por Hugging Face.

Con el objetivo de crear más variedad y enriquecer el estudio y la investigación entre las diferentes arquitecturas propuestas, hemos decidido aplicar modelos preentrenados, tanto a la red neuronal convolucional como a la arquitectura Transformer para estudiar y comparar como se desarrollan sus comportamientos cuando hacen uso de dicho catalizador. Así pues, con las cuatro propuestas actuales comenzaremos un estudio para dictaminar, a través de métricas reales, cual de todas es la más completa y la que emplearemos en AvlAry para que los usuarios disfruten.

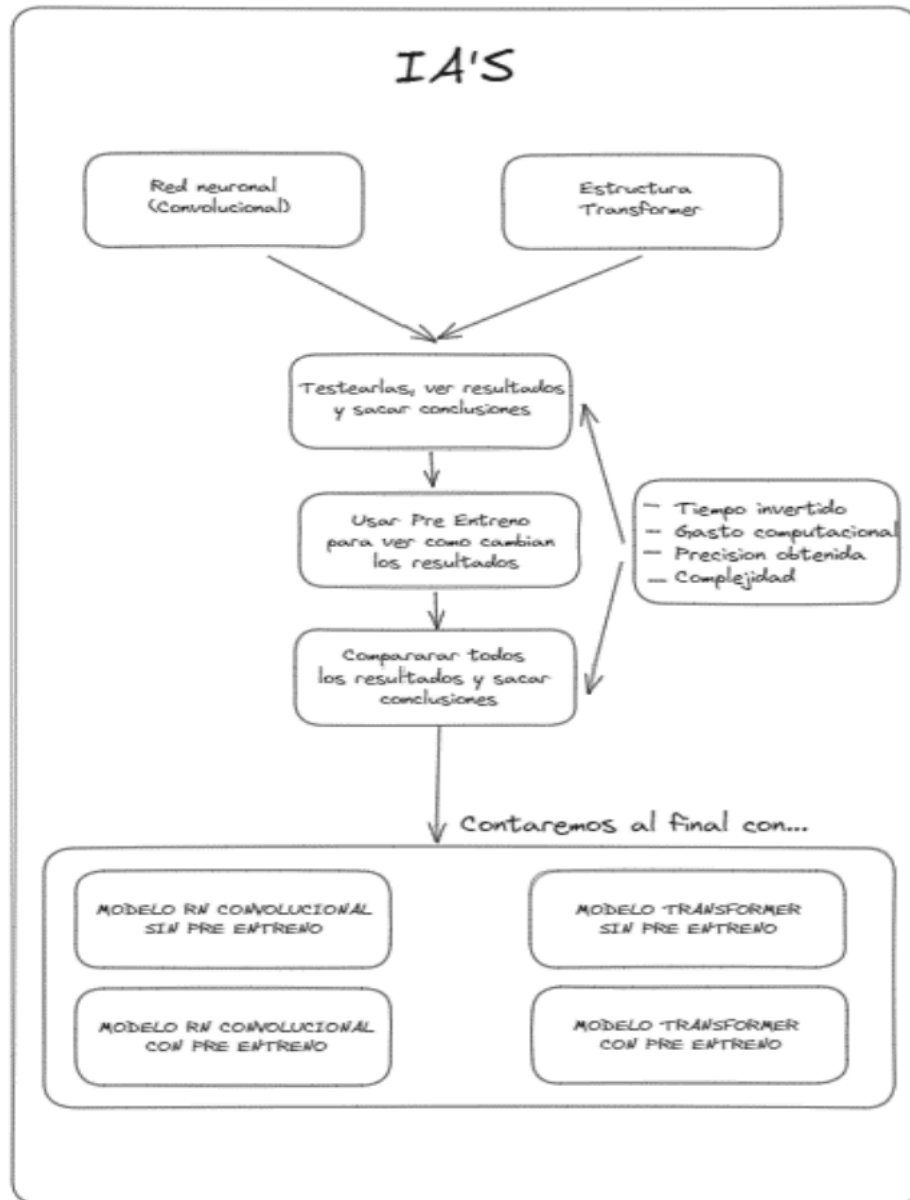


Imagen 65: Mapa conceptual sobre las IA's a crear

6.2.5 ESTUDIO DE LAS IA'S, RESULTADOS OBTENIDOS Y COMPARATIVA

Tras explicar de forma detenida cómo funciona una arquitectura Transformer, es hora de ponerla a prueba frente a una red neuronal convolutiva (CNN) para ver cuál de ellas posee un mejor rendimiento. Además, cada una de ellas hará uso de un preentrenamiento; por un lado, la CNN usará el modelo preentrenado ResNet18 perteneciente a ResNet, mientras que por otro lado El Transformer hará uso del punto de control Base-patch-16 de Hugging Face.

El objetivo al final del estudio será ver cómo actúa cada estructura y como ese comportamiento puede llegar a variar cuando se le aplica un preentrenamiento.

Para poder estudiar el rendimiento de nuestras diferentes implementaciones, debemos tener claros cuales son los parámetros a tener en cuenta. En este caso, como queremos abordar todas las métricas relevantes, hemos decidido calcular:

- ❖ **Exactitud (Accuracy):** La exactitud es una métrica básica que indica la proporción de predicciones correctas sobre el total de predicciones.

$$\text{Exactitud} = \text{Predicciones correctas} / (\text{Predicciones totales} * 100)$$

- ❖ **Precisión (Precision):** Proporción de verdaderos positivos sobre la suma de verdaderos positivos y falsos positivos.

$$\text{Precisión} = \text{Verdaderos positivos} / (\text{Verdaderos positivos} + \text{Falsos positivos})$$

- ❖ **Sensibilidad (Recall):** Proporción de verdaderos positivos sobre la suma de verdaderos positivos y falsos negativos.

$$\text{Sensibilidad} = \text{Verdaderos positivos} / (\text{Verdaderos positivos} + \text{Falsos negativos})$$

- ❖ **Puntuación F1 (F1 Score):** Es la media armónica de precisión y sensibilidad.

$$\text{Puntuación F1} = 2 * (\text{Precisión} * \text{Sensibilidad}) / (\text{Precisión} + \text{Sensibilidad})$$

- ❖ **Tiempo lúcido:** Tiempo que se necesita para llegar a la última época donde las métricas son ascendentes, es decir, cuando una época proporciona resultados peores que la anterior, por regla general, los resultados no suelen ser notoriamente mejores.

$$\text{Tiempo lúcido} = \text{Tiempo medio por época} * \text{nº épocas}$$

- ❖ **Gasto computacional:** Cantidad de recursos energéticos que se necesitan para el correcto de aprendizaje de la estructura.

$$\text{Gasto computacional} = \text{Tiempo lúcido} * \text{nº Kw/h consumidos}$$

- ❖ **Pérdida de la época (Epoch loss):** Se refiere a la pérdida (loss) acumulada durante una época (epoch) completa durante el entrenamiento de un modelo. A mayor pérdida, menos precisas están siendo las predicciones del modelo.

A) ESTUDIO DEL RENDIMIENTO DE LA RED NEURONAL CONVOLUCIONAL (CNN)

Tras la creación del código y la lógica referente a la CNN es hora de ver el desempeño que esta ha tenido a la hora de entrenar.

```
Epoch 18: 100% ██████████ | 1525/1525  
[35:15<00:00, 1.21s/it, acc=73.7, loss=1.022]  
Training dataset. Got 37894 out of 48788 images correctly (79.131%). Epoch loss: 1.022  
Accuracy: 73.75%  
Precision: 79.13%  
Recall: 73.75%  
F1 Score: 73.61%
```

Imagen 66: Valores en consola durante el entreno con CNN

❖ RED NEURONAL CONVOLUCIONAL (CNN) SIN PREENTRENO

Una de las mejores formas de ver cómo evoluciona un modelo a lo largo de su entrenamiento es viendo sus resultados de forma progresiva a lo largo de las épocas en una gráfica, por lo que para cada una de las métricas anteriores (excepto para el tiempo lúcido y el gasto computacional que requieren de tiempo en segundos) hemos creado su correspondiente gráfica.

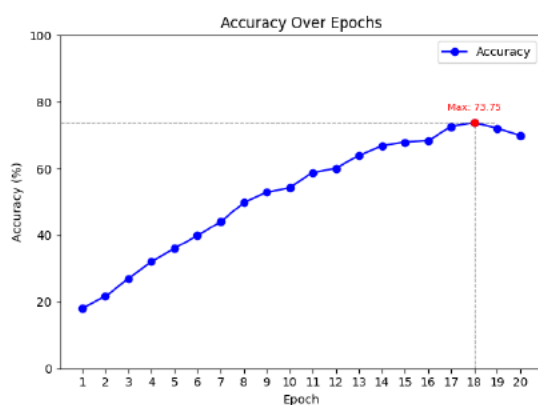


Imagen 67: Gráfica exactitud CNN sin preentrenado

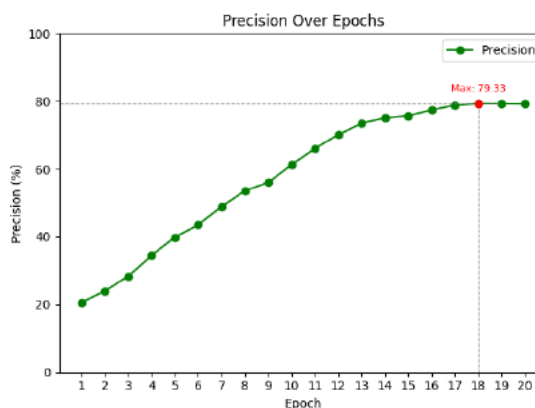


Imagen 68: Gráfica precisión CNN sin preentrenado

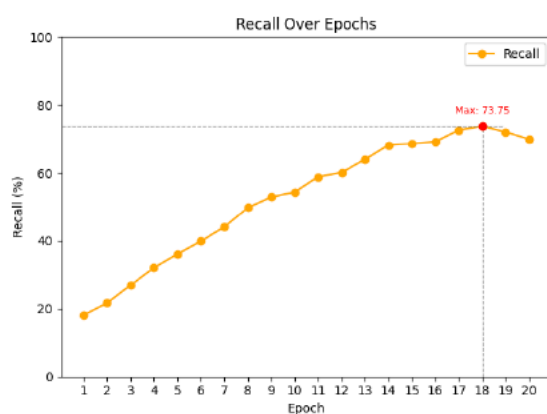


Imagen 69: Gráfica recall CNN sin preentrenamiento

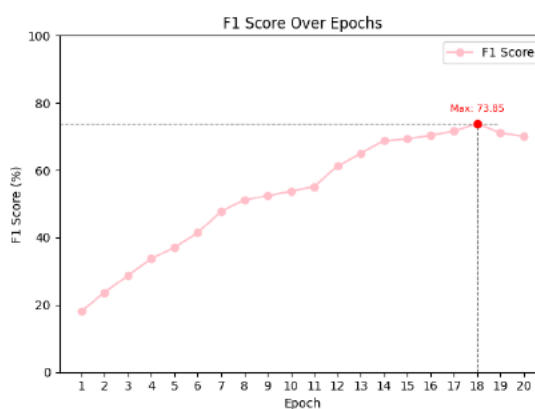


Imagen 70: Gráfica puntuación F1 CNN sin preentrenado

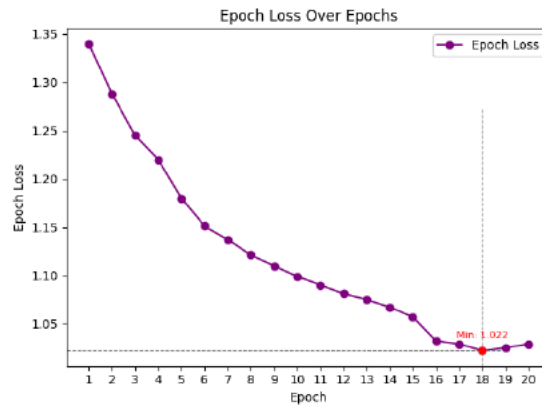


Imagen 71: Gráfica pérdida en épocas CNN sin preentrenamiento

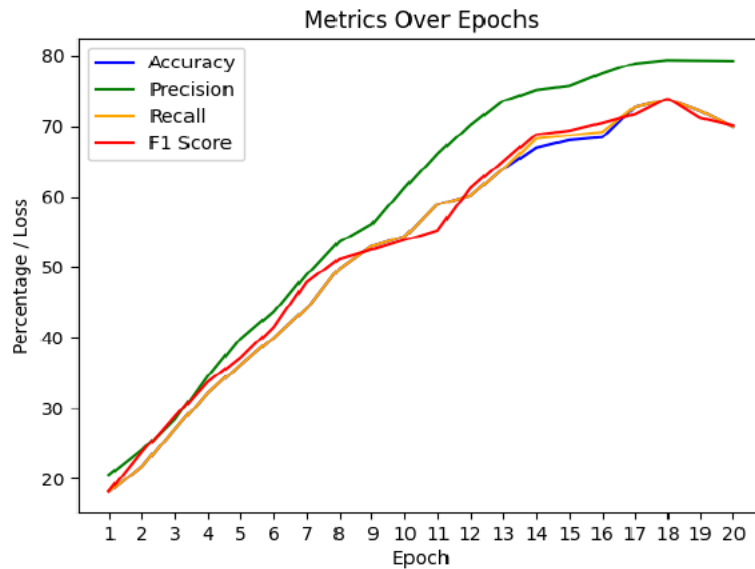


Imagen 72: Gráfica todas las métricas CNN sin preentrenamiento

Para ver el desempeño final hemos creado esta tabla a modo de ficha técnica donde podemos ver de forma clara las métricas explicadas anteriormente.

CNN SIN PREENTRENO	
Exactitud	73.75%
Precisión	79.13%
Sensibilidad	73.75%
Puntuación F1	73.61%
Pérdida de la época	1.022
Tiempo lúcido	9.9 horas
Gasto computacional	2.2473 Kw

Tabla 28: Resultados CNN simple

Como podemos ver en las gráficas, el momento del entrenamiento donde el modelo ha contado con mejor rendimiento ha sido en la época número 18 por lo que todos los datos reflejados en la tabla anterior pertenecen a ese momento. Momentos después el

rendimiento ha caído de forma notoria por lo que podemos confirmar que su rendimiento óptimo y su tiempo lúcido ha sido hasta la época 18.

En este caso, esta tabla muestra un resumen de como la red ha actuado en primer lugar sin un modelo preentrenado como soporte. Teniendo unas métricas moderadas y bastante decentes teniendo en cuenta que es una red neuronal básica y sin preentrenado. No obstante, esto hace que su gasto computacional sea bastante elevado debido a su largo periodo de entrenamiento de casi de 10 horas.

❖ RED NEURONAL CONVOLUCIONAL (CNN) CON PREENTRENO

Habiendo visto el potencial de una red neuronal convolucional simple, es el turno de ver que ocurre cuando la respalda un modelo preentrenado como ResNet18.

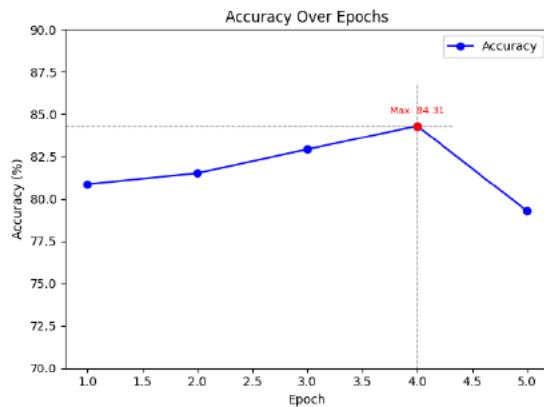


Imagen 73: Gráfica exactitud CNN con preentrenado

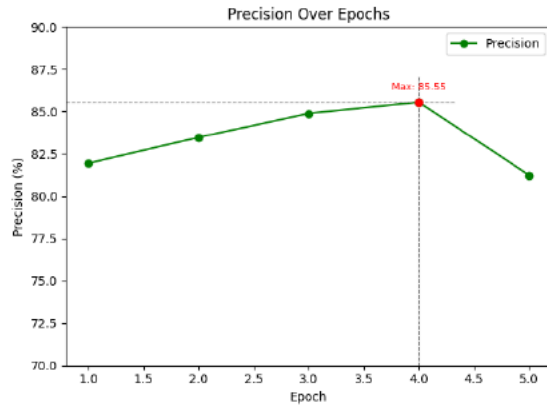


Imagen 74: Gráfica precisión CNN con preentrenado

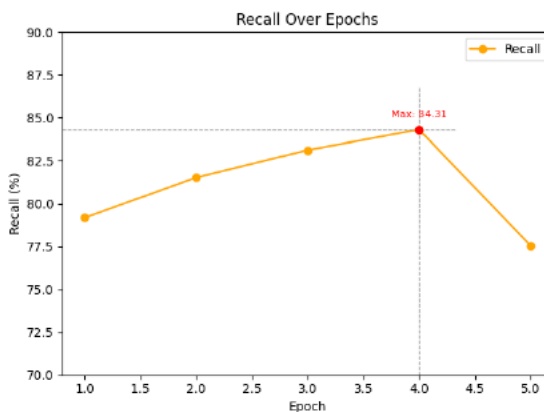


Imagen 75: Gráfica recall CNN con preentrenado

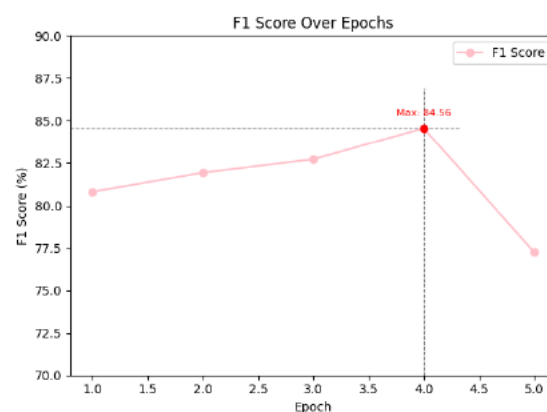


Imagen 76: Gráfica F1 score CNN con preentrenado

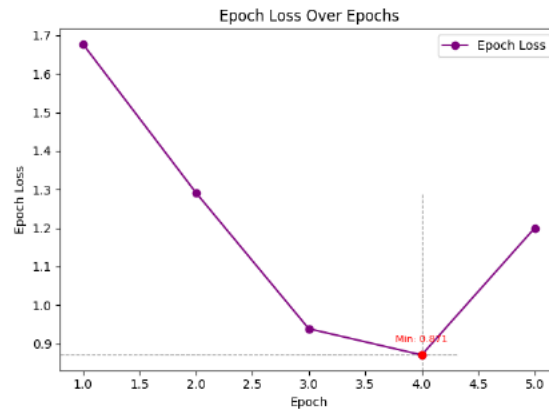


Imagen 77: Gráfica pérdida en épocas CNN con preentrenado

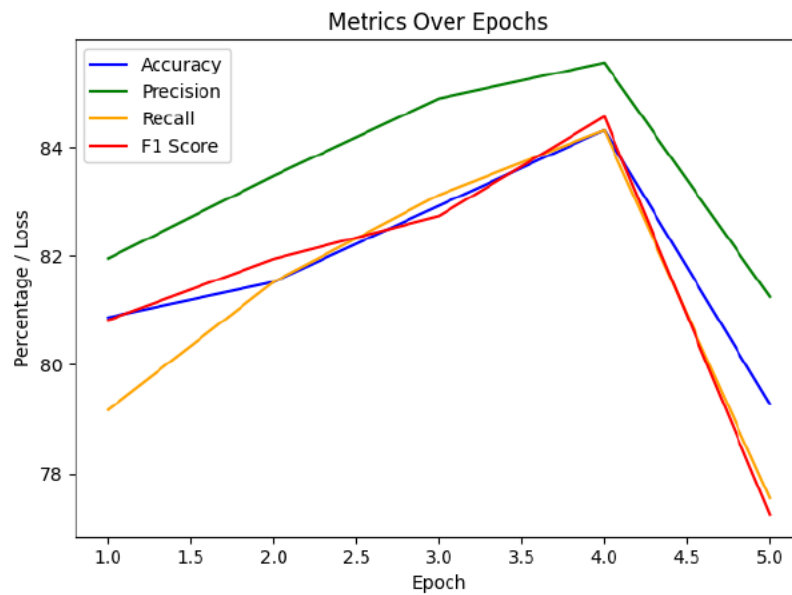


Imagen 78: Gráfica todas las métricas CNN con preentrenado

CNN CON PREENTRENO	
Exactitud	84.31%
Precisión	85.55%
Sensibilidad	84.31%
Puntuación F1	84.56%
Pérdida de la época	0.871
Tiempo lúcido	1.66 horas
Gasto computacional	0.378 Kw

Tabla 29: Resultados CNN con ResNet18

Como podemos ver en las gráficas, el momento del entrenamiento donde el modelo ha contado con mejor rendimiento ha sido en la época número 4 por lo que todos los datos

reflejados en la tabla anterior pertenecen a ese momento. Momentos después el rendimiento ha caído de forma empicada por lo que podemos confirmar que su rendimiento óptimo y su tiempo lúcido ha sido hasta la época 4.

En este caso, esta tabla muestra un resumen de como la red ha actuado respaldada por un modelo preentrenado como soporte. Teniendo unas métricas muy buenas, notándose sobre todo la presencia de ResNet18. Además, gracias a este, podemos ver cómo su gasto computacional es mucho menor debido su corto período de entrenamiento ya que solo hemos necesitado un poco más de una hora y media para demostrar la potencia de un modelo como ResNet.

❖ CNN SIN PREENTRENO VS CNN CON PREENTRENO

Tras analizar el rendimiento de la red neuronal convolucional de ambas formas por separado, es hora de compararlas para analizar como de diferencial ha sido el respaldo que ResNet18 ha ofrecido.

	CNN SIN PREENTRENO	CNN CON PREENTRENO	DIFERENCIA
Exactitud	73.75%	84.31%	+ 10.56 %
Precisión	79.13%	85.55%	+ 6.42%
Sensibilidad	73.75%	84.31%	+ 10.56%
Puntuación	73.61%	84.56%	+ 10.95%
Pérdida de la época	1.022	0.871	- 0.151 (- 14.78%)
Tiempo lúcido	9.9 horas	1.66 horas	- 8.24 horas (- 83.23%)
Gasto computacional	2.2473 Kw	0.378 Kw	- 1.8693 Kw (- 83.18%)

Tabla 30: Tabla comparativa CNN

Todos los análisis apuntan a una gran mejora del modelo gracias a ResNet18. No destacan principalmente las mejoras en la certeza de las respuestas dadas por el propio modelo, aunque son del 10% más fiables, sino que destaca por la disminución drástica del gasto computacional (83.18% menos), debiéndose esto a la disminución abrupta del tiempo lúcido que necesita el modelo con el respaldo de ResNet18.

Así pues, debido al respaldo de ResNet la red neuronal ha necesitado muchas menos épocas para lograr un resultado superior a esa misma red sin el respaldo, lo que ha significado un gasto computacional mucho menor para obtener un resultado satisfactorio.

B) ESTUDIO DEL RENDIMIENTO DEL VISION TRANSFORMER (ViT)

❖ VISION TRANSFORMER (ViT) SIN PREENTRENO

De la misma forma que hemos actuado con la CNN, es hora de ver el desempeño que el ViT ha tenido a la hora de entrenar.

Una de las mejores formas de ver cómo evoluciona un modelo a lo largo de su entrenamiento es viendo sus resultados de forma progresiva a lo largo de las épocas en una gráfica, por lo

que para cada una de las métricas anteriores (excepto para el tiempo lúcido y el gasto computacional que requieren de tiempo en segundos) hemos creado su correspondiente gráfica.

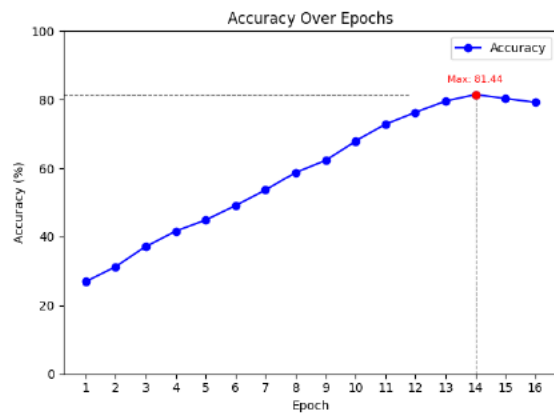


Imagen 79: Gráfica exactitud VIT sin preentrenamiento

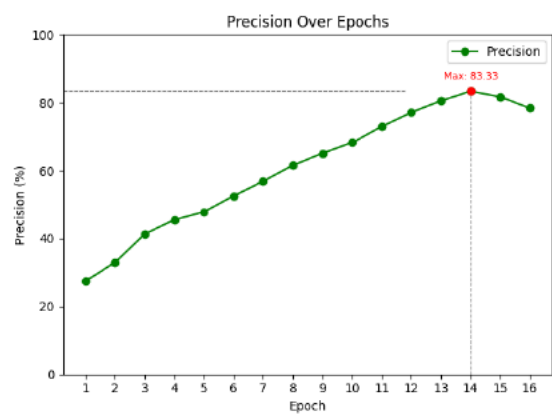


Imagen 80: Gráfica precisión VIT sin preentrenamiento

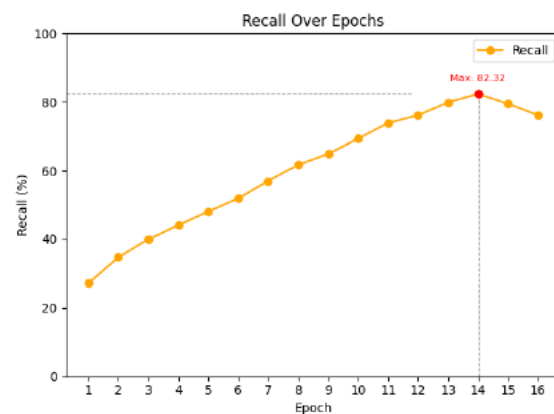


Imagen 81: Gráfica recall VIT sin preentrenamiento

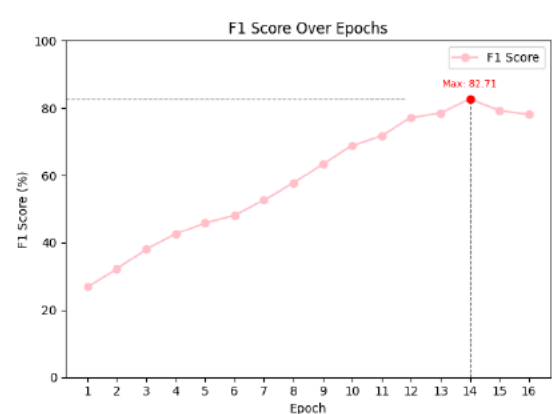


Imagen 82: Gráfica F1 Score VIT sin preentrenamiento

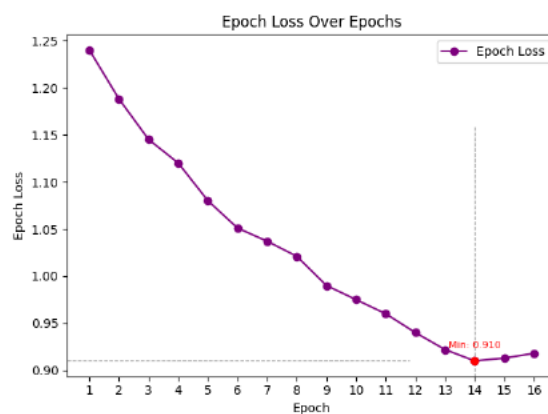


Imagen 83: Gráfica pérdidas en épocas VIT sin preentrenamiento

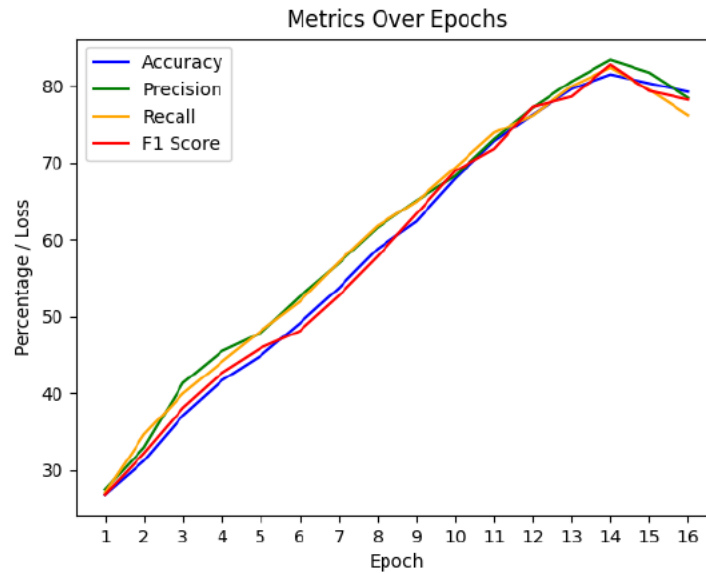


Imagen 84: Gráfica todas las métricas ViT sin preentrenamiento

Para ver el desempeño final hemos creado esta tabla a modo de ficha técnica donde podemos ver de forma clara las métricas explicadas anteriormente.

ViT SIMPLE	
Exactitud	81.44%
Precisión	83.33%
Sensibilidad	82.32%
Puntuación F1	82.71%
Pérdida de la época	0.910
Tiempo lúcido	5.23 horas
Gasto computacional	1.18721 Kw

Como podemos ver en las gráficas, el momento del entrenamiento donde el modelo ha contado con mejor rendimiento ha sido en la época número 14 por lo que todos los datos reflejados en la tabla anterior pertenecen a ese momento. Momentos después el rendimiento ha caído de forma notoria por lo que podemos confirmar que su rendimiento óptimo y su tiempo lúcido ha sido hasta la época 14.

En este caso, esta tabla muestra un resumen de como el ViT ha actuado en primer lugar sin un modelo preentrenado como soporte. Teniendo unas métricas notorias, superando el 80% de fiabilidad en todos los aspectos. Resultado que además ha sido obtenido en un tiempo bastante aceptable teniendo en cuenta que no cuenta con ningún modelo a sus espaldas.

❖ VISION TRANSFORMER (ViT) CON PREENTRENO

Habiendo visto el potencial de un ViT simple, es el turno de ver que ocurre cuando ese ViT cuenta con la ayuda de un potente modelo preentrenado de Hugging Face.

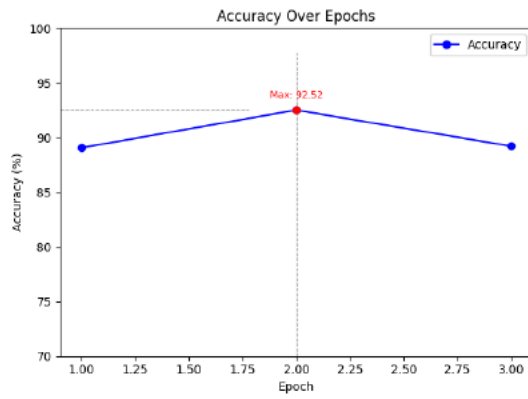


Imagen 85: Gráfica exactitud VIT con preentrenno

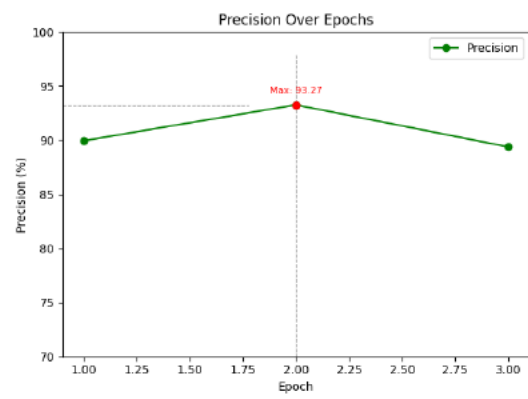


Imagen 86: Gráfica precisión VIT con preentrenno

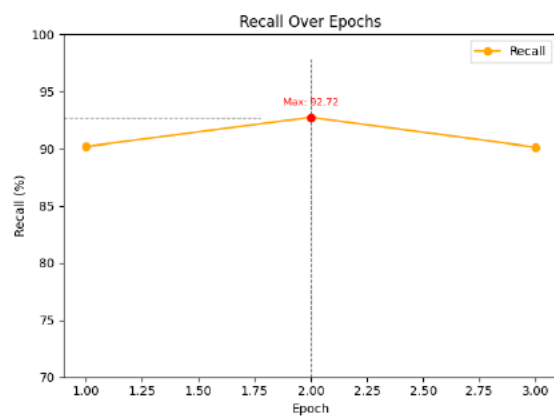


Imagen 87: Gráfica Recall VIT con preentrenno

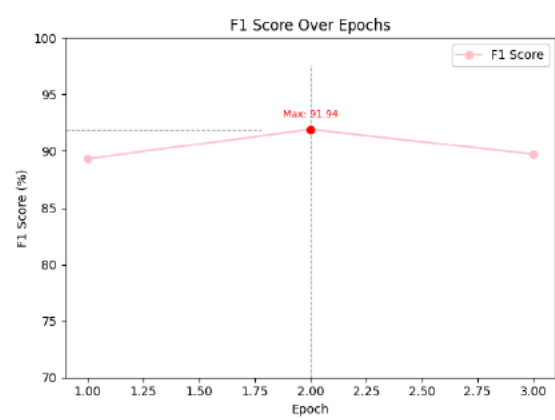


Imagen 88: Gráfica F1 Score VIT con preentrenno

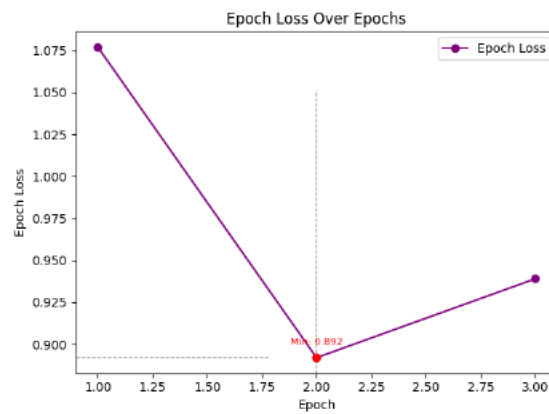


Imagen 89: Gráfica pérdidas en épocas VIT con preentrenno

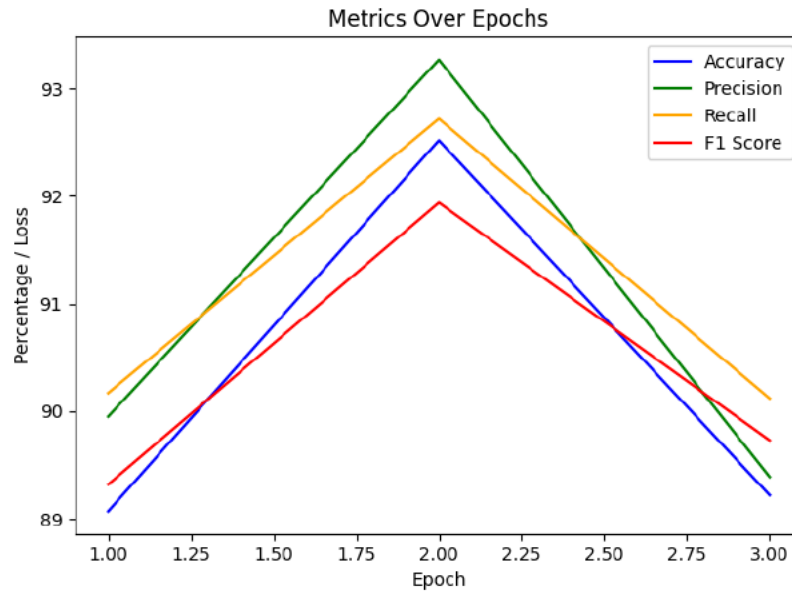


Imagen 90: Gráfica todas las métricas ViT con preentrenamiento

ViT CON HUGGING FACE	
Exactitud	92.52%
Precisión	93.27%
Sensibilidad	92.72%
Puntuación F1	91.94%
Pérdida de la época	0.892
Tiempo lúcido	2.86 horas
Gasto computacional	0.64922 Kw

Como podemos ver en las gráficas, el momento del entrenamiento donde el modelo ha contado con mejor rendimiento ha sido en la época número 2 por lo que todos los datos reflejados en la tabla anterior pertenecen a ese momento. Momentos después el rendimiento ha caído de forma empicada por lo que podemos confirmar que su rendimiento óptimo y su tiempo lúcido ha sido hasta la época 2.

En este caso, esta tabla muestra un resumen de como el ViT ha actuado respaldado por un modelo preentrenado como soporte. Teniendo unas métricas muy buenas, notándose sobre todo la presencia del modelo de Hugging Face, tanto para aspectos positivos como negativos. En los resultados podemos ver que son excelentes, superando el umbral del 90% de fiabilidad en todos los aspectos, no obstante, debido a la gran cantidad de datos con los que cuenta el modelo (86.6M de parámetros entrenables) las épocas, aunque son pocas, son muy pesadas de entrenar lo que han mermado en cierta parte el tiempo lúcido de la tabla superior.

❖ ViT SIMPLE VS ViT CON HUGGING FACE

Tras analizar el rendimiento del enfoque Vision Tranformer de ambas formas por separado, es hora de compararlas para analizar como de diferencial ha sido el respaldo que el modelo de Hugging Face ha ofrecido.

	VIT SIMPLE	VIT CON HUGGING FACE	DIFERENCIA
Exactitud	81.44%	92.52%	+ 11.08 %
Precisión	83.33%	93.27%	+ 9.94%
Sensibilidad	82.32%	92.72%	+ 10.40%
Puntuación	82.71%	91.94%	+ 9.23%
Pérdida de la época	0.910	0.892	- 0.018 (- 1.98%)
Tiempo lúcido	5.23 horas	2.86 horas	- 2.37 horas (- 45.32%)
Gasto computacional	1.18721 Kw	0.64922 Kw	- 0.53799 Kw (- 45.30%)

Tabla 31: Comparativa ViT

Usando la tabla comparativa superior podemos apreciar una gran mejora en los resultados cuando el modelo preentrenado está presente aumentando el rendimiento general en un 10% y sobre todo disminuyendo el tiempo lúcido y el gasto computacional en casi un 50% lo que es algo muy positivo.

Así pues, podemos afirmar que el rendimiento que tiene el enfoque Vision Transformer para estos casos es estupendo y muestra grandes resultados y que el apoyo de un modelo preentrenado en estos casos hace que el proceso se simplifique mucho y todo sea relativamente mucho más rápido.

C) CONCLUSIONES TRAS EL ESTUDIO

Después de una meticulosa evaluación de diversas arquitecturas de inteligencia artificial para tareas de visión, las conclusiones extraídas arrojan luz sobre las ventajas y desventajas de los modelos considerados, destacando la importancia de la arquitectura y el preentrenamiento en el rendimiento general.

En la retaguardia de nuestro análisis, **la CNN sin preentrenamiento** reveló un rendimiento modesto del 70%, a pesar de la inversión de casi 10 horas en su entrenamiento. Este resultado pone de manifiesto la influencia significativa del preentrenamiento en las CNN, resaltando la necesidad de transferencia de conocimientos previos para optimizar el rendimiento en tareas específicas.

El Vision Transformer (ViT) sin preentrenamiento ocupó el tercer lugar con un rendimiento del 82%, indicando mejoras respecto a la CNN sin preentrenamiento. No obstante, este resultado subraya la importancia del preentrenamiento en los modelos basados en Transformer para maximizar la capacidad de generalización.

Avanzando en la clasificación, **la CNN con respaldo de ResNet18 se situó en el segundo puesto**, alcanzando un rendimiento del 85% en tan solo una hora y media de entrenamiento. La integración de la arquitectura ResNet18 demostró ser esencial para mejorar la eficiencia del aprendizaje de características, equilibrando rendimiento y tiempo de entrenamiento.

El Vision Transformer con preentrenamiento destacó como el claro vencedor con un rendimiento excepcional del 92% en aproximadamente 3 horas de entrenamiento. La eficacia de esta arquitectura respalda la noción de que los modelos basados en Transformer, especialmente cuando se combinan con preentrenamiento, superan a las arquitecturas convencionales en tareas de visión.

A pesar de los notables logros del ViT, en AvIAry reconocemos que la elección de la arquitectura también debe considerar aspectos ambientales y costes de mantenimiento en la nube. La arquitectura del ViT, con sus capacidades superiores en la tarea de visión, se debe en parte a su diseño innovador basado en Transformer. No obstante, **en línea con nuestra misión de sostenibilidad, hemos optado por la CNN con preentrenamiento**, equilibrando un rendimiento aceptable (alrededor del 85%) con un tiempo de entrenamiento más eficiente (aproximadamente una hora y media) y un peso en la nube mucho menos costoso de mantener. **Esta elección**, aunque sacrifica algunos puntos porcentuales en rendimiento, **se alinea con nuestro compromiso ambiental**, demostrando que en AvIAry, la toma de decisiones en inteligencia artificial va más allá del rendimiento puro y considera el impacto ecológico, **fomentando así un equilibrio sostenible entre eficacia y responsabilidad ambiental**.

6.2.4 SUBIDA DE LA ARQUITECTURA A LA NUBE

Para poder subir la arquitectura y toda su lógica a la nube e implementarla con el front-end fue necesario crear un servidor Flask que funcionase como un endpoint para hacer las peticiones necesarias, en este caso peticiones POST, cada vez que el usuario solicite su uso.

```
# Ruta para cargar una imagen y obtener la clasificación
@app.route('/upload', methods=['POST'])
def upload():
    if 'image' not in request.files:
        return jsonify({'error': 'No se recibió una imagen válida'})

    file = request.files['image']
    basepath = os.path.dirname(__file__)
    filename = secure_filename(file.filename)
    upload_path = os.path.join(basepath, 'static/archivos', filename)
    file.save(upload_path)

    image = load_and_preprocess_image(os.path.join('static/archivos', filename))
    resultados = classify_image(image, model, classes)

    return jsonify(resultados)
```

Imagen 91: Método POST para las predicciones de la IA

Para subir el propio servidor Flask, usamos los servicios de Render, una plataforma que permite subir servidores a la nube de forma gratuita, pero con limitaciones de rendimiento.

Así pues, ahora contamos con un front-end con el que el usuario puede interactuar y un servidor que aloja a una inteligencia artificial capaz de clasificar con éxito una imagen de un pájaro que el usuario le solicite.

6.3 BACK-END: EL ALMACÉN DE INFORMACIÓN DE AVIARY

Con la interfaz y la inteligencia artificial listos, el siguiente paso es contar una base de datos lo suficientemente grande como para, en un principio, almacenar la información de las 400 especies más comunes del mundo. En AvIAry creemos fielmente en la credibilidad de los datos por lo que todos los datos que se encuentran en nuestro poder son reales y han sido corroborados contrastando con diferentes fuentes especializadas en el ámbito aviar.

6.3.1 ESTRUCTURA DE LOS DATOS DE AVIARY

La forma más sencilla de organizar los datos es en un archivo. JSON. Este nos permite ver de forma clara cuales son las propiedades de cada objeto creado, en este caso, para cada especie, la estructura de los datos es la siguiente.

```
_id: ObjectId('653640ed3b3488e5b19a56e4')
id: 1
name: "EMU"
imagen: "https://imgs.search.brave.com/TAEFpKttvfWsEugGb6brDPvrG20LpVTnu5qP8Nx8..."
habitat: "Australia"
esperanza_de_vida: "10-20 years"
colores: "Brown"
tamaño: "1.5-2 meters"
peso: "30-45 kg"
alimentación: "Herbivore"
```

Imagen 92: Formato del cada Objeto en el archivo .JSON

Donde podemos ver los principales atributos relevantes con los que una especie puede contar, como, por ejemplo, su **hábitat, esperanza de vida, colores, tamaño, peso y alimentación**.

Además, podemos destacar el propio nombre de la especie, así como un id numérico diferente a cada especie para poder tratar los datos de forma sencilla. No obstante, como se puede observar, también contamos con un id formado por letras y números creado por la propia base de datos para gestiones internas dentro de la plataforma.

6.3.2 ECOSISTEMA MONGODB

Como hemos hablado previamente en el apartado 5.5 Razones detrás de las decisiones tecnológicas, la elección de MongoDB y su modularidad NoSQL se basan en la flexibilidad y la escalabilidad que esta plataforma ofrece ya que esta permite un esquema dinámico, proporcionando flexibilidad sin comprometer el rendimiento.

Además, era necesario manejar grandes volúmenes de datos teniendo un rendimiento constante, donde MongoDB es una gran elección. Así pues, con el objetivo de simplificar el proceso lo máximo posible hemos contado con el ecosistema de MongoDB, donde destacamos el uso de MongoDB Compass como una interfaz gráfica para la visualización de datos a través de colecciones y MongoDB Atlas para la subida de datos dentro del propio ecosistema.

El proceso al estar todo muy bien conectado, es bastante sencillo ya que solo fue necesario importar el archivo .JSON a MongoDB Compass y posteriormente vincularlo con MongoDB Atlas.

En nuestra base de datos contamos con la información de 400 especies donde cada una cuenta con todos los atributos destacados anteriormente, lo que hace que AvIary cuente con un contenido rico en información variada y accesible para personas que no cuenten con ningún tipo de conocimiento sobre el mundo aviar.

6.3.3 CREACIÓN DE UNA API REST

Para completar el ciclo de información desde que el usuario manda la imagen, el servidor dictamina que especie es y el usuario recibe la información completa de la especie, es necesaria la creación de un lugar donde se encuentren alojados los datos y el sistema a través de una petición pueda recuperarlos sin problemas. A partir de este dilema nace la necesidad de crear una API con la información de los pájaros para que el flujo de datos se complete de forma exitosa y rápida.

Para la creación de dicha API REST hemos hecho uso de Express, un marco de aplicación web de Node.js que ofrece un entorno de ejecución de JavaScript del lado de servidor.

Los datos alojados hasta ahora en MongoDB, son vinculados a esta API gracias a la conectividad con la que nuestra base de datos cuenta.

```
const connection_url =
  "mongodb+srv://example:12345@birds.3qoysym.mongodb.net/Birds-Info?retryWrites=true&w=majority";

// Conecta a la base de datos y a la colección
mongoose.connect(connection_url, {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});

mongoose.connection.on("error", err => {
  console.error("Error de conexión a la base de datos:", err);
});
```

Imagen 93: Vinculación de MongoDB con la API

Con la base de datos ya vinculada solo queda crear un punto de flujo de datos gracias a una URL para obtener la información de las especies y el trabajo estaría completo.

```
app.get("/birds/get", async (req, res) => {
  try {
    const data = await Bird.find().exec();
    res.status(200).send(data);
  } catch (err) {
    res.status(500).send(err);
  }
});

app.listen(port, () => {
  console.log(`Servidor Express escuchando en el puerto ${port}`);
});
```

Imagen 94: Servidor Express con la API

6.3.4 SUBIDA DE LOS DATOS A LA NUBE

Para subir esta API a la nube y que la aplicación pueda rescatar la información en cualquier momento de forma rápida y efectiva, hemos usado Render, al igual que con la inteligencia artificial ya que nos permite subir la API creada en Express de forma sencilla y gratuita. La API es pública y de uso gratuita en la URL <https://aviary-backend.onrender.com> donde si viajamos hasta el endpoint <https://aviary-backend.onrender.com/birds/get> podemos obtener todos los datos de cada especie.

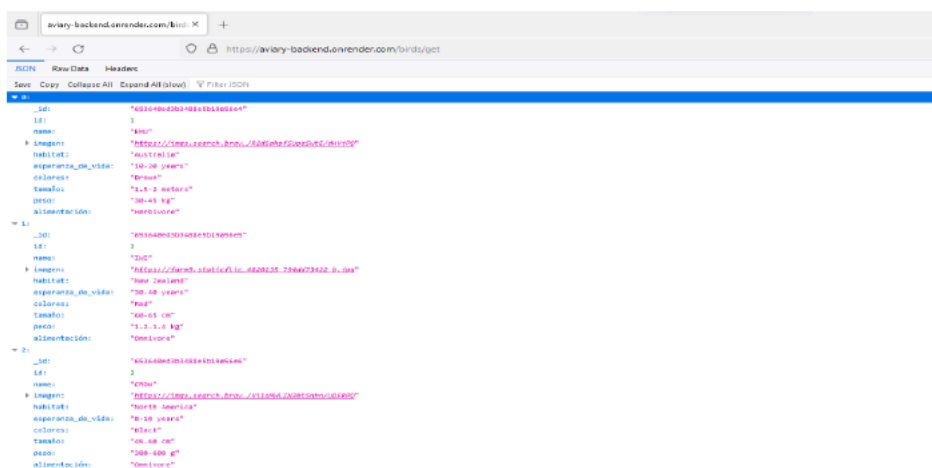


Imagen 95: Resultado de la API REST en línea

6.4 ¿QUÉ ES REALMENTE AVIARY? ¿UNA APP O UNA WEB?

Aviary se define como una **Progressive Web App (PWA)**, una categoría de aplicaciones que combina lo mejor de las aplicaciones web y las aplicaciones móviles tradicionales. A diferencia de una aplicación web estándar, una PWA ofrece una experiencia más avanzada y cercana a la de una aplicación nativa, con la ventaja adicional de ser accesible a través de navegadores web.

¿Qué hace que Aviary sea una PWA?

- ❖ **Accesibilidad universal:** Aviary es accesible desde cualquier navegador web, eliminando la necesidad de descargas e instalaciones. Los usuarios pueden acceder a la aplicación simplemente a través de la URL, facilitando la difusión y la adopción.
- ❖ **Experiencia de usuario inmersiva:** Aviary ofrece una experiencia de usuario inmersiva similar a la de una aplicación nativa. Permite a los usuarios añadir un acceso directo a la pantalla de inicio de su dispositivo, brindando una apariencia y funcionalidad similares a las aplicaciones tradicionales.
- ❖ **Actualizaciones automáticas:** Las PWAs se actualizan automáticamente, lo que significa que los usuarios siempre disfrutan de las últimas mejoras y características sin la necesidad de intervención manual. Esto garantiza que Aviary esté constantemente optimizada y enriquecida con nuevas funcionalidades.
- ❖ **Seguridad mejorada:** Al ser servida a través de HTTPS, Aviary ofrece una capa adicional de seguridad, protegiendo la información sensible y proporcionando un entorno seguro para los usuarios.

¿Por qué usar una PWA para Aviary?

La elección de Aviary como PWA se basa en varios beneficios:

- ❖ **Accesibilidad inmediata:** Al no requerir descargas ni instalaciones, Aviary se vuelve accesible instantáneamente a través de navegadores web, eliminando barreras para la adopción.

- ❖ **Experiencia del usuario continua:** La capacidad de funcionar sin conexión y la experiencia de usuario similar a una aplicación nativa garantizan que los usuarios disfruten de un servicio continuo, independientemente de las limitaciones de conectividad.
- ❖ **Eficiencia en el desarrollo:** La implementación de una PWA simplifica el proceso de desarrollo, ya que una única base de código es compatible con múltiples plataformas y dispositivos, reduciendo los costos y el tiempo de desarrollo.
- ❖ **Actualizaciones transparentes:** Las actualizaciones automáticas aseguran que los usuarios siempre tengan acceso a las últimas características y mejoras sin esfuerzo adicional.
- ❖ **Versatilidad y alcance global:** Al ser accesible a través de navegadores, AvIary puede llegar a una audiencia global sin restricciones asociadas con las tiendas de aplicaciones específicas de cada plataforma.

¿Cómo es el proceso de instalación?

Para instalar la aplicación de AvIary es bastante sencillo, debes seguir los siguientes pasos:

Si estás en un ordenador debes dirigirte a www.aviary-app.me y una vez allí tu navegador te avisará de forma sutil que puedes instalarlo, puedes ir al menú lateral de tu navegador que aparecerá la opción de 'Instalar AvIary'.

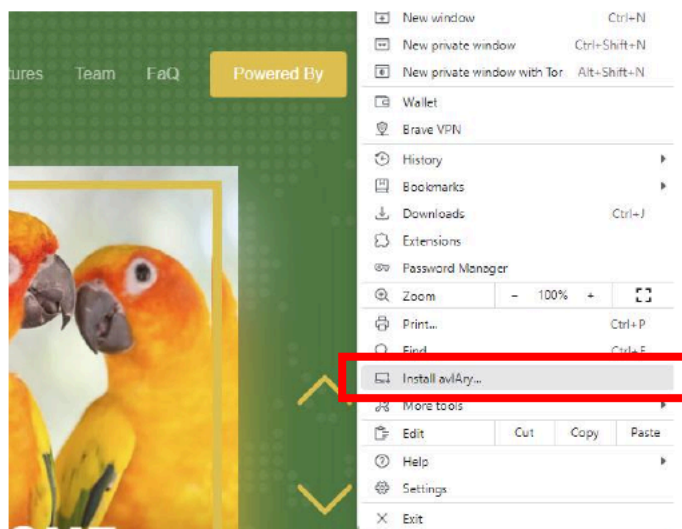


Imagen 96: Descarga de AvIary en la web

Así pues, después de haberlo instalado debe aparecer un icono en el Escritorio perteneciente a la aplicación, y si lo ejecutas, AvIary se abrirá y funcionará perfectamente.

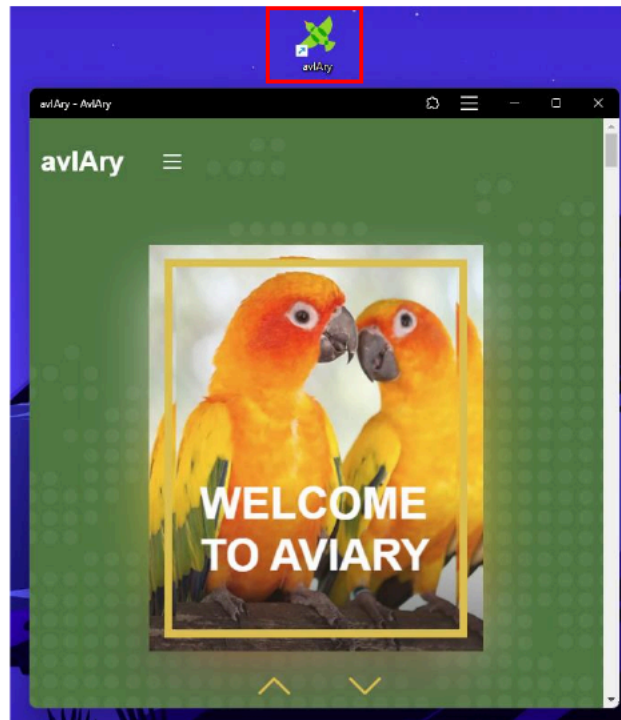


Imagen 97: Aplicación e icono en el escritorio

Si en caso contrario estamos en un dispositivo móvil, en este caso en iPhone, debemos ir igualmente a www.aviary-app.me y en el navegador buscar la opción 'Añadir a la pantalla de inicio'. Esta aparecerá como una aplicación cualquiera instalada mediante cualquier tienda de aplicaciones, solo debemos pulsar en el icono de la app y funcionará perfectamente.

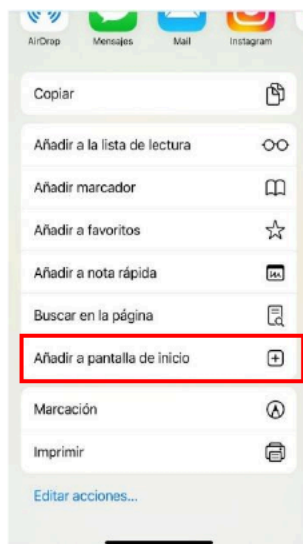


Imagen 98: Descargar AvlAry en móviles

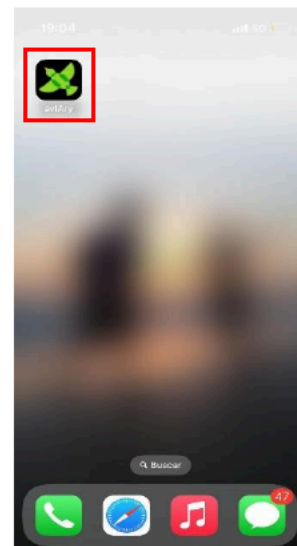


Imagen 99: AvlAry como una aplicación

7. PRUEBAS DEL SISTEMA

Para comprobar que el diseño de la arquitectura completa del sistema ha sido implementado de forma correcta y todo funciona como ha sido previsto es necesario contar con pruebas en todos los ámbitos posibles para asegurar que el producto final satisface el

umbral de calidad deseado. En este caso, en AvIAry queremos que los usuarios cuenten con una aplicación sin errores imprevistos y asegurada de la mejor forma posible siguiendo con el estándar de la industria, por lo que las diferentes pruebas que se han realizado son las siguientes:

7.1 PRUEBAS UNITARIAS CON JEST

Jest se ha convertido en una herramienta esencial para realizar pruebas unitarias en el código. Estas pruebas se centran en evaluar cada unidad individual del código, asegurando su correcto funcionamiento de manera aislada. Jest permite realizar aserciones, mocks y snapshots, proporcionando así una base sólida para la integridad de las funciones y métodos desarrollados.

```
test('handleSendImage sends Image and updates state', async () => {
  const { getByLabelText, getByText } = render(<Ia />);
  const input = getByLabelText('Select Image');
  const sendButton = getByText('Send Image');

  const file = new File(['r-c-o'], 'test.jpg', { type: 'image/jpeg' });

  fireEvent.change(input, { target: { files: [file] } });
  fireEvent.click(sendButton);

  mockAxios.onPost('https://aviary-ia.onrender.com/upload').reply(200, [
    { class: 'Bird', confidence: 0.85 },
  ]);

  await waitFor(() => {
    expect(getByText('Classification analysis')).toBeInTheDocument();
    expect(getByText('Specie: Bird')).toBeInTheDocument();
    expect(getByText('Confidence: 85.00%')).toBeInTheDocument();
  });
});
```

Imagen 100: Prueba Unitaria

7.2 PRUEBAS DE INTERFAZ DE USUARIO (UI)

La experiencia del usuario es un factor crítico en el éxito de cualquier aplicación web. La realización de **reuniones con usuarios testers** proporcionaron una visión valiosa sobre la usabilidad del sistema. Estas pruebas permitieron evaluar la interfaz de usuario, identificar problemas de navegación y obtener comentarios directos de los usuarios finales, contribuyendo así a la mejora continua del diseño.

7.3 PRUEBAS DE NAVEGADOR CRUZADO

Garantizar que la aplicación funcione de manera consistente en diferentes navegadores y dispositivos es fundamental. **BrowserStack** facilita las pruebas de navegador cruzado al ofrecer una plataforma en la nube que permite ejecutar pruebas en una amplia variedad de navegadores y versiones. Esto asegura que la aplicación sea accesible y funcional para todos los usuarios, independientemente del entorno de navegación que utilicen.

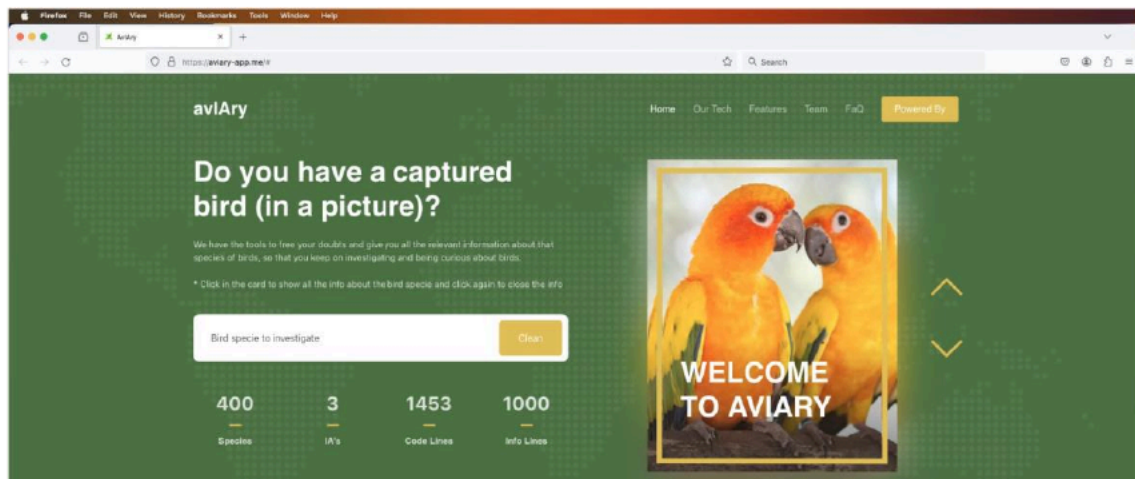


Imagen 101: Prueba de AvlAry en Safari

7.3 PRUEBAS DE VELOCIDAD CON WEBPAGETEST.COM

La velocidad de carga de una aplicación web es crucial para retener a los usuarios y mejorar el posicionamiento en los motores de búsqueda. **Webpagetest.com** proporciona métricas detalladas sobre el rendimiento de la página, incluyendo el tiempo de carga, el tiempo de renderizado y otros factores clave. Estas pruebas ayudan a identificar cuellos de botella y optimizar la velocidad de la aplicación.

Los números que reflejan el resultado del rendimiento son bastante buenos y óptimos para una web como la nuestra, destacan un atributo poco común y del que estamos muy orgullosos es la **huella de carbono de nuestra página** siendo esta muy por debajo de la media.

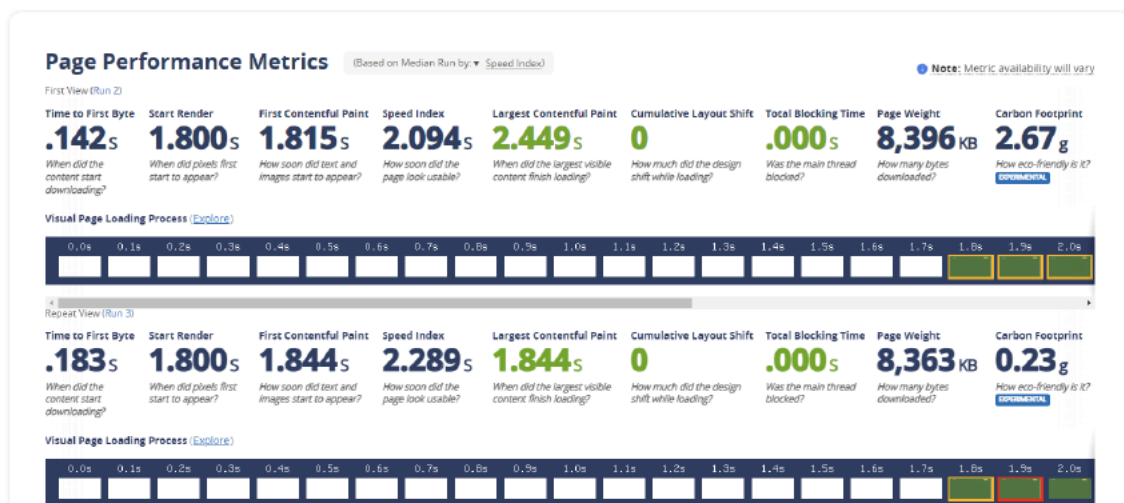


Imagen 102: Métricas de la aplicación

7.4 PRUEBAS DE RENDIMIENTO CON K6 Y GRAFANA LABS

Para evaluar cómo la aplicación maneja cargas de trabajo intensivas, se llevaron a cabo diferentes pruebas de rendimiento. **K6**, una herramienta de prueba de carga de código abierto se utilizó junto con **Grafana Labs** para visualizar y analizar el rendimiento en tiempo real. Estas pruebas permiten identificar posibles problemas de escalabilidad y

garantizar que la aplicación pueda manejar un volumen significativo de usuarios concurrentes.

❖ PRUEBA DE ESTRÉS

Simula condiciones extremas de carga para evaluar el rendimiento y la capacidad de recuperación de la aplicación.

En este caso, el estrés al que ha sido sometido ha sido suficiente para que el servidor de Netlify deje de emitir peticiones por seguridad debido a la alta demanda gracias a la prueba.

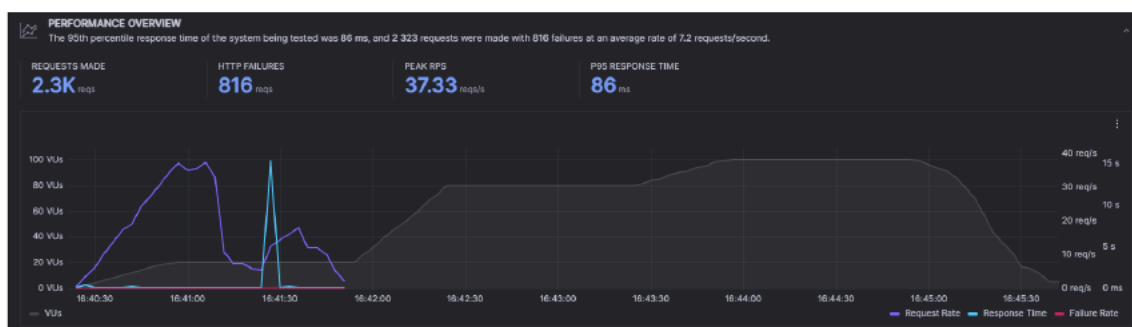


Imagen 103: Gráfica de prueba de estrés

❖ PRUEBA DE SOAK

Evalúa el rendimiento de la aplicación durante un período prolongado bajo carga constante para identificar posibles problemas de memoria o fugas de recursos.

Durante el período de tiempo que el sistema ha sido puesto a prueba, vemos como ha sido capaz de mantener el rendimiento de forma notable sin fallos y de forma estable.

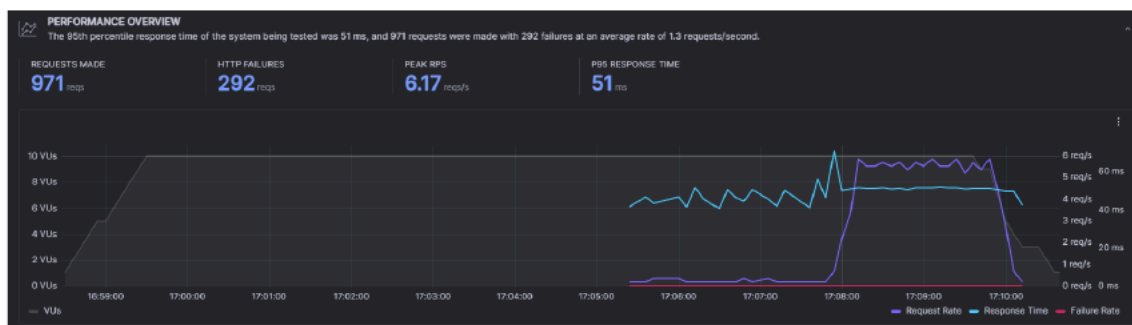


Imagen 104: Gráfica de prueba de soak

❖ PRUEBA DE SPIKE

Evalúa la capacidad de la aplicación para manejar picos repentinos de tráfico.

Destacamos la alta capacidad para mantenerse firme pese a altos picos de tráfico, no obstante, podemos destacar dos picos de escasez de rendimiento debidos a varias peticiones que tardaron en procesarse y entregarse.

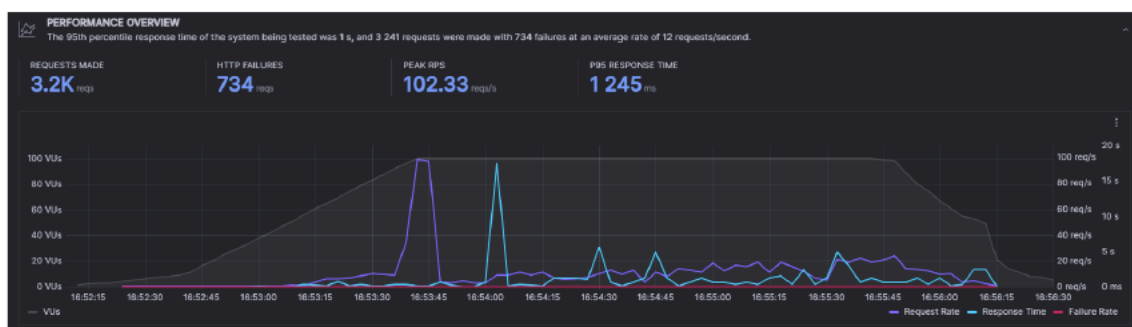


Imagen 105: Gráfica de prueba de spike

7.5 REPORTE GENERAL CON LIGHTHOUSE DE GOOGLE

Lighthouse de Google proporciona una evaluación integral del rendimiento, accesibilidad, mejores prácticas, SEO y más. Este conjunto de pruebas automatizadas permite obtener un informe detallado sobre la salud general de la aplicación. En este caso, AvIAry fue evaluada por la herramienta y su puntuación fue la siguiente:

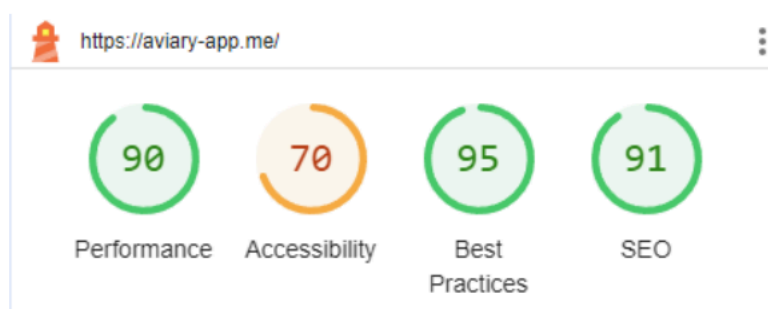


Imagen 106: Resultados en Light House

❖ RENDIMIENTO IMPRESIONANTE

Nuestra plataforma ha alcanzado una puntuación destacada del 90 en el aspecto de rendimiento de Lighthouse. Este resultado resalta la eficacia de nuestras prácticas de desarrollo para garantizar tiempos de carga rápidos y una experiencia de usuario fluida. Estamos comprometidos a ofrecer un rendimiento excepcional para que los usuarios disfruten de una navegación ágil y eficiente.

❖ ACCESIBILIDAD EN EVOLUCIÓN

Reconocemos la importancia de la accesibilidad web y, aunque hemos obtenido una puntuación del 70, estamos comprometidos a mejorar continuamente. Estamos trabajando activamente para hacer que nuestra plataforma sea más accesible para todos los usuarios, asegurándonos de que la información crítica sea fácilmente accesible y comprensible. Nuestro objetivo es garantizar que cada visitante, independientemente de sus capacidades, tenga una experiencia positiva en nuestra plataforma.

❖ ADHERENCIA A LAS MEJORES PRÁCTICAS

La puntuación del 95 en mejores prácticas destaca nuestro compromiso con los estándares de desarrollo web. Estamos implementando las últimas y más efectivas prácticas

recomendadas para asegurar la integridad y la calidad de nuestro código. Esta puntuación refleja nuestra dedicación a seguir las pautas de la industria y adoptar prácticas que mejoren la seguridad y el rendimiento de nuestra plataforma.

❖ OPTIMIZACIÓN PARA MOTORES DE BÚSQUEDA

La optimización para motores de búsqueda (SEO) es fundamental para la visibilidad en línea, y nuestra puntuación del 91 en este aspecto demuestra nuestra atención a este aspecto crítico. Estamos implementando estrategias sólidas de SEO para asegurar que nuestra plataforma sea fácilmente descubrible y clasificada de manera efectiva en los resultados de búsqueda.

C) PLANIFICACIÓN DEL PROYECTO

8. PLANIFICACIÓN INICIAL

8.1 PLANIFICACIÓN TEMPORAL

8.1.1 FASE 1: DISEÑO Y PLANIFICACIÓN

A) Definición de requisitos (17h)

- ❖ Detallar los requisitos específicos del proyecto.
- ❖ Identificar las funcionalidades clave de la aplicación web y la IA.

B) Diseño de la interfaz de usuario (UI) (15h)

- ❖ Crear bocetos y wireframes del Front-end utilizando React y Bootstrap.

C) Diseño de la arquitectura del sistema (15h)

- ❖ Planificar la estructura del servidor Flask y la integración con el método de IA en PyTorch.
- ❖ Definir la estructura de la base de datos MongoDB y la API de Express.

8.1.2 FASE 2: DESARROLLO DEL FRONT-END

A) Configuración del entorno de desarrollo (1h)

- ❖ Instalar y configurar las herramientas necesarias para el desarrollo con React y Bootstrap.

B) Desarrollo de componentes Front-end (55h)

- ❖ Comenzar a desarrollar las interfaces de usuario según el diseño previo.
- ❖ Implementar la integración con el método de IA para el procesamiento de imágenes.

C) Pruebas iniciales del Front-end (5h)

- ❖ Realizar pruebas unitarias y de integración del Front-end para garantizar el funcionamiento básico.

8.1.3 FASE 3: DESARROLLO DEL SERVIDOR IA

A) Configuración del servidor Flask y PyTorch (10h)

- ❖ Establecer el entorno para el servidor Flask y la integración con el método de IA en PyTorch.

B) Desarrollo de funcionalidades del servidor (10h)

- ❖ Implementar las funciones necesarias para procesar solicitudes de imágenes y devolver resultados de la IA.

C) Integración con Front-end (5h)

- ❖ Configurar la base de datos MongoDB y establece la conexión con el servidor Flask.

8.1.4 FASE 4: DESARROLLO DEL BACK-END

A) Desarrollo de la API con Express (2h)

- ❖ Crear endpoints para acceder a la información de las especies de aves almacenadas en MongoDB.

B) Integración completa del Back-end (15h)

- ❖ Asegurar la integración adecuada entre el servidor Flask, el método de IA y el servidor Express.

C) Pruebas y depuración del Back-end (3h)

- ❖ Realizar pruebas exhaustivas del back-end y resuelve cualquier problema identificado.

8.1.5 FASE 5: INVESTIGACIÓN Y DESARROLLO DE LA IA

A) Investigación comparativa de CNN y Transformer (30h)

- ❖ Realizar un análisis detallado comparando las dos arquitecturas para la detección de especies de aves.

B) Desarrollo del modelo de IA (75h)

- ❖ Implementar y entrenar el modelo de IA utilizando PyTorch, eligiendo la arquitectura más adecuada.

C) Pruebas y ajustes del método de IA (30h)

- ❖ Realizar pruebas con conjuntos de datos y ajusta los hiperparámetros para mejorar el rendimiento.

8.1.6 FASE 6: DESARROLLO DEL BACKEND DE INFORMACIÓN DE ESPECIES

A) Integración de datos de especies (5h)

- ❖ Importar y cargar datos de especies de aves en la base de datos MongoDB.

B) Desarrollo de endpoints de la API de especies (2h)

- ❖ Implementar endpoints que proporcionen información detallada sobre las especies de aves.

8.1.7 FASE 7: PRUEBAS Y OPTIMIZACIÓN

A) Pruebas de integración completa (15h)

- ❖ Realizar pruebas exhaustivas que abarquen todas las partes del sistema.

B) Optimización del rendimiento (10h)

- ❖ Identificar y abordar posibles cuellos de botella en el rendimiento del sistema.

8.1.8 FASE 8: DESPLIEGUE Y DOCUMENTACIÓN

A) Despliegue en un entorno de producción (2h)

- ❖ Configurar el despliegue de la aplicación web y los servicios asociados en un entorno de producción.

B) Creación de documentación (20h)

- ❖ Documentar el código, la arquitectura y la utilización de la aplicación y la IA.

C) Pruebas finales y ajustes menores (3h)

- ❖ Realizar pruebas finales en el entorno de producción y realiza ajustes según sea necesario.

Todo el conjunto completo de fases hace un proyecto con una planificación inicial de cinco meses con un total de 345 horas, desde finales de septiembre de 2023 hasta finales de febrero de 2024, con el objetivo de tener margen más que de sobra por si surgiera un problema de cualquier índole que retrasase el proyecto y su desarrollo programado. Para ver de forma gráfica el desarrollo programado podemos crear un diagrama Gantt.

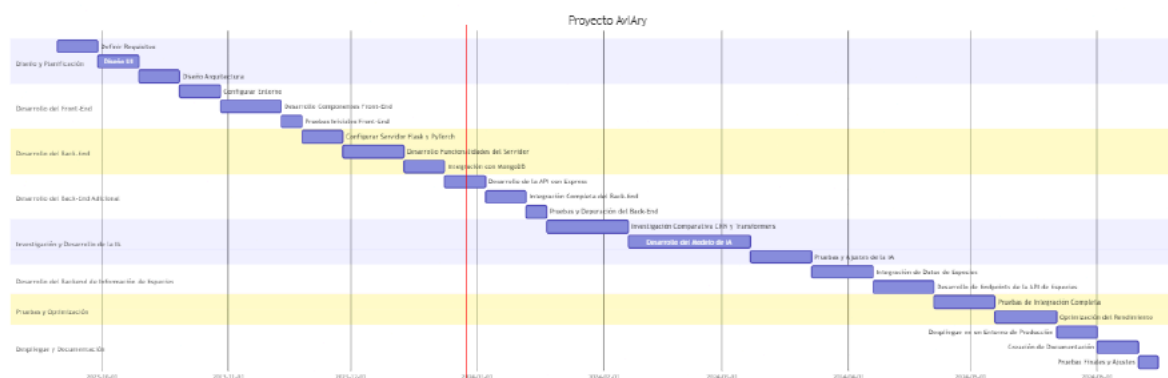


Imagen 107: Diagrama de Gantt inicial

8.2 PLANIFICACIÓN FINANCIERA

Para tener claro cuanto esfuerzo financiero debemos realizar para cumplimentar de forma exitosa este proyecto, debemos tener en cuenta de forma principal diferentes factores:

- ❖ El sueldo de un programador full-stack con conocimientos en deep learning, ya que intervienen tanto Front-end como Back-end e IA. Según el mercado un programador de esta índole, el salario anual bruto es de 35.000€, es decir, sobre unos 25€/h, por lo que esta será la cifra empleada.
- ❖ El número de horas que inicialmente están previstas, en este caso 345 horas.
- ❖ El precio de un ordenador medio con la potencia necesaria para suplir las exigencias de rendimiento es de 950€.
- ❖ Debemos también tener en cuenta si es necesario suplir algún coste de software o de servicio en la nube. En este caso todas las herramientas necesarias son de uso gratuito o contemplan un plan gratuito entre sus planes de precios por lo que no será necesario añadir ningún sobre coste de esta índole.

Con todo ello haciendo un simple cálculo podemos afirmar que el precio inicial bruto del diseño y desarrollo de AvIAry es de:

$$\text{Precio Inicial sin impuestos} = 8625 (345h * 25€/h) + 950 = 9575 \text{ €}$$

Contando ahora con que tributamos en el territorio español en un tramo de impuestos de un 21% que es el estándar, el coste final sería:

$$\text{Precio Inicial Total} = 9575 \text{ €} + 21\% = 11585.75 \text{ €}$$

9. PLANIFICACIÓN FINAL

9.1 PLANIFICACIÓN TEMPORAL

Para la planificación final se ha respetado todo lo planificado al detalle en el apartado 8.1 Planificación Temporal, no obstante, debido a la quiebra de la empresa en la que trabajaba y a la escasez de asignaturas en el primer cuatrimestre, mi cantidad de tiempo libre aumentó de forma considerable, por lo que he sido capaz de reducir en casi un mes y medio el desarrollo de las tareas planificadas pese a tener un sobre aumento de 10 horas en tareas de optimización. Con todo ello, el proyecto ha finalizado un mes y medio antes de lo esperado, contando con las mismas tareas por completar y con una compensación de tareas de +10 horas extras invertidas respecto a la previsión inicial dada, es decir, he invertido 355 horas. Así pues, parto aún con más tiempo de margen hasta la convocatoria contra cualquier contra tiempo. El diagrama de Gantt que refleja este progreso es el siguiente:

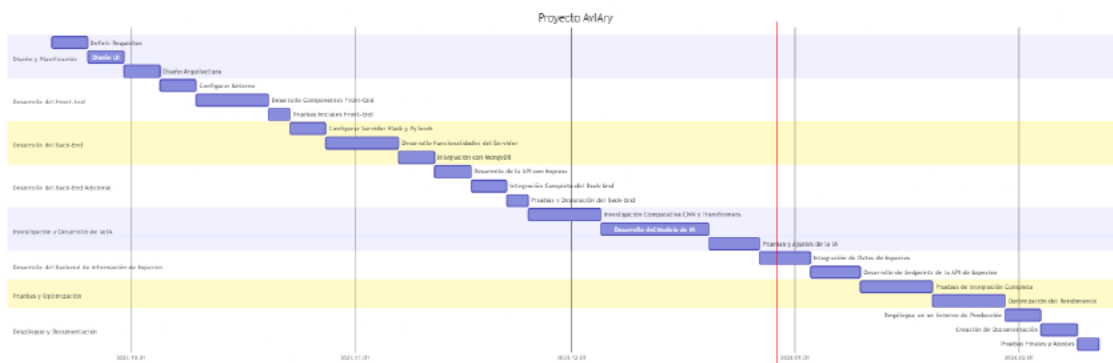


Imagen 108: Diagrama de Gantt final

9.2 PLANIFICACIÓN FINANCIERA

En este caso, como hemos trabajado durante más horas, el coste total sufrirá leves diferencias respecto al original, siendo este finalmente:

Precio Inicial sin impuestos= $8875 (355h * 25€/h) + 950 = 9825 €$

Contando ahora con que tributamos en el territorio español en un tramo de impuestos de un 21% que es el estándar, el coste final sería:

Precio Inicial Total = 9825 € + 21% = 11888.25 €

10. ESTUDIO DE MERCADO

10.1 CLIENTES POTENCIALES

AvlAry se erige como una guía amigable y accesible para aquellos que disfruten con la observación de aves, sin importar su nivel de experiencia. Este análisis meticuloso nos lleva a identificar segmentos clave dentro de este público aficionado, menos exigente pero no menos apasionado:

10.1.1 NOVATOS EN LA OBSERVACIÓN DE AVES EN BUSCA DE DESCUBRIMIENTOS (14-65 años)

Entre aquellos que recién comienzan su viaje en la observación de aves, AvIAry emerge como una herramienta amena y educativa. Este segmento, compuesto por novatos entusiastas, busca una aplicación que simplifique el proceso de identificación de aves sin sacrificar la calidad de la información. AvIAry responde a esta necesidad, proporcionando una interfaz intuitiva y fácil de usar, acompañada de contenido informativo digestible que permite a estos recién llegados adentrarse en el fascinante mundo de las aves con confianza.

10.1.2 FOTÓGRAFOS AFICIONADOS EN BÚSQUEDA DE VALIDACIÓN Y CONEXIÓN (14-65 años)

Los fotógrafos aficionados, aunque pueden carecer de la experiencia técnica de profesionales, comparten una pasión por capturar nuevas especies. AvlAry se presenta como la herramienta perfecta para este grupo, brindando validación instantánea de sus logros. La interfaz de usuario simplificada y las funciones intuitivas permiten a estos fotógrafos centrarse en lo que más aman: capturar momentos únicos.

10.1.3 ESTUDIANTES OCASIONALES EN BIOLOGÍA Y ECOLOGÍA (14-25 años)

Para aquellos que tienen un interés ocasional en la biología y ecología de las aves, AvlAry se adapta a su ritmo. Los estudiantes que buscan información de manera puntual encuentran en la aplicación una fuente fácil de entender y utilizar. AvlAry les ofrece datos concisos y relevantes, sin abrumarlos con detalles científicos complejos, fomentando un aprendizaje agradable y gradual.

10.1.4 AMANTES DE LAS AVES COMO MASCOTAS (14-40 años)

El segmento de amantes de las aves como mascotas, en su mayoría, busca información práctica y utilitaria. AvlAry proporciona respuestas claras y directas a sus preguntas sobre alimentación, peso o tamaño habitual de sus compañeros emplumados. La simplicidad en el diseño y la presentación de información contribuyen a que los usuarios obtengan respuestas rápidas y útiles.

10.2 PLAN DE COMERCIALIZACIÓN

Desde AvlAry apoyamos el uso gratuito de nuestra plataforma para promover la educación y la conciencia por la biodiversidad y el mundo aviar; no obstante, también hemos creemos que la aplicación podría expandirse y comercializarse de diferentes formas, siendo estas las siguientes:

10.2.1 MODELOS DE IA PERSONALIZADOS

A través de un sistema de usuarios poder implementar un sistema freemium en el que todas las funcionalidades básicas fueran de uso público, pero para aquel usuario que desee una mayor precisión, podrá pagar por elegir entre los diferentes modelos de inteligencia artificial disponibles. Hay que destacar que el modelo por defecto no será el más potente, sino que este estará reservado para aquellos usuarios que paguen un plan mensual de 3€/mes o un plan anual de 30€/año.

10.2.2 NÚMERO LIMITADO DE ESCANEOS

De la misma forma que el anterior, aquellos usuarios que quieran usar la aplicación contarán con un máximo de 15 escaneos por dispositivos, por lo que cuando quieran más deberán abonar una cuota de desbloqueo completa de 3€/mes o un plan anual de 30€/año.

10.2.3 PAQUETES DE ESPECIES EXCLUSIVOS

Para aquellos usuarios que sean más curiosos o quieran usar AvlAry de forma más profesional pueden contar con un catálogo más extenso más allá de lo habitual para un usuario promedio con nuevas especies y nuevos datos sobre ellas. Aquellos que lo deseen deberán abonar la cantidad de 3€/mes o un plan anual de 30€/año para desbloquear todo el contenido.

Como el precio de todos los casos de comercialización son los mismos, podemos usarlos para afirmar que necesitaremos para cubrir los gastos de la aplicación:

- ❖ 397 usuarios suscritos en el plan anual
- ❖ 166 usuarios suscritos al plan mensual durante 2 años
- ❖ 331 usuarios suscritos al plan mensual durante un año
- ❖ 660 usuarios suscritos al plan mensual durante 6 meses
- ❖ 1321 usuarios suscritos al plan mensual durante 3 meses
- ❖ 3963 usuarios suscritos al plan mensual durante un mes

10.3 ESTRATEGIAS DE MARKETING

10.3.1 LANZAMIENTO CON OFERTAS ESPECIALES

En la salida de AvIAry, se ofrecerán al público características premium de forma gratuita durante un período de prueba inicial para generar interés y obtener retroalimentación.

10.3.2 CAMPAÑAS DE SESIBILIZACIÓN

Destacaremos el impacto positivo de las compras premium en la conservación de aves y en el desarrollo continuo de la aplicación.

10.3.3 PROGRAMA DE REFERIDOS CON BENEFICIOS

Implementaremos un programa de referidos donde los usuarios pueden ganar acceso a características premium al referir con éxito a nuevos usuarios.

10.3.4 COLABORACIONES CON ORGANIZACIONES DE CONSERVACIÓN

Creemos que es de gran importancia asociarnos con organizaciones de conservación aviar para destacar la contribución de las compras premium a proyectos de conservación y sostenibilidad.

Este plan de comercialización busca no solo monetizar la aplicación de manera sostenible sino también mejorar significativamente la experiencia del usuario premium, ofreciendo valor adicional y apoyo continuo a la pasión por las aves.

11. CONCLUSIONES

El proyecto AvIAry ha sido un viaje apasionante que ha dejado una huella profunda en mi corazón y en mi visión del potencial que la tecnología tiene para conectarnos con la naturaleza. Trabajar en este proyecto no solo ha sido una experiencia profesional, sino un verdadero viaje de descubrimiento, crecimiento y conexión con una comunidad apasionada.

En el ámbito personal, la inmersión en el mundo de las aves, su observación y estudio, ha abierto mi mente a la riqueza de la biodiversidad que nos rodea. La pasión compartida por la ornitología, expresada a través de la aplicación AvIAry, ha sido una fuente constante de inspiración.

Profesionalmente, el desarrollo de AvIAry ha sido un desafío enriquecedor que ha ampliado mis habilidades y conocimientos. Desde la investigación comparativa de modelos de inteligencia artificial hasta la implementación de una interfaz de usuario intuitiva, cada paso ha sido una lección valiosa. La respuesta entusiasta de los usuarios que han probado la

aplicación ha sido la recompensa más gratificante, validando no solo el trabajo técnico, sino también el propósito detrás de AvIAry: hacer que la ornitología sea accesible y emocionante para todos.

Para finalizar, quiero comentar que AvIAry ha sido más que un proyecto; ha sido un viaje de autodescubrimiento, aprendizaje y conexión con la naturaleza. La pasión y el entusiasmo de crear algo nuevo, la satisfacción de superar desafíos técnicos y la contribución a la conservación aviar han convertido este proyecto en una experiencia digna para finalizar mi etapa como estudiante de ingeniería informática. AvIAry no solo identifica aves; ha identificado una nueva perspectiva sobre cómo la tecnología puede inspirar y unir a las personas a través de la maravilla de las aves y la naturaleza.

12. TRABAJO FUTURO

A medida que AvIAry ha evolucionado, nuevas oportunidades y desafíos han surgido. En esta sección, exploraremos áreas que no se abordaron completamente en la fase actual del proyecto, delineando posibilidades para mejorar y expandir la aplicación. Desde la diversificación de modelos de inteligencia artificial hasta el desarrollo de características específicas para usuarios, estas líneas de trabajo futuro señalan la dirección hacia la evolución continua de AvIAry.

12.1 EXPANSIÓN DE LA BASE DE DATOS DE ESPECIES

Aunque AvIAry ya cuenta con una sólida base de datos de especies, el trabajo futuro podría centrarse en su expansión para incluir una mayor diversidad global de aves. Esto implica una investigación extensa y colaboraciones con expertos en aves de diversas regiones del mundo.

12.2 DESARROLLO DE COLABORACIONES EDUCATIVAS

Establecer asociaciones con instituciones educativas para desarrollar módulos educativos específicos. La creación de herramientas pedagógicas y recursos adaptados para entornos educativos podría ampliar la utilidad de AvIAry en el ámbito académico.

12.3 IMPLEMENTACIÓN DE FUNCIONALIDADES DE REGISTRO OPCIONAL

Ofrecer la opción de registro a los usuarios para mejorar la personalización de la experiencia. Permitir que los usuarios creen perfiles personales, almacenen sus observaciones y compartan sus hallazgos con la comunidad, fomentando una participación más activa.

12.4 INTEGRACIÓN DE REDES SOCIALES

Mejorar las opciones de interacción social permitiendo a los usuarios compartir fácilmente sus observaciones, logros y fotografías de aves en redes sociales. Esto no solo promovería la aplicación, sino que también fortalecería la comunidad AvIAry en plataformas externas.

12.5 DIVERSIFICACIÓN DE MODELOS DE IA

Explorar la posibilidad de ofrecer modelos de inteligencia artificial especializados para diferentes regiones geográficas. Esto garantizaría una identificación precisa de especies específicas de cada área, adaptándose a las necesidades de la comunidad global de usuarios.

12.6 USO DE PLATAFORMAS DE PAGO PARA LA ARQUITECTURA

En AvlAry creemos que todo se puede lograr únicamente usando herramientas de uso gratuito, es así como AvlAry se nutre de todos los planes gratuitos de herramientas como Netflix, MongoDB o Render, no obstante, esto ocasiona que los tiempos de respuesta no sean óptimos y se produzcan retrasos.

Uno de los principales problemas que nos encontramos a la hora de probar la aplicación es que Render, para no gastar recursos, si el método de IA no recibe o envía peticiones en un margen corto de tiempo la obliga a hibernar hasta que reciba una petición nueva. Esta situación, fácilmente abordable con un plan de pago, provoca que el método de IA tarde casi un minuto en despertar tras la petición, lo que provoca confusión al usuario. No obstante, cuando el método de IA se encuentra en funcionamiento no tarda más de 5 segundos en ofrecer una respuesta por lo que la solución en este caso es pagar para que Render no actúe en la actividad del servidor.

13. BIBLIOGRAFÍA Y ANEXOS

- [1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *Imagenet classification with deep convolutional neural networks*. <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
- [2] Erhan, D., Bengio, Y., Courville, A., Manzagol, P. A., Vincent, P., & Bengio, S. (2010). *Why does unsupervised pre-training help deep learning?* <https://dl.acm.org/doi/10.1145/1756006.1756034>
- [3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). *Attention is all you need*. <https://arxiv.org/abs/1706.03762>
- [4] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., & Hinton, G. (2020). *An image is worth 16x16 words: Transformers for image recognition at scale*. <https://arxiv.org/abs/2010.11929>
- [5] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). *End-to-End Object Detection with Transformers*. <https://arxiv.org/abs/2005.12872>
- [6] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*. <https://arxiv.org/abs/1602.07261>
- [7] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. <https://arxiv.org/abs/1704.04861>
- [8] PyTorch Documentation. (2024). *PyTorch - An open-source deep learning platform*. <https://pytorch.org/>

- [9] *Node.js*. (2024). Express - Fast, unopinionated, minimalist web framework for Node.js. <https://expressjs.com/>
- [10] Flask Documentation. (2024). *Flask - A micro web framework written in Python*. <https://flask.palletsprojects.com/>
- [11] Bootstrap Documentation. (2024). *Bootstrap - The most popular HTML, CSS, and JS library in the world*. <https://getbootstrap.com/>
- [12] React Documentation. (2022). *React - A JavaScript library for building user interfaces*. <https://reactjs.org/>
- [13] Reactiflux Community. (2022). *Reactiflux - A community of React developers*. <https://www.reactiflux.com/>
- [14] DotCSV Canal de Youtube. (2024). *Canal de divulgación científica y programación*. <https://www.youtube.com/c/DotCSV>
- [15] SensIO YouTube Channel. (2024). *Canal de divulgación científica y programación*. <https://www.youtube.com/@juansensio>
- [16] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2019). *Transformers: State-of-the-Art Natural Language Processing*. <https://arxiv.org/abs/1910.03771>
- [17] Hugging Face Forum. (2022). *Transformers - Hugging Face Community*. <https://discuss.huggingface.co/>
- [18] GitHub Repository. (2022). *Hugging Face Transformers GitHub Repository*. <https://github.com/huggingface/transformers>
- [19] PyTorch Tutorials. (2022). *Transfer Learning for Computer Vision Tutorial*. https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html
- [20] GitHub Repository. (2022). *ResNet Applications - Collection of pre-trained models using ResNet*. <https://github.com/KaimingHe/deep-residual-networks#applications-and-performance>
- [21] GitHub Repository. (2022). *Deep Residual Learning for Image Recognition - Official Implementation*. <https://github.com/KaimingHe/deep-residual-networks>