



**Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria de
Ingeniería campus Zacatecas**

**Área de ubicación para el desarrollo del
trabajo**

Ingeniería en Sistemas Computacionales

Línea de investigación

Desarrollo de aplicaciones móviles

Título del proyecto de Trabajo Terminal
Sistema para muestreo de aves en la ciudad de
Zacatecas

Presenta(n):

Axel Frederick Félix Jiménez
Vania Stephany Sánchez Lee.

Director:

ISC. Efraín Arredondo Morales

Asesores:

M.I.S Isaul Ibarra Belmonte



Zacatecas, Zacatecas a 12 de 06 de 2024

Índices

Índice de contenido

Resumen.....	1
Definición del problema.....	1
Contexto y antecedentes generales del problema.....	2
Situación problemática o problema de investigación.....	3
Estado del arte.....	4
Descripción del proyecto.....	9
Objetivo general del proyecto.....	10
Objetivos particulares del proyecto.....	10
Justificación.....	10
Marco teórico.....	14
Marco Metodológico.....	26
Metodologías.....	26
Procedimientos y Actividades.....	28
Análisis y Discusión de los Resultados.....	31
Análisis de la selección de la metodología (TT-I):	31
Gestión del proyecto	34
1. Plan del proyecto.....	34
2. Manejo de desviaciones en la ejecución del plan.....	36
3. Plan de los riesgos del proyecto.....	38
Desarrollo del proyecto	38
1. Resumen del análisis del sistema.....	38
2. Diseño del sistema.....	41
a. Arquitectura del sistema.....	41
b. Matriz de trazabilidad	44
c. Diseño de la base de datos.	45
d. Manejo de archivos.....	47

e.	Diagramas de actividad	49
f.	Diagramas de secuencia	50
g.	Mockups.....	52
	Análisis de resultados.....	53
	Conclusiones y Recomendaciones	57
	Fuentes de consulta	59
	Firmas.....	61
	Autorización.....	61
	Apéndices	62

Índice de tablas

Tabla I Ventajas y desventajas del sistema de muestreos manual.....	3
Tabla II Comparación de software para la identificación de aves.....	4
Tabla III Características de Merlin Bird ID	5
Tabla IV Características de eBird	6
Tabla V Características de Birdsnap	8
Tabla VI Actividades del manual de métodos de observación de aves.....	11
Tabla VII Tabla comparativa del método manual y IdBird	12
Tabla VIII Tabla comparativa del sistema propuesto con los existentes	13
Tabla IX Características mínimas de teléfonos de gama media-alta.....	18
Tabla XI Matriz de trazabilidad	45

Índice de figuras

Figura 1 Muestra de Merlin Bird ID	6
Figura 2 Muestra de eBird.....	7
Figura 3 Muestra de Birdsnap	8
Figura 4 Carpintero de Arizona.....	15
Figura 5 Papamoscas Pinero	15
Figura 6 Carbonero Mexicano.....	16
Figura 7 Zzacatonero Serrano	16
Figura 8 Papamoscas Cardenalito	17
Figura 9 Logo de TensorFlow Lite	19
Figura 10 Logo de Kotlin	20
Figura 11 Logo de SQL.....	21
Figura 12 Logo de python	21
Figura 13 Logo de SQLite.....	22

Figura 14 Logo de MongoDB	22
Figura 15 Logo de PostageApp.....	23
Figura 16 Representación gráfica del modelo V	24
Figura 17 Evidencia de las horas estimadas y registradas en el proyecto	35
Figura 18 Evidencia de las horas estimadas y registradas por etapa del proyecto	35
Figura 19 Plan de proyecto de TT1 completo	36
Figura 20 Evidencia de las desviaciones del proyecto	36
Figura 21 Evidencia de las desviaciones del reporte de proyecto	37
Figura 22 Diagrama de casos de uso	39
Figura 23 Diagrama de contexto	41
Figura 24 Diagrama de contenedor	42
Figura 25 Diagrama de componentes.....	43
Figura 26 Diagrama de clases	44
Figura 27 Diagrama de base de datos.....	46
Figura 28 Diagrama de entidad – relación	47
Figura 29 Diagrama de manejo de archivos.....	48
Figura 30 Diagrama de actividad de exportación de bitácoras de campo	50
Figura 31 Diagrama de secuencia de inicio de sesión.....	51
Figura 32 Diseño de prototipos	53
Figura 33 Cronograma propuesto para TT2	58

Resumen.

Este proyecto tiene como objetivo desarrollar una herramienta para muestrear aves en la ciudad de Zacatecas, utilizando una cámara de celular para capturar imágenes de aves y mostrar información sobre sus características al usuario. El proyecto busca mejorar el sistema de inventario de aves utilizado por los biólogos al facilitar la toma de datos y reducir el tiempo requerido para la recolección de información. Se utilizarán técnicas de procesamiento de imágenes y reconocimiento de patrones para identificar y clasificar las aves de manera rápida y efectiva, lo que permitirá un seguimiento más preciso de las aves en la región. La implementación del sistema se realizará a través de una aplicación móvil para dispositivos Android. Por lo que, este proyecto tiene el potencial de mejorar significativamente el sistema de inventario de aves utilizado por los biólogos en la ciudad de Zacatecas empleando diversas tecnológicas e inteligencia artificial.

Palabras clave: Sistema, Aves, Ciudad de Zacatecas, Cámara de celular, Procesamiento de imágenes, Reconocimiento de patrones, Mejoramiento del sistema de inventario de Aves, Implementación del Sistema, Aplicación móvil.

Definición del problema.

La falta de información que permite conocer los procesos evolutivos, biogeográficos y ecológicos generados por la limitada investigación y monitoreo de las aves existentes en Zacatecas, provoca falta de conocimiento del sistema biológico enfocado a las aves de Zacatecas, vital para su estudio y preservación, así como dar seguimiento a las especies en peligro de extinción y especies endémicas [1].

Según un informe de la Comisión Nacional para el Conocimiento y Uso de la Biodiversidad (CONABIO), México enfrenta una escasez significativa de datos sobre muchas especies de aves, especialmente en regiones menos estudiadas como Zacatecas [29][30]. Esta falta de información ha sido señalada como un obstáculo importante para la conservación de la biodiversidad [31]. Además, estudios recientes indican que el monitoreo continuo y

sistemático es crucial para detectar cambios en las poblaciones de aves y responder adecuadamente a las amenazas.

La falta de un sistema de monitoreo robusto restringe nuestra capacidad para comprender los cambios en la distribución y abundancia de las aves, lo que a su vez dificulta la toma de decisiones informadas **realizadas específicamente por los biólogos encargados de la Universidad Autónoma de Zacatecas**. Esta deficiencia de datos es particularmente problemática en regiones como Zacatecas, donde muchas especies pueden estar en declive sin que haya información precisa que lo confirme. Sin un monitoreo adecuado, es difícil evaluar el estado real de las poblaciones de aves y las amenazas que enfrentan, lo que impide la implementación de estrategias de conservación efectivas.

Contexto y antecedentes generales del problema.

La problemática de la falta de un sistema eficiente para el muestreo de aves en la ciudad de Zacatecas es la más importante a considerar. Actualmente, los biólogos y científicos que se dedican a la investigación de la fauna silvestre en la zona tienen que realizar conteos de manera manual y visual, lo que puede ser muy limitado y poco efectivo en términos de la calidad y cantidad de información recopilada. Además, la falta de un sistema automatizado de inventario de aves puede obstaculizar la toma de decisiones en relación con la conservación y protección de estas especies.

Por otro lado, existen diversos proyectos y sistemas en todo el mundo que utilizan tecnologías avanzadas, como cámaras trampa, drones y sistemas de seguimiento GPS, para la recolección y análisis de datos de fauna silvestre. Sin embargo, en la región de Zacatecas, la mayoría de los estudios de fauna silvestre se han realizado de manera manual y visual, lo que ha limitado significativamente la calidad y cantidad de información recopilada.

A continuación, en la Tabla I, se comparan las ventajas y desventajas del sistema de muestreos actual hecho de manera manual.

Tabla I Ventajas y desventajas del sistema de muestreos manual

Ventajas	Desventajas
<ul style="list-style-type: none"> • Permite un control detallado y específico del monitoreo y manejo directo de las aves. • Adaptable a diversos tipos de estudios y necesidades específicas. • Proporciona una interacción directa con las aves y el entorno, lo que puede ser importante para ciertos estudios. • El registro manual permite anotar observaciones detalladas y contextualizadas. 	<ul style="list-style-type: none"> • Mayor tiempo y probabilidad de errores humanos durante la transcripción de datos. • Depende en gran medida de la experiencia y habilidades individuales del observador. • Puede ser invasivo para las aves y perturbador para su entorno natural. • La variabilidad en la ejecución puede llevar a inconsistencias en los datos. • Menos capacidad para manejar grandes volúmenes de datos o realizar análisis complejos.

Sin embargo, en la región de Zacatecas, la mayoría de los estudios de fauna silvestre se han realizado de manera manual y visual, lo que ha limitado significativamente la calidad y cantidad de información recopilada. Por lo tanto, la creación de un sistema automatizado para el muestreo de aves podría llenar este vacío en la investigación de fauna silvestre en la región de Zacatecas.

Situación problemática o problema de investigación.

La situación problemática que se aborda en este proyecto es la necesidad de mejorar el sistema de inventario de aves en la ciudad de Zacatecas, México. Actualmente, los biólogos y expertos en aves utilizan métodos manuales para identificar y contabilizar las especies de aves presentes en la zona, lo que puede ser muy limitado y propenso a errores. Además, esta tarea puede ser muy laboriosa y costosa en términos de tiempo y recursos [2] Por lo tanto, se requiere una solución más eficiente y precisa para la identificación y el registro de las aves en la ciudad.

Para respaldar esta problemática, se han realizado varios estudios y publicaciones sobre la importancia de mejorar los sistemas de inventario y monitoreo de aves [4] [5]. Un estudio reciente realizado en Poza Rica, Veracruz, México, resalta la necesidad de realizar inventarios de aves, especialmente en áreas donde el conocimiento de la avifauna es aún deficiente1. Este estudio también destaca cómo la urbanización rápida ha modificado los entornos donde viven las aves, lo que resulta en cambios en la composición de las aves y lleva a la necesidad de realizar inventarios [6].

Además, con el avance de la tecnología, se han desarrollado herramientas de aprendizaje automático para la identificación de aves, ayudando a los ecólogos a monitorear especies raras en la naturaleza [7]. También se han desarrollado métodos automáticos de reconocimiento de especies de aves a partir de imágenes que aprovechan el aprendizaje contrastante y la mejora de características para abordar estos desafíos [8].

Estos avances tecnológicos pueden ser de gran ayuda para mejorar los sistemas de inventario y monitoreo de aves en la ciudad de Zacatecas, y en otras partes de México.

Estado del arte.

El muestreo de aves es una actividad que se realiza para el estudio y conservación de la fauna. Sin embargo, la identificación de aves puede resultar una tarea difícil para personas que no son expertas en el tema. Con el fin de facilitar la tarea de identificación de aves, surge la idea de desarrollar un sistema para muestreo de aves utilizando el reconocimiento de imágenes mediante una cámara de celular.

En la actualidad, existen diversas aplicaciones móviles y software especializado en la identificación de aves mediante el reconocimiento de imágenes. A continuación, se presenta la Tabla II que compara las características de algunos de este software.

Tabla II Comparación de software para la identificación de aves

No.	Características	Merlin Bird ID	eBird	Birdsapp
1	Identificación de aves mediante imágenes	X	-	X
2	Identificación de aves mediante sonidos	X	-	-
3	Gratis	X	X	-
4	Disponible para iOS	X	-	-
5	Disponible para Android	X	-	-
6	Registro de observaciones	-	X	-
7	Generación de mapas de distribución	-	X	-
8	Información detallada sobre las aves	X	X	-
9	Uso de inteligencia artificial	X	-	X
10	Enfoque en aficionados	X	-	-
11	Enfoque en investigación científica	-	X	X
12	Plataforma en línea	-	X	-

Desglose de Software y Proyectos:

1. Merlin Bird ID:

- **Descripción:** Aplicación móvil desarrollada por el Cornell Lab of Ornithology que utiliza inteligencia artificial para identificar aves a partir de fotos o grabaciones de su canto.
- **Características:** Gratuita, disponible para dispositivos iOS y Android, ofrece información detallada sobre las características de cada ave, incluyendo su distribución geográfica y los sonidos que emite [9]. Mostradas en la Tabla III.
- **Enfoque:** Identificación de aves para aficionados y observadores de aves.

Tabla III Características de Merlin Bird ID

Característica	Detalles de Merlin Bird ID
Identificación de aves mediante imágenes	Utiliza inteligencia artificial para identificar aves a partir de fotos.
Identificación de aves mediante sonidos	Reconoce aves mediante grabaciones de su canto.
Gratis	Sí
Disponible para iOS	Sí, disponible en la App Store.
Disponible para Android	Sí, disponible en Google Play.
Registro de observaciones	No
Generación de mapas de distribución	No
Información detallada sobre las aves	Proporciona características, distribución geográfica y sonidos de cada ave.
Uso de inteligencia artificial	Sí, utiliza algoritmos avanzados para el reconocimiento de imágenes y sonidos.
Enfoque en aficionados	Sí, diseñado para observadores de aves de todos los niveles.
Enfoque en investigación científica	No, no está específicamente diseñado para investigación científica.
Plataforma en línea	No, es una aplicación móvil.

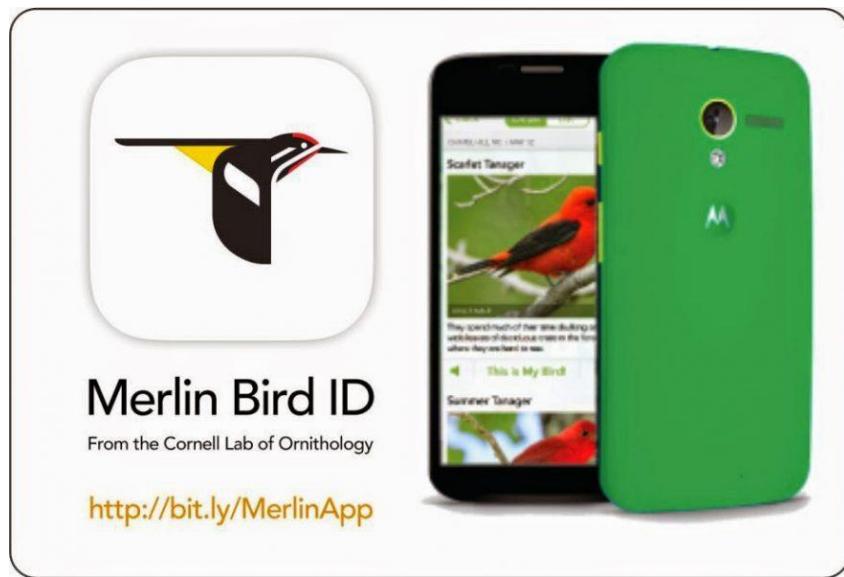


Figura 1 Muestra de Merlin Bird ID

2. eBird:

- **Descripción:** Plataforma en línea desarrollada por el Cornell Lab of Ornithology para la observación y registro de aves en todo el mundo.
- **Características:** Permite cargar registros de observación con ubicación, fecha, hora, especies observadas y número de aves. Genera mapas de distribución de aves en todo el mundo [10]. Mostradas en la tabla IV.
- **Enfoque:** Herramienta valiosa para la investigación científica y la conservación de aves, proporcionando datos sobre la presencia y abundancia de especies en diferentes áreas geográficas [12].

Tabla IV Características de eBird

Característica	Detalles de Merlin Bird ID
Identificación de aves mediante imágenes	No
Identificación de aves mediante sonidos	No
Gratis	Sí, es una plataforma gratuita.
Disponible para iOS	No
Disponible para Android	Sí, disponible en Google Play.

Registro de observaciones	Sí, permite a los usuarios registrar observaciones con ubicación, fecha, hora, especies y número de aves.
Generación de mapas de distribución	Sí, genera mapas de distribución de aves en todo el mundo basados en las observaciones de los usuarios.
Información detallada sobre las aves	Proporciona datos sobre la presencia y abundancia de especies en diferentes áreas geográficas.
Uso de inteligencia artificial	No especificado.
Enfoque en aficionados	No, está enfocado principalmente en la investigación y conservación.
Enfoque en investigación científica	Sí, proporciona datos valiosos para la investigación científica y la conservación de aves.
Plataforma en línea	Sí, es una plataforma en línea accesible desde cualquier navegador web.

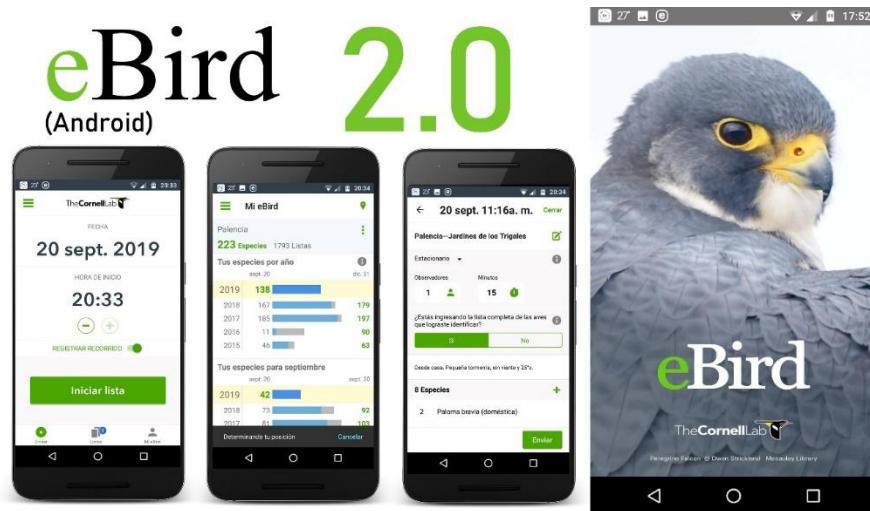


Figura 2 Muestra de eBird

3. Birdsnap:

- **Descripción:** Proyecto de investigación desarrollado por la Universidad de Columbia y la Universidad de Maryland.
- **Características:** Utiliza algoritmos de aprendizaje automático para identificar aves a partir de imágenes [13]. Mostradas en la tabla V.

- **Enfoque:** Investigación en técnicas de inteligencia artificial para la identificación de aves.

Tabla V Características de Birdsnap

Característica	Detalles de Merlin Bird ID
Identificación de aves mediante imágenes	Utiliza algoritmos de aprendizaje automático para identificar aves a partir de imágenes.
Identificación de aves mediante sonidos	No
Gratis	No especificado.
Disponible para iOS	No especificado.
Disponible para Android	No especificado.
Registro de observaciones	No permite el registro de observaciones.
Generación de mapas de distribución	No genera mapas de distribución.
Información detallada sobre las aves	No especificado.
Uso de inteligencia artificial	Sí, utiliza algoritmos avanzados de aprendizaje automático para el reconocimiento de imágenes.
Enfoque en aficionados	No está específicamente diseñado para aficionados.
Enfoque en investigación científica	Sí, está diseñado para la investigación y el desarrollo de técnicas de inteligencia artificial en la identificación de aves.
Plataforma en línea	No especificado.

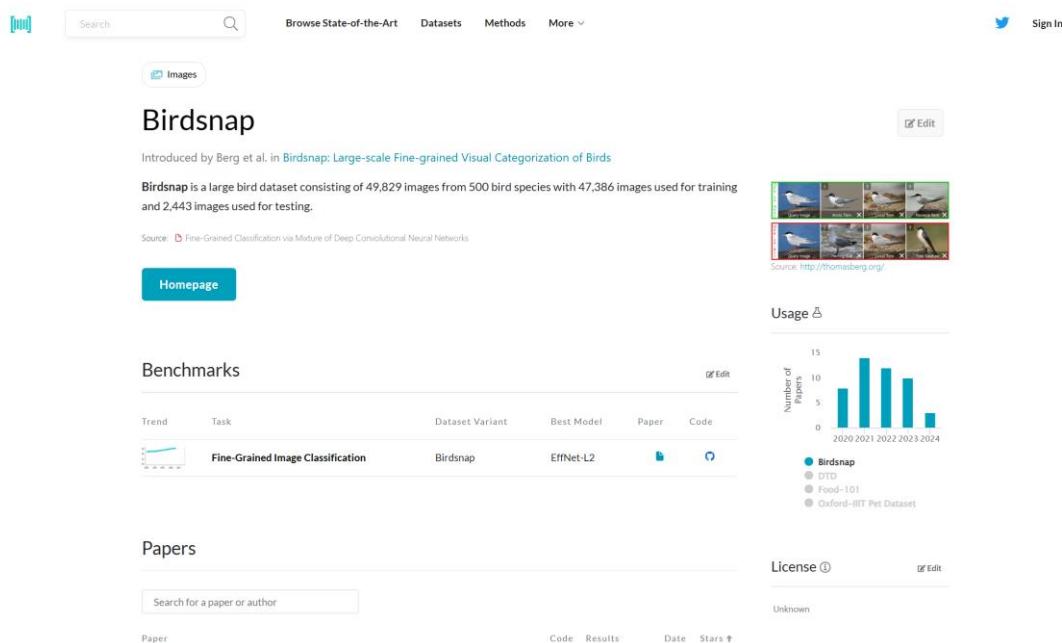


Figura 3 Muestra de Birdsnap

En el contexto mexicano, se han desarrollado proyectos para el estudio de la fauna, como el proyecto Biósfera Urbana, que busca documentar la biodiversidad de la ciudad de México utilizando tecnología y técnicas de ciencia ciudadana. Además, la Comisión Nacional para el Conocimiento y Uso de la Biodiversidad (CONABIO) cuenta con un catálogo de aves mexicanas, que incluye información acerca de las características de las especies y su distribución geográfica [14].

Descripción del proyecto.

El proyecto que se propone es el desarrollo de un sistema para muestreo de aves en la ciudad de Zacatecas, el cual utilizará una cámara de celular para identificar las características de las aves captadas por el usuario. El objetivo principal de este proyecto es mejorar el sistema de inventario de aves utilizado por los biólogos, proporcionando una herramienta fácil y accesible para la identificación y recopilación de datos de las aves presentes en la ciudad.

El sistema propuesto funcionará mediante la utilización de técnicas de procesamiento de imágenes y aprendizaje automático. Cuando el usuario capture por medio de la cámara del celular o seleccione una imagen de un ave en particular, la aplicación utilizará algoritmos de reconocimiento de patrones para identificar la especie de ave y recopilar datos relevantes sobre su tamaño, forma, coloración, entre otros aspectos. Estos datos serán almacenados en una base de datos y estarán disponibles para su posterior análisis por parte de los biólogos.

La aplicación también contará con una interfaz de usuario donde se podrán ingresar datos del muestreo, registrar la ubicación y hora de la observación, así como tomar notas adicionales sobre el comportamiento y hábitat de las aves observadas. Además, se considera la posibilidad de integrar una funcionalidad de georreferenciación para obtener datos precisos sobre la ubicación de las aves y su relación con el entorno.

Este sistema proporcionará una herramienta valiosa para la identificación y seguimiento de las poblaciones de aves en la ciudad de Zacatecas, al permitir a los usuarios identificar y registrar las características de las aves utilizando una cámara de celular y una base de datos de referencia. Al utilizar tecnologías de reconocimiento de imágenes y aprendizaje automático, la aplicación podrá identificar la especie de ave y proporcionar información útil sobre sus características físicas y hábitat [3]. Esto permitirá una identificación más precisa y detallada de las especies de aves presentes en la ciudad, lo que puede ser muy valioso para los biólogos y otros expertos en aves que trabajan en la conservación y el estudio de la fauna local.

Objetivo general del proyecto.

Llevar a cabo un muestreo no invasivo de las aves en la zona conurbada para generar un registro de las características de las aves analizadas y llevar dichos a la visualización del usuario.

Objetivos particulares del proyecto.

- Identificar, registrar, analizar, clasificar a las aves muestreadas.
- Identificar conforme a la normativa de PROFEPA (Procuraduría Federal de Protección al Ambiente) si las aves muestreadas se encuentran en peligro de extinción.
- Generar una bitácora con las aves registradas.
- Localizar la posición del registro en tiempo real a través de coordenadas UTP.

Justificación.

El proyecto de Sistema para muestreo de aves en la ciudad de Zacatecas busca crear un sistema de inventario de aves más eficiente y efectivo para los biólogos que se encargan de la supervisión de las poblaciones de aves en la región. Actualmente, los métodos de

recolección de datos son en su mayoría manuales, tal como se describe en el manual de métodos[55], y requieren de mucho tiempo por parte de los biólogos en cada actividad que solicita el manual según la Tabla VI, lo que limita su capacidad para monitorear de manera constante las poblaciones de aves y su comportamiento.

Tabla VI Actividades del manual de métodos de observación de aves

Actividad	Manual de Métodos
Preparación y Desplazamiento	1-2 horas
Instalación de Equipos (como redes de niebla)	2-3 horas
Captura y Anillamiento de Aves	4-6 horas (por día)
Recolección de Datos y Desmontaje de Equipos	1-2 horas
Procesamiento y Análisis de Datos	2-3 horas por día de muestreo
Total Aproximado por Día de Muestreo	10-16 horas
Duración del Muestreo Completo	2-4 semanas (dependiendo de la frecuencia y el área de muestreo)

La implementación de una aplicación móvil con la capacidad de identificar y clasificar las características de las aves a través del uso de cámaras de celular permitiría a los biólogos recolectar información de manera más rápida y efectiva. Además, el uso de técnicas de procesamiento de imágenes y reconocimiento de patrones ayudará a mejorar la precisión en la identificación y seguimiento de las poblaciones de aves, lo que permitirá una mejor comprensión de su comportamiento y hábitat. Se puede apreciar estas ventajas en la Tabla VII.

Este proyecto no solo tendría un impacto significativo en la investigación y conservación de las poblaciones de aves en la región de Zacatecas, sino que también podría tener aplicaciones en otras áreas de la biología y la conservación. Además, el uso de tecnología accesible como las cámaras de celular, hace que la implementación de este sistema sea accesible y de bajo costo.

Tabla VII Tabla comparativa del método manual y IdBird

Actividad	Manual de Métodos	IdBird
Preparación y Desplazamiento	Requiere tiempo considerable para preparación y traslado al sitio de muestreo.	Requiere tiempo mínimo para preparación y traslado al sitio de muestreo.
Instalación de Equipos	Necesita instalación de equipos como redes de niebla.	No requiere instalación de equipos adicionales.
Captura de Datos	Captura y anillamiento de aves, métodos laboriosos y manuales.	Captura de imágenes de aves mediante dispositivo móvil.
Recolección de Datos	Recolección manual de datos y desmontaje de equipos.	Recolección de datos automática al tomar las fotos.
Procesamiento de Datos	Procesamiento y análisis de datos requiere tiempo y es manual.	Procesamiento automático y en tiempo real de las imágenes.
Análisis de Datos	Ánalisis detallado y manual de datos recolectados.	Ánalisis automático y rápido mediante algoritmos de reconocimiento de patrones.
Manejo de Datos	Datos almacenados manualmente en registros físicos o bases de datos tradicionales.	Datos almacenados automáticamente en bases de datos electrónicas.
Frecuencia de Muestreo	Frecuencia de muestreo limitada por el tiempo y recursos disponibles.	Mayor frecuencia de muestreo posible debido a la eficiencia del proceso.
Personal Necesario	Requiere un equipo de personal capacitado para la instalación de equipos y recolección de datos.	Puede ser realizado por un número reducido de personas con dispositivos móviles.
Capacitación del Personal	Requiere capacitación extensa para el uso de métodos tradicionales y equipos.	Requiere capacitación mínima para el uso de la aplicación móvil.
Condiciones Meteorológicas	Las condiciones climáticas adversas pueden afectar significativamente el proceso de muestreo.	Menor impacto de las condiciones climáticas debido a la rapidez del proceso.

La implementación de una aplicación móvil con la capacidad de identificar y clasificar las características de las aves a través del uso de cámaras de celular permitiría a los biólogos recolectar información de manera más rápida. Además, el uso de técnicas de procesamiento de imágenes y reconocimiento de patrones ayudará a mejorar la precisión en la identificación y seguimiento de las poblaciones de aves, lo que permitirá una mejor comprensión de su comportamiento y hábitat. Se pueden apreciar estas ventajas en la Tabla VII.

Tabla VIII Tabla comparativa del sistema propuesto con los existentes

No.	Característica	IdBird	Merlin Bird ID	eBird	Birdsnap	Monitoreo Manual
1	Identificación de aves mediante imágenes	X	X	-	X	-
2	Identificación de aves mediante sonidos	-	X	-	-	-
3	Gratis	X	X	X	-	X
4	Disponible para iOS	-	X	-	-	-
5	Disponible para Android	X	X	-	-	-
6	Registro de observaciones	X	-	X	-	X
7	Generación de mapas de distribución		-	X	-	-
8	Información detallada sobre las aves	X	X	X	-	X
9	Uso de inteligencia artificial	X	X	-	X	-
10	Enfoque en aficionados	X	X	-	-	-
11	Enfoque en investigación científica	X	-	X	X	X
12	Plataforma en línea	-	-	X	-	-

Al analizar la Tabla VIII comparativa de los diferentes sistemas de identificación y monitoreo de aves, se observa que IdBird se destaca significativamente por su integración de múltiples funcionalidades clave. IdBird utiliza inteligencia artificial para identificar aves mediante imágenes, lo que facilita el proceso de identificación para los usuarios, similar a Merlin Bird ID y Birdsnap.

Una de las mayores fortalezas de IdBird es su capacidad para permitir el registro de observaciones, características que no están presentes en Merlin Bird ID y Birdsnap, pero que comparte con eBird. Estas funciones son cruciales para la investigación y conservación de aves, proporcionando datos valiosos y visualizaciones geográficas de las observaciones.

IdBird también proporciona información detallada sobre las aves y utiliza inteligencia artificial para la identificación de estas. Estas características la comparan favorablemente con otras herramientas en estas áreas. Además, IdBird es útil tanto para aficionados como para la investigación científica.

Marco teórico.

1. Sistema:

Un sistema es un conjunto de componentes y procesos que trabajan juntos para procesar y almacenar información. Estos componentes incluyen hardware, software y datos, cuya interacción permite que el sistema cumpla con su función específica [15].

En el presente proyecto, el sistema desarrollado es una aplicación móvil que utiliza sensores, como la cámara del celular, para registrar y procesar datos con el objetivo de llevar un conteo de las aves endémicas en la región de Zacatecas.

2. Aves:

Son un grupo de vertebrados endotérmicos y ovíparos pertenecientes a la clase Aves. Se caracterizan por tener un cuerpo cubierto de plumas, un pico córneo sin dientes, un esqueleto ligero y huesos huecos, un sistema respiratorio altamente eficiente que incluye sacos aéreos y un corazón con cuatro cámaras. Además, la mayoría de las aves tienen la capacidad de volar [16]. En el contexto del proyecto "IdBird", características como color, dimensiones, y rasgos distintivos, servirán para la identificación y clasificación de las aves mediante técnicas de procesamiento de imágenes.

Las aves que se van a utilizar para su identificación en este proyecto incluyen:

- **Carpintero de Arizona (Dryobates arizonae):** Es una especie de pájaro carpintero pequeño, de aproximadamente 20 cm de longitud. Tiene un plumaje principalmente negro con barras blancas en las alas y la espalda. Los machos presentan una mancha roja en la parte posterior de la cabeza [32].



Figura 4 Carpintero de Arizona

- **Papamoscas Pinero (Empidonax affinis):** Este pequeño pájaro cantor mide alrededor de 13 cm de longitud. Tiene un plumaje verde oliva en el dorso y partes inferiores más claras, casi blancas. Sus alas presentan dos barras alares claras. Los ojos están rodeados por un anillo blanco tenue. Este pequeño pájaro cantor mide alrededor de 13 cm de longitud. Tiene un plumaje verde oliva en el dorso y partes inferiores más claras, casi blancas. Sus alas presentan dos barras alares claras. Los ojos están rodeados por un anillo blanco tenue [33].



Figura 5 Papamoscas Pinero

- **Carbonero Mexicano (*Poecile sclateri*):** Mide alrededor de 12 cm de longitud. Tiene un plumaje gris en el dorso y blanco en las partes inferiores. La característica distintiva de esta especie es su gorra negra que cubre la cabeza hasta el cuello, y sus mejillas blancas. Los ojos son oscuros y el pico corto y cónico [34].



Figura 6 Carbonero Mexicano

- **Zacatonero Serrano (*Oreothlypis superciliosus*):** Con una longitud de aproximadamente 12 cm, esta ave presenta un plumaje verde oliva en el dorso y amarillo en las partes inferiores. Una característica distintiva es la franja blanca sobre el ojo, que contrasta con su cara oscura [35].



Figura 7 Zacatonero Serrano

- **Papamoscas Cardenalito (*Pyrocephalus rubinus*):** Los machos de esta especie son llamativos, con un plumaje rojo brillante en la cabeza, el pecho y el vientre, y un dorso marrón oscuro. Las hembras tienen un plumaje más discreto, de color marrón grisáceo con tonos rojizos en el pecho. Mide aproximadamente 14 cm de longitud [36].



Figura 8 Papamoscas Cardenalito

3. Niveles de Endemismo:

El endemismo se refiere a la condición de una especie que se encuentra restringida a una ubicación geográfica específica y no se encuentra naturalmente en ningún otro lugar. Los niveles de endemismo pueden variar desde endémico (exclusivo de una región particular) hasta casi endémico (presente en una región limitada, pero también en áreas adyacentes) [37].

- **Endémico:** Una especie que es exclusiva de una región geográfica específica. Por ejemplo, el Zzacatonero Serrano (*Oreothlypis superciliosus*) es una especie endémica de México, lo que significa que no se encuentra naturalmente en ningún otro lugar del mundo [38].
- **Cuasiendémico:** Una especie que se encuentra principalmente en una región, pero también puede estar presente en áreas adyacentes. En este proyecto, las siguientes aves son consideradas casi endémicas [39]:
 - Carpintero de Arizona (*Dryobates arizonae*)[40].
 - Papamoscas Pinero (*Empidonax affinis*)[41].
 - Carbonero Mexicano (*Poecile sclateri*)[42]
 - Papamoscas Cardenalito (*Pyrocephalus rubinus*)[43].

4. Cámara de celular:

Es un dispositivo que se encuentra en la mayoría de los teléfonos móviles modernos y que permite capturar imágenes y videos de alta calidad. Esta cámara utiliza una lente para enfocar la luz en un sensor de imagen, que convierte la luz en señales eléctricas que se procesan para formar una imagen digital. La calidad de la cámara del celular puede variar según el modelo, y generalmente se mide en megapíxeles. Otros factores que pueden influir en la calidad de la cámara son el tamaño de la lente, la apertura y el procesamiento de la imagen por software [17].

Para garantizar un desempeño óptimo de la aplicación "IdBird", se requiere un teléfono móvil de gama media-alta con las siguientes características mínimas:

Tabla IX Características mínimas de teléfonos de gama media-alta

Aspectos Diferenciadores de un Teléfono Móvil de Gama Media-Alta	
Procesador	Snapdragon serie 700, Mediatek Dimensity serie 800/900, o equivalentes. Octa-core, hasta 2.4 GHz, tecnología de 7nm o 6 nm para eficiencia y potencia.
Memoria y Almacenamiento:	4-8 GB Ram
Almacenamiento Interno:	128-256 Gb
Cámaras:	Características: Gran angular de 13 MP (f/1.8) con magnificación máxima de 0.21x, Ultra gran angular de 2 MP (f/2.2) con magnificación máxima de 0.21x, Lente de profundidad de 2 MP (f/2.4), Funciones: Enfoque automático, flash LED, geoetiquetado, HDR.

Estas características aseguran que el dispositivo tenga la capacidad de capturar y procesar las imágenes, lo cual es esencial para el funcionamiento de la aplicación de una manera optima. La cámara del celular en particular se utiliza para capturar las fotos o imágenes de las aves que serán procesadas para su identificación y clasificación dentro de la aplicación.

5. Procesamiento de imágenes:

Es el conjunto de técnicas y algoritmos que se utilizan para analizar y manipular imágenes digitales con el fin de mejorar su calidad, extraer información útil, realizar reconocimiento de patrones, entre otros objetivos. El procesamiento de imágenes se utiliza en una amplia gama de campos, como la medicina, la industria, la seguridad, la vigilancia, la ciencia, entre

otros. El procesamiento de imágenes implica el uso de herramientas y técnicas como filtrado, segmentación, detección de bordes, transformaciones geométricas, extracción de características, entre otras [18].

En "IdBird", el procesamiento de imágenes permite analizar las fotografías tomadas por los usuarios para identificar y clasificar las aves observadas.

6. Reconocimiento de patrones:

Es una rama de la inteligencia artificial que se enfoca en el análisis de datos y el aprendizaje automático para identificar patrones en los datos y hacer predicciones sobre nuevas entradas.

Se utiliza en una amplia gama de aplicaciones, como el reconocimiento de voz, el procesamiento de imágenes, el análisis de texto, la detección de fraudes, entre otros. El reconocimiento de patrones implica el uso de técnicas y algoritmos para extraer características de los datos, construir modelos estadísticos y de aprendizaje automático, y evaluar el rendimiento de estos modelos en conjuntos de datos de prueba [19]. En "IdBird", se emplearán técnicas de reconocimiento de patrones utilizando la librería TensorFlow Lite, para implementar el modelo MobileNet V3:

- **TensorFlow Lite:** Es una librería de TensorFlow diseñada para dispositivos móviles y embebidos. Ofrece un conjunto de herramientas y APIs que permiten la ejecución eficiente de modelos de aprendizaje automático en dispositivos con recursos limitados. TensorFlow Lite es conocido por su capacidad de optimizar modelos para reducir el tamaño y mejorar el rendimiento, lo cual es esencial para aplicaciones móviles que requieren procesar datos en tiempo real sin comprometer la precisión [44].



Figura 9 Logo de TensorFlow Lite

- **MobileNet V3:** Es un modelo de red neuronal convolucional optimizado para dispositivos móviles. MobileNet V3 combina técnicas avanzadas de arquitectura de redes neuronales y eficiencia computacional, permitiendo realizar tareas de

clasificación de imágenes con alta precisión y bajo consumo de recursos. Este modelo es ideal para ser implementado en aplicaciones móviles debido a su balance entre velocidad y precisión, lo cual es crucial para el reconocimiento de patrones en "IdBird" [45].

7. Sistema de inventario:

Es un conjunto de herramientas y técnicas que se utilizan para gestionar y controlar el inventario de una empresa. Estos sistemas ayudan a las empresas a optimizar sus procesos de inventario, reducir los costos de almacenamiento y mejorar la eficiencia operativa. Los sistemas de inventario pueden ser manuales o automatizados, y se utilizan en una amplia gama de sectores, como la manufactura, el comercio minorista, la logística, entre otros [20]. En el proyecto "IdBird", el sistema desarrollado es una herramienta automatizada que facilita el registro y monitoreo de aves, así como facilitación de la creación de bitácoras y muestras, haciendo mas accesible este proceso para los biólogos.

8. Lenguajes de programación:

Un lenguaje de programación es un conjunto de reglas y sintaxis que permite a los desarrolladores escribir programas que pueden ser ejecutados por una computadora. Los lenguajes de programación son esenciales para el desarrollo de software y aplicaciones, ya que permiten la creación de algoritmos y la gestión de datos de manera eficiente [46].

En el proyecto "IdBird", se utilizarán los siguientes lenguajes de programación:

- **Kotlin:** Kotlin es un lenguaje de programación moderno y conciso que se utiliza principalmente para el desarrollo de aplicaciones Android. Puede trabajar junto con Java y ofrece características avanzadas que mejoran la productividad del desarrollador. En "IdBird", Kotlin se utilizará para desarrollar la interfaz de usuario y la lógica de la aplicación móvil [47].



Figura 10 Logo de Kotlin

- **SQL:** SQL (Structured Query Language) es un lenguaje estándar para gestionar bases de datos relacionales. Permite la creación, modificación y consulta de datos de manera eficiente. En "IdBird", SQL se utilizará para manejar la base de datos local, que almacenará las bitácoras y los datos de muestreo. La naturaleza relacional de SQL es ideal para la organización y el acceso rápido a grandes volúmenes de datos estructurados [48].



Figura 11 Logo de SQL

- **Python:** Python es un lenguaje de programación de alto nivel conocido por su simplicidad y versatilidad. Se utiliza ampliamente en ciencia de datos, inteligencia artificial y desarrollo web. En "IdBird", Python se utilizará para desarrollar el sistema de reconocimiento de patrones, aprovechando librerías como TensorFlow, NumPy.. Estas librerías permitirán implementar y optimizar los modelos de clasificación de aves [49].



Figura 12 Logo de python

Estos lenguajes de programación se seleccionaron por su capacidad para satisfacer los requisitos técnicos y de rendimiento del proyecto.

9. Base de Datos:

Una base de datos es un conjunto organizado de datos que se almacena y gestiona electrónicamente. Las bases de datos permiten almacenar grandes cantidades de información de manera estructurada y eficiente, lo que facilita la recuperación, actualización y análisis de datos [50]. En el proyecto "IdBird", se utilizarán dos tipos de bases de datos:

- **SQLite:** SQLite es una biblioteca de software que proporciona una base de datos relacional ligera, ideal para aplicaciones móviles. Permite la creación, modificación y consulta de datos de manera eficiente [51]. En "IdBird", SQLite se utilizará para manejar la base de datos local, que almacenará las bitácoras y los datos de muestreo. La naturaleza relacional de SQLite,



Figura 13 Logo de SQLite

- **MongoDB:** MongoDB es una base de datos NoSQL orientada a documentos que permite el almacenamiento de datos en un formato flexible y escalable. MongoDB es especialmente adecuado para aplicaciones que requieren alta disponibilidad y escalabilidad horizontal [52]. En "IdBird", MongoDB se utilizará para almacenar la información de los usuarios que se registren en la aplicación. Esto incluye datos como nombres de usuario (nombre y apellido), contraseñas y correos electrónicos.



Figura 14 Logo de MongoDB

10. API:

Una API (Interfaz de Programación de Aplicaciones) es un conjunto de definiciones y protocolos que permite a diferentes aplicaciones de software comunicarse entre sí. Las APIs son esenciales para la integración de sistemas, permitiendo el intercambio de datos y la funcionalidad entre diferentes aplicaciones de manera segura y eficiente [53].

En el proyecto "IdBird", se utilizarán dos APIs principales:

- Postage App: Postage App es un servicio de envío de correos electrónicos que ofrece una API para validar y confirmar correos electrónicos. En "IdBird", Postage App se utilizará para enviar correos electrónicos de verificación a los usuarios que se registren en la aplicación. Esto asegura que las direcciones de correo electrónico proporcionadas sean válidas y que los usuarios puedan confirmar su registro de manera segura.



Figura 15 Logo de PostageApp

- MongoDB API: Se utilizará la API de MongoDB para enviar y gestionar los datos de los usuarios, incluyendo nombre de usuario, contraseña encriptada y correo electrónico. Esta API facilitará la comunicación entre la aplicación móvil y la base de datos, asegurando que la información de los usuarios se maneje de manera eficiente y segura.

11. Aplicación móvil:

Es un software diseñado para ser utilizado en dispositivos móviles como teléfonos inteligentes y tabletas. Estas aplicaciones se ejecutan en el sistema operativo del dispositivo móvil y pueden ser descargadas a través de una tienda de aplicaciones. Las aplicaciones móviles se utilizan para una amplia variedad de propósitos, como entretenimiento, comunicación, educación, negocios, finanzas, entre otros [21]. La aplicación "IdBird" se desarrollará para dispositivos Android, permitiendo a los usuarios capturar imágenes de aves, procesarlas y obtener información detallada sobre las especies. Los usuarios podrán tomar fotos de las aves con sus dispositivos, y la procesará las imágenes, identificará las especies y proporcionará información sobre cada una de ellas.

12. Modelo V:

El modelo V es un modelo de desarrollo de sistemas diseñado para simplificar la complejidad asociada con el desarrollo en ingeniería de sistemas. Se utiliza para definir un procedimiento uniforme para el desarrollo de productos o proyectos. El modelo V es un proceso de desarrollo de software que se puede suponer que es la extensión del modelo en casaca. El modelo V demuestra las relaciones entre cada fase del ciclo de vida del desarrollo y su fase asociada de pruebas [22].

En lugar de moverse hacia debajo de forma lineal, los pasos del proceso se doblan hacia arriba después de la fase de codificación, para formar la típica forma de V, como se ilustra en la Figura 1. El propósito del modelo V es mejorar la eficiencia y calidad del desarrollo de software y reflejar la relación entre las actividades de prueba y las actividades de desarrollo. Las pruebas de software son demasiado importantes como para dejarlas para el final del proyecto, y el modelo V de pruebas incorpora las pruebas en todo el ciclo de vida del desarrollo de software [23].

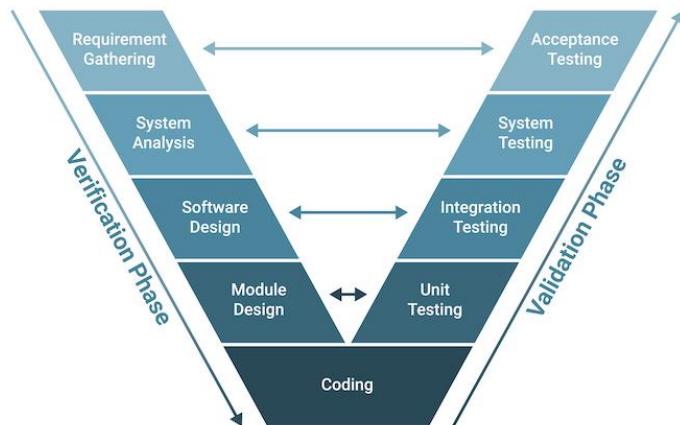


Figura 16 Representación gráfica del modelo V

13. Arquitectura C4

La arquitectura C4 significa contexto, contenedores, componentes y código: un conjunto de diagramas jerárquicos que puede usar para describir su arquitectura de software en diferentes niveles de zoom [24].

La arquitectura C4 se utiliza para describir la arquitectura del software en diferentes niveles de abstracción, proporcionando una visión clara y comprensible de cómo se estructura el sistema.

Este modelo evita que la documentación de la arquitectura sea compleja y confusa, así como evita que tenga poca información [27], se emplea para describir la arquitectura del software en diferentes niveles jerárquicos de abstracción:

- Nivel 1: Diagrama de contexto del sistema: Sirve para mostrar las interacciones de forma macro, sin mucho detalle, centrándose en las comunicaciones y dependencias entre los sistemas y los usuarios que componen e interactúan con el software.
- Nivel 2: Diagrama de contenedores: Sirve para describir sus contenedores y cómo se comunican/interactúan, se hace hincapié en la arquitectura y las tecnologías utilizadas.
- Nivel 3: Diagrama de componentes: Se detallan los componentes clave dentro de cada contenedor y se describen de las partes que lo componen.
- Nivel 4: Diagrama de clases o código: Explica cómo se implementa cada componente y para ello utiliza el diagrama de clases UML, se usa para una vista detallada del código fuente.

14. Patrón de diseño MVC(Model-View-Controller):

El patrón de diseño MVC (Model-View-Controller) es un patrón de diseño de software comúnmente utilizado para implementar interfaces de usuario, datos y lógica de control [25].

El patrón de diseño Modelo-Vista-Controlador (MVC) se emplea para organizar el código de la aplicación, separando la lógica de negocio, la interfaz de usuario y el control de entrada [26]. Se utiliza para organizar el código de la aplicación, usado en el diseño de la arquitectura, separando la lógica de negocio (Model), la interfaz de usuario (View) y el control de entrada (Controller). Esta separación facilita el mantenimiento del código y mejora la escalabilidad y reutilización de los componentes.

Las ventajas de utilizar MVC incluyen:

1. **Separación de Preocupaciones:** MVC separa la aplicación en tres componentes principales: el Modelo (gestión de datos y lógica de negocio), la Vista (interfaz de usuario) y el Controlador (manejo de la entrada del usuario y la lógica de flujo). Esta separación facilita el desarrollo y el mantenimiento del código, permitiendo que cada componente se desarrolle y se pruebe de manera independiente.
2. **Reutilización de Código:** La estructura MVC permite una mayor reutilización de código. Por ejemplo, las vistas pueden reutilizarse para diferentes controladores, y los modelos pueden ser utilizados por múltiples controladores. Esto reduce la duplicación de código y mejora la eficiencia del desarrollo.
3. **Facilidad de Mantenimiento y Escalabilidad:** Con MVC, es más sencillo localizar y corregir errores, ya que cada componente está claramente definido. Además, esta estructura modular facilita la adición de nuevas funcionalidades sin afectar negativamente al sistema existente.

Marco Metodológico.

Metodologías

La metodología seleccionada para el desarrollo de este proyecto es el modelo V. Este modelo se adapta adecuadamente a las necesidades del proyecto, ya que combina un enfoque estructurado y secuencial con la posibilidad de realizar pruebas de validación desde las primeras fases del desarrollo.

El modelo V garantiza que cada componente del software sea rigurosamente validado antes de proceder a la siguiente etapa. En cada fase del ciclo de vida del desarrollo del software, se realiza una fase correspondiente de pruebas y validación, lo que asegura que los errores se identifiquen y se corrijan de manera temprana, en lugar de acumularlos para el final del desarrollo.

Además, el modelo V facilita una mejor comprensión y seguimiento del progreso del proyecto, al proporcionar una estructura clara y bien definida. La implementación de esta modelo contribuye a asegurar que el sistema desarrollado cumple con los requisitos especificados y que cada componente funcione correctamente para integrarse en el sistema.

Adicionalmente se ha optado por una el uso de la arquitectura C4 y el patrón de diseño MVC, ya que esta combinación ofrece un enfoque integral que asegura que cada etapa del desarrollo esté validada y verificada, garantizando la calidad del producto final.

Los diagramas de la arquitectura C4 facilitan la comunicación entre los miembros del equipo y las partes interesadas, proporcionando una representación visual clara de la arquitectura del sistema y ayudando a todos a entender cómo se organiza y cómo interactúan sus componentes. Al descomponer la arquitectura del software en niveles manejables de detalle, facilita el enfoque en diferentes aspectos del sistema en cada fase del desarrollo y ayuda a gestionar la complejidad, haciendo más sencillo identificar y abordar cambios o mejoras necesarias sin afectar negativamente al sistema global.

El patrón de diseño MVC complementa esta estructura organizando el código de manera modular, mejorando la eficiencia en el desarrollo y mantenimiento del sistema, permitiendo que los cambios en una parte del sistema no afecten negativamente a otras partes.

La combinación del modelo V, la arquitectura C4 y el patrón MVC proporciona una metodología robusta que mejora la claridad, estructura, calidad y eficiencia del desarrollo del sistema, asegurando que se cumplan los requisitos del proyecto de manera efectiva y sostenible. La validación continua y la estructura clara y comprensible de la arquitectura del software garantizan un desarrollo de alta calidad y la implementación eficiente de funcionalidades como el reconocimiento de imágenes y la georreferenciación.

Herramientas y Técnicas:

Para el desarrollo del proyecto, se planea utilizar Kotlin como lenguaje de programación para la creación de la aplicación móvil, y Python para desarrollar las APIs necesarias y conectar con el servicio de PostageApp y así como entrenar el modelo de reconocimiento de imágenes.

En cuanto a frameworks y librerías, se utilizará TensorFlow Lite en conjunto con el modelo de redes neuronales convolucionales (CNN) MobileNet v3 para el procesamiento de imágenes y reconocimiento de patrones.

Además, se gestionarán las tareas mediante un tablero Kanban para la organización y seguimiento durante la implementación, complementado con un cronograma para la distribución de tareas a lo largo del proyecto.

Procedimientos y Actividades

1. Planificación del Proyecto

- Elección de Metodología:
 - Revisión de metodologías disponibles conforme al proyecto, entre ellas modelo cascada, modelo V y metodologías agiles como modelo SCRUM.
 - Lectura y selección de la metodología adecuada, tras la investigación de las diferentes metodologías se determinó que el modelo V basada en las necesidades del proyecto era el más apropiado.
- Planificación de Cronograma:
 - Creación y revisión del cronograma del proyecto conforme a la metodología elegida. Así como su validación del cronograma con el equipo y los asesores.

2. Análisis de Requerimientos

- Especificación de Requerimientos de Software (SRS):
 - Levantamiento de requisitos con el cliente para determinar las necesidades principales.

- Análisis y documentación de los requisitos obtenidos con el equipo de trabajo.
- Revisión y corrección del documento SRS y validación del documento final con el equipo y asesores.
- Diseño de Prototipos:
 - Creación y revisión de prototipos iniciales de la interfaz de usuario.
 - Corrección y verificación de los prototipos con el cliente para comprobar que su visión del proyecto sea plasmada en su realización.
- Matriz de Trazabilidad:
 - Creación de la matriz de trazabilidad para asegurar la relación entre requisitos, diseño y pruebas.
 - Revisión, actualización y validación de la matriz de manera continua conforme se completaban los documentos que la componen.
- Plan de Riesgos:
 - Identificación y evaluación de riesgos potenciales que se pueden presentar a lo largo del proyecto.
 - Documentación y validación de estrategias de prevención y de mitigación a aplicar en caso necesario.

3. Diseño del Sistema

- Diseño de la Arquitectura:
 - Creación de diagramas de contexto del sistema, contenedores, componentes y código siguiendo el modelo de modelo C4.
 - Revisión y validación de los diagramas.
- Diseño de Base de Datos y Manejo de Archivos:
 - Creación de diagramas de la base de datos y especificación del manejo de archivos.

- Revisión, corrección y verificación de los diseños.
- Diseño Detallado:
 - Creación de diagramas UML de comportamiento, para el proyecto se realizaron diagramas de casos de uso, diagramas de secuencia y diagramas de actividades.
 - Desarrollo del plan de pruebas de acuerdo a cada uno de los casos de uso.
 - Revisión y verificación de los diagramas y plan de pruebas.

Las siguientes secciones se realizarán en la segunda parte del trabajo terminal.

4. Desarrollo del Sistema

- Selección de tareas:
 - Selección y distribución de tareas conforme a su dificultad y prioridad.
- Codificación:
 - Desarrollo de cada módulo de manera progresivo asegurando que cada parte funcione correctamente.
- Pruebas Unitarias e Integración:
 - Ejecución de pruebas unitarias para cada componente.
 - Realización de pruebas de integración para verificar la correcta interacción entre los componentes.
- Corrección de Errores:
 - Identificación y corrección de errores detectados durante las pruebas unitarias e integración.

5. Pruebas de Sistema:

- Ejecución de pruebas completas del sistema para asegurar el cumplimiento de todos los requisitos.

6. Despliegue del Sistema

- Manual de Usuario, Operaciones y Mantenimiento:
 - Creación, revisión y validación de los manuales de usuario, operaciones y mantenimiento.
 - Preparación del sistema para su despliegue final.

7. Documentación y Presentación

- Reporte Final TT1:
 - Creación y revisión de múltiples versiones del reporte final.
 - Validación, ensayos y presentación del reporte.
- Implementación y Validación Final:
 - Despliegue final del sistema en el entorno de producción.
 - Validación final del sistema con todas las funcionalidades implementadas.

Análisis y Discusión de los Resultados

Análisis de la selección de la metodología (TT-I):

En esta sección se analizan las ventajas y desventajas resultantes de la implementación de la metodología elegida para la fase de documentación del "Sistema para muestreo de aves en la ciudad de Zacatecas". La metodología seleccionada combina el modelo V, el arquitectura c4 y el patrón de diseño MVC, como se puede ver en el documento de Marco Metodológico y Plan de Proyecto en el Anexo B. Aunque hasta ahora no se ha realizado ningún desarrollo de software, la documentación ha seguido estos modelos, y se evaluará cómo han influido en esta etapa del proyecto.

Ventajas de la Implementación de la Metodología:

1. Estructura y Organización Clara:

- Modelo V: La aplicación del modelo V en la fase de documentación ha permitido una estructuración clara y secuencial de los documentos. Cada fase de la documentación ha sido acompañada de su correspondiente fase de validación, asegurando que cada componente de la documentación cumpla con los requisitos establecidos.

2. Comunicación Eficaz:

- Arquitectura c4: Los diagramas de la arquitectura c4 han sido esenciales para proporcionar una visión clara de la arquitectura propuesta del sistema. Estos diagramas han facilitado la comunicación entre los miembros del equipo y con las partes interesadas, asegurando una comprensión común y evitando malentendidos.

3. Claridad en la Definición de Roles y Responsabilidades:

- Patrón de diseño MVC: Aunque el patrón MVC se aplicará principalmente en la fase de desarrollo, su consideración en la documentación ha permitido una clara definición de los componentes del sistema y sus interacciones. Esto ha facilitado la planificación y organización del trabajo futuro.

La combinación de la arquitectura y el modelo ha proporcionado una base sólida para la fase de desarrollo (TT-I). La documentación detallada y validada asegura que el equipo esté bien preparado para iniciar el desarrollo, con una comprensión clara de los requisitos y la arquitectura del sistema.

Desventajas de la Implementación de la Metodología:

1. Complejidad en la Documentación:

- La integración de múltiples metodologías ha añadido un nivel adicional de complejidad a la documentación. Asegurarse de que todos los miembros del equipo comprendan y sigan las metodologías elegidas ha requerido un esfuerzo considerable

en términos de planificación y coordinación, así como de investigación y capacitación respecto a cada metodología.

2. Requerimientos de Tiempo:

- La necesidad de realizar una validación continua y documentar detalladamente cada fase ha incrementado el tiempo necesario para completar la documentación. Esto ha sido un desafío, especialmente en términos de cumplir con los plazos establecidos.

3. Curva de Aprendizaje:

- La adopción de estos modelos ha implicado una curva de aprendizaje para los miembros del equipo. La familiarización con los conceptos y herramientas específicos de cada modelo ha requerido tiempo adicional para capacitación y adaptación.

A pesar de las desventajas mencionadas, la elección de estos modelos ha sido beneficiosa para la fase de documentación del proyecto. La combinación del modelo V, el arquitectura c4 y el patrón MVC ha proporcionado una estructura robusta y clara para la documentación, asegurando la calidad y facilitando la comunicación.

La documentación detallada y bien estructurada ha preparado al equipo para la fase de desarrollo, proporcionando una base sólida sobre la cual construir. Aunque la implementación de estas metodologías ha requerido un esfuerzo significativo, los beneficios obtenidos en términos de claridad, organización y preparación justifican plenamente la elección de estos modelos.

Gestión del proyecto

1. Plan del proyecto.

El cronograma presentado es el plan detallado realizado para la fase de Trabajo Terminal 1 del proyecto. Durante esta fase, se llevaron a cabo la planificación del proyecto, el análisis de requerimientos y el diseño del sistema, además de horas adicionales planificadas para la capacitación del equipo.

El plan inicial establecía un régimen de trabajo de 6 horas diarias, divididas en 3 horas por la mañana y 3 horas por la noche, durante 4 días a la semana, de lunes a jueves, además de 1 hora los viernes de cada dos semanas para reuniones. Este horario se mantendría a lo largo de 5 meses, desde febrero hasta junio, proporcionando un total máximo de 490 horas de trabajo por persona, debido a que el equipo está conformado por dos personas da un total de 980 de horas en total máximo.

Según el cronograma creado, se estimaba un total de 791 horas de trabajo, que debían ser divididas entre los dos integrantes del equipo, dependiendo de la tarea a realizar y permitiendo una distribución equitativa de la carga de trabajo. Fueron distribuidas entre las siguientes actividades:

- **Plan de Proyecto:** 57 horas
- **Análisis de Requerimientos:** 216 horas
- **Diseño del Sistema:** 348 horas
- **Capacitación del Equipo:** 60 horas
- **Reporte Final TT1:** 110 horas

Sin embargo, al finalizar el semestre de TT1 el equipo completó un total de 422 horas de trabajo, no obstante al ser un proyecto de dos individuos el esfuerzo por persona duplica las horas de trabajo dándonos un total de 844 horas , que distribuidas de la siguiente manera.

- **Plan de Proyecto:** 47 horas
- **Análisis de Requerimientos:** 88 horas

- **Diseño del Sistema:** 175 horas
- **Reporte Final TT1:** 82 horas
- **Investigación y Capacitación:** 30 horas

La Figura 2 demuestra las horas estimadas al inicio del proyecto y su comparación con las horas registradas al finalizar la parte del proyecto trabajada en trabajo terminal I.



Figura 17 Evidencia de las horas estimadas y registradas en el proyecto

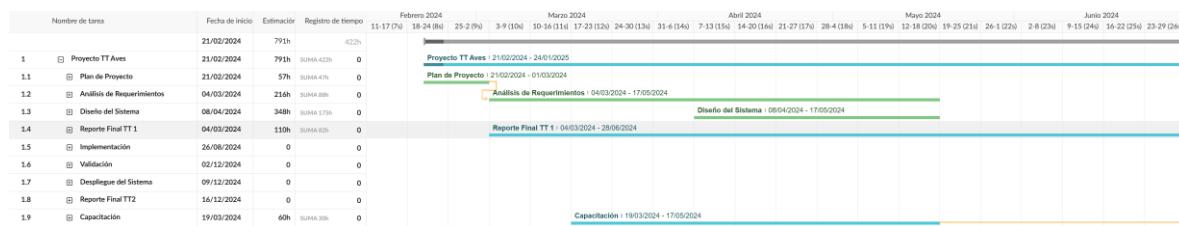


Figura 18 Evidencia de las horas estimadas y registradas por etapa del proyecto

La diferencia de horas de 791 a 422 se realizó principalmente a que el equipo de trabajo realizaba diferentes tareas de manera simultánea. Este total fue significativamente menor que las 791 horas estimadas inicialmente, lo que permitió ahorrar casi la mitad del tiempo planeado. Esta estrategia de trabajo en paralelo fue importante para optimizar el uso del tiempo y asegurar que las actividades se completaran dentro del plazo establecido.

Como se puede observar en la Figura 3, se muestran las actividades realizadas para cada una de las etapas de plan de proyecto, análisis de requerimientos, diseño del sistema y reporte final TT1, así como las horas estimadas y realizadas de cada una.

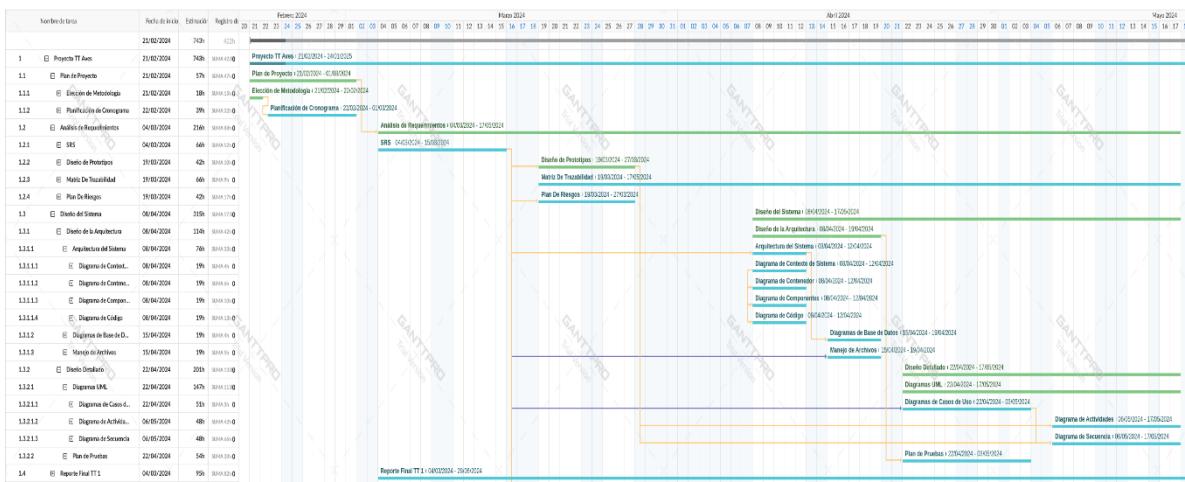


Figura 19 Plan de proyecto de TTI completo

2. Manejo de desviaciones en la ejecución del plan.

Durante la realización del proyecto se encontraron retrasos y desviaciones del plan de proyecto que se tenía establecido inicialmente. Estas desviaciones se presentaron principalmente en la parte de diseño, ya que se tenía contemplado una semana para los diagramas de arquitectura, una semana para los diagramas de base de datos y manejo de archivos, dos semanas para el diagrama de casos de uso y el plan de pruebas y dos semanas extras para los diagramas de actividades y de secuencia, ilustrado por la Figura 4 donde se aprecian la cantidad de horas estimadas y las horas realizadas, así como las semanas en las que se desempeñaron las actividades.

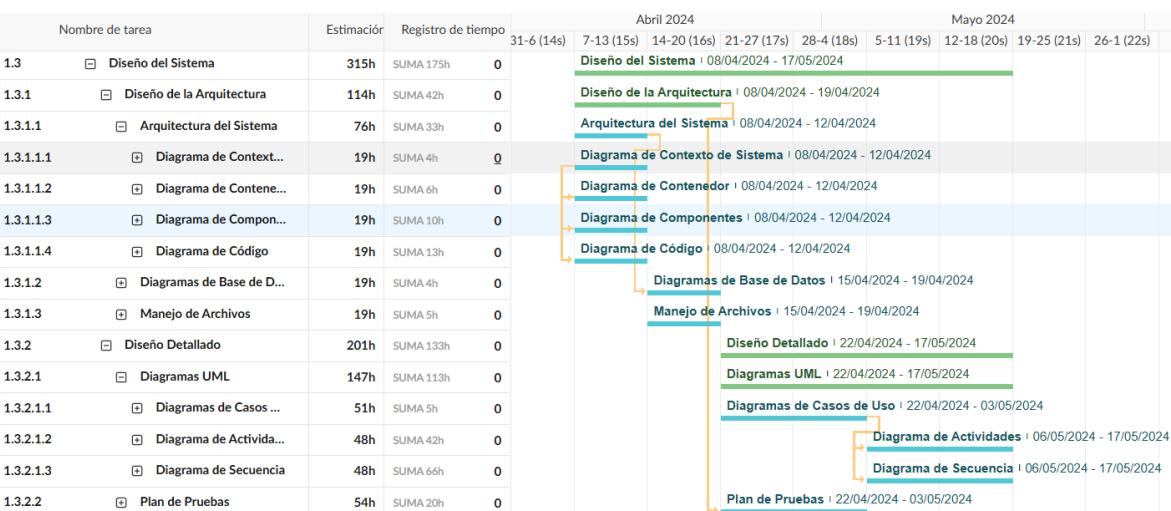


Figura 20 Evidencia de las desviaciones del proyecto

Sin embargo, en la realización de cada uno de los diagramas se percibió que los tiempos previamente contemplados eran erróneos en la mayoría de los diagramas. Puesto que para la arquitectura del sistema fue más extensa de lo esperado y en total tardó dos semanas en completarse, lo que llevó a un desplazamiento de una semana a los demás diagramas. Sin embargo, se pudo recuperar esta semana de atraso al realizar los diagramas de base de datos, manejo de archivos y casos de uso de manera rápida y eficaz.

Otra desviación que se tuvo fue con los diagramas UML, específicamente los diagramas de actividad y secuencia, que en vez de llevarse dos semanas simultáneas se realizaron en semana y media cada uno iniciando con los diagramas de actividad para posteriormente elaborar los diagramas de secuencia. Al desconocer los pasos y los formatos necesarios para un diagrama de secuencia pues no se contaba con la experiencia necesaria, fue necesario emplear más tiempo para la capacitación y los intentos fallidos, hasta conseguir una versión de calidad para describir los procesos del sistema.

Por último, se elaboró el plan de pruebas al finalizar todos los diagramas de diseño, finalizando todo el diseño del sistema para la semana del 19 al 25 de mayo y dar comienzo al reporte final de trabajo terminal 1.

Por otro lado, para la realización del reporte final se había contemplado realizar avances conforme a la finalización de cada bloque de documentos, sin embargo, por cuestión de tiempo y múltiples pruebas y validaciones no se fueron agregando al tiempo y se prefirió agregar al reporte cuando se hubieran completados todos los documentos y se haya validado de nueva cuenta los documentos producidos en conjunto para garantizar la calidad y alineación a los objetivos del proyecto. Esta estimación se ilustra en la Figura 5, donde las etapas de creación y sus versiones se omitieron al esperar a completar los documentos previos y agregarlos al reporte de manera total en la etapa considerada para corrección.

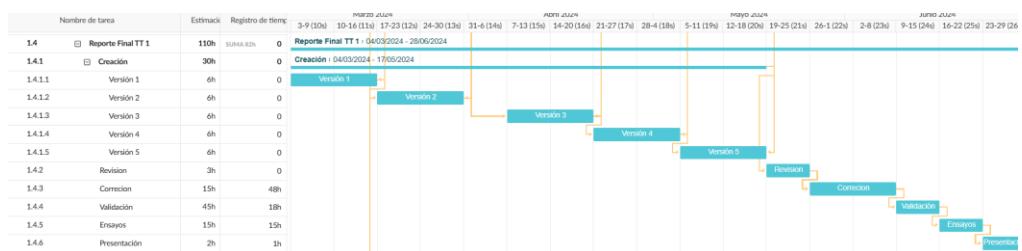


Figura 21 Evidencia de las desviaciones del reporte de proyecto

3. Plan de los riesgos del proyecto.

El manejo de riesgos del proyecto se centró en identificar, evaluar y mitigar los posibles eventos que pudieran afectar negativamente el desarrollo del proyecto. A lo largo del ciclo de vida del proyecto, se implementaron diversas estrategias para gestionar eficazmente los riesgos, con especial atención a aquellos que se detonaron y requirieron acciones específicas para su mitigación, como se pueden ver en el documento de Plan de Riesgos en el Anexo D.

Uno de los principales riesgos que ocurrieron durante el proyecto fue la falta de asistencia en ciertos días de trabajo por parte de algunos miembros del equipo. Este riesgo tenía el potencial de retrasar el progreso del proyecto debido a la ausencia del equipo de trabajo. Para mitigar este riesgo cuando se presentó, se replanificaron las actividades considerando el tiempo disponible y la cantidad de tareas pendientes. Esto implicó una redistribución de las responsabilidades entre los miembros presentes, asegurando que las actividades críticas continuaran sin interrupciones.

Otro riesgo que se presentó fue el conflicto de puntos de vista del proyecto en el equipo de trabajo sobre detalles del diseño y técnicas para realizarlos. Para resolver estos conflictos, se implementó un consenso que involucraba las opiniones del director, asesor y cliente, asegurando que todas las partes interesadas tuvieran la oportunidad de expresar sus preocupaciones y sugerencias. Este enfoque colaborativo facilitó la llegada a un acuerdo común, alineando los objetivos del equipo y mejorando la cohesión y la cooperación entre los miembros.

Desarrollo del proyecto

1. Resumen del análisis del sistema.

En esta sección se presentará un resumen del análisis realizado para el "Sistema para muestreo de aves en la ciudad de Zacatecas". Se incluyen la representación gráfica de los requerimientos de usuario y una síntesis de los requerimientos establecidos en la Especificación de Requerimientos de Software (SRS) mostrado en el Anexo C.

Representación Gráfica de los Requerimientos de Usuario (Diagrama Conceptual):

El diagrama conceptual es una representación visual de los principales requerimientos del usuario y cómo estos se interrelacionan dentro del sistema. A continuación en la Figura 6 se presenta el diagrama conceptual del sistema:

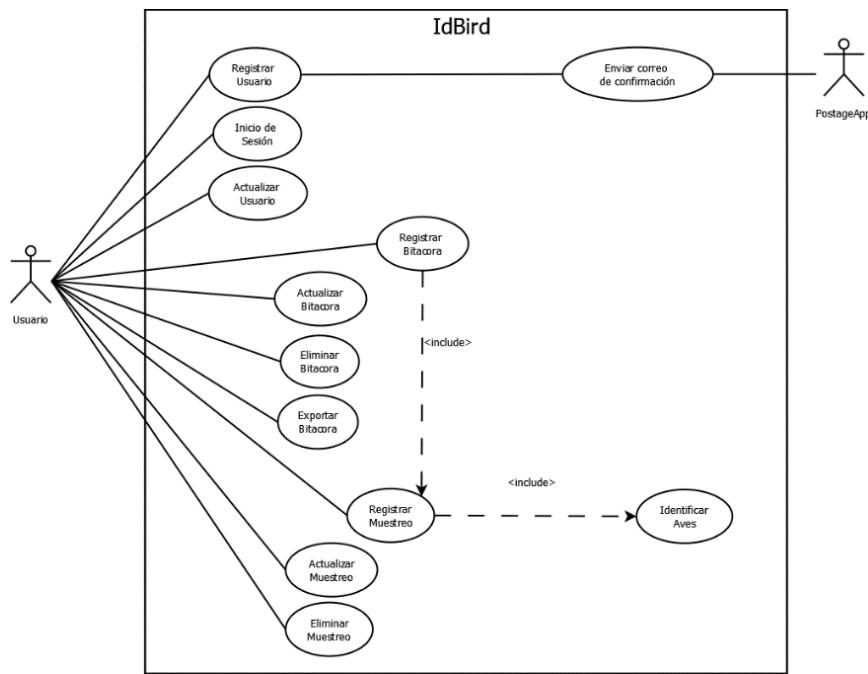


Figura 22 Diagrama de casos de uso

El diagrama describe como el usuario interactúa con el sistema "IdBird" a través de múltiples funcionalidades, tales como registrar usuarios, iniciar sesión, actualizar usuarios, registrar, actualizar y eliminar bitácoras, así como exportar bitácoras. También incluye la funcionalidad de registrar, actualizar y eliminar muestreos, y la identificación de aves. Ademas que PostageApp es el sistema encargado de enviar el correo de confirmación al usuario.

Los Requerimientos Establecidos (SRS):

El documento de Especificación de Requerimientos de Software (SRS) detalla todos los requerimientos funcionales y no funcionales necesarios para el desarrollo del sistema. A continuación en la Tabla III se presenta un resumen de los principales requerimientos establecidos:

Tabla Tabla de requerimientos

ID	Nombre del requerimiento	Tipo de requerimiento
RF-01	Registro, Inicio y Actualización de Usuarios	Funcional
RF-02	Registro, Actualización y Eliminación de Bitácoras de Campo	Funcional
RF-03	Registro, Actualización y Eliminación de Muestreos.	Funcional
RF-04	Identificación de Aves	Funcional
RF-05	Exportación de Bitácoras de Campo	Funcional
RNFD-01	Preprocesamiento de Imágenes	No funcional de desempeño
RNFD-02	Localización en tiempo real	No funcional de desempeño

Requerimientos Funcionales:

- RF-01: Registro y Actualización de Usuarios:
Permite a los usuarios registrarse y actualizar su información personal.
- RF-02: Registro, Actualización y Eliminación de Bitácoras de Campo:
Permite a los usuarios crear, actualizar y eliminar bitácoras de observación de aves.
- RF-03: Registro, Actualización y Eliminación de Muestreos:
Permite a los usuarios registrar, actualizar y eliminar datos de muestreo de aves.
- RF-04: Identificación de Aves:
Permite identificar aves a partir de imágenes capturadas con la cámara del celular.
- RF-05: Exportación de Bitácoras de Campo:
Permite exportar las bitácoras de observación de aves para su análisis o archivo.

Requerimientos No Funcionales

- RNFD-01: Preprocesamiento de Imágenes;
Implica el uso de técnicas de procesamiento de imágenes para mejorar la calidad y precisión de la identificación de aves.
- RNFD-02: Localización en Tiempo Real:
Permite registrar la ubicación geográfica de las observaciones de aves en tiempo real.

2. Diseño del sistema.

a. Arquitectura del sistema.

Para la representación gráfica de la arquitectura del sistema "IdBird" se ha utilizado el arquitectura c4, que descompone la arquitectura en cuatro niveles de abstracción: Contexto, Contenedores, Componentes y Clases. A continuación, se describen cada una de las imágenes proporcionadas para cada nivel.

Diagrama de Contexto:

El diagrama de contexto muestra la interacción del usuario con la aplicación móvil "IdBird". En esta vista de alto nivel, el usuario es la principal entidad interactuante que utiliza la aplicación móvil para el muestreo e identificación de aves en la ciudad de Zacatecas. La aplicación móvil "IdBird" es el sistema principal que proporciona estas funcionalidades al usuario, ilustrado en la Figura 7. En este nivel, se destaca la simplicidad y el enfoque en la interacción principal del usuario con el sistema.

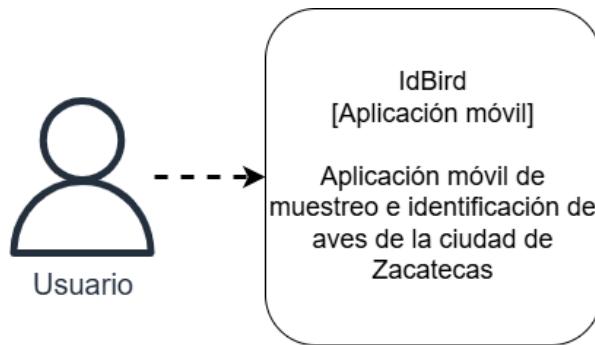


Figura 23 Diagrama de contexto

Diagrama de Contenedor:

El diagrama de contenedores descompone el sistema en sus principales contenedores de software, mostrando cómo interactúan entre sí. En esta vista, se identifica la aplicación móvil desarrollada en Kotlin y utilizando el patrón de diseño MVC. Como se puede observar en la Figura 8, la aplicación móvil se comunica con una API de correos desarrollada en Node.js y Express, que a su vez se conecta con el sistema de envío de correos PostageApp. La API de

correo maneja la verificación del registro de usuarios a través del envío de correos electrónicos. La base de datos MongoDB se utiliza para almacenar las credenciales de los usuarios y confirmar su registro. Esta vista destaca la estructura del sistema y la interacción entre sus principales contenedores.

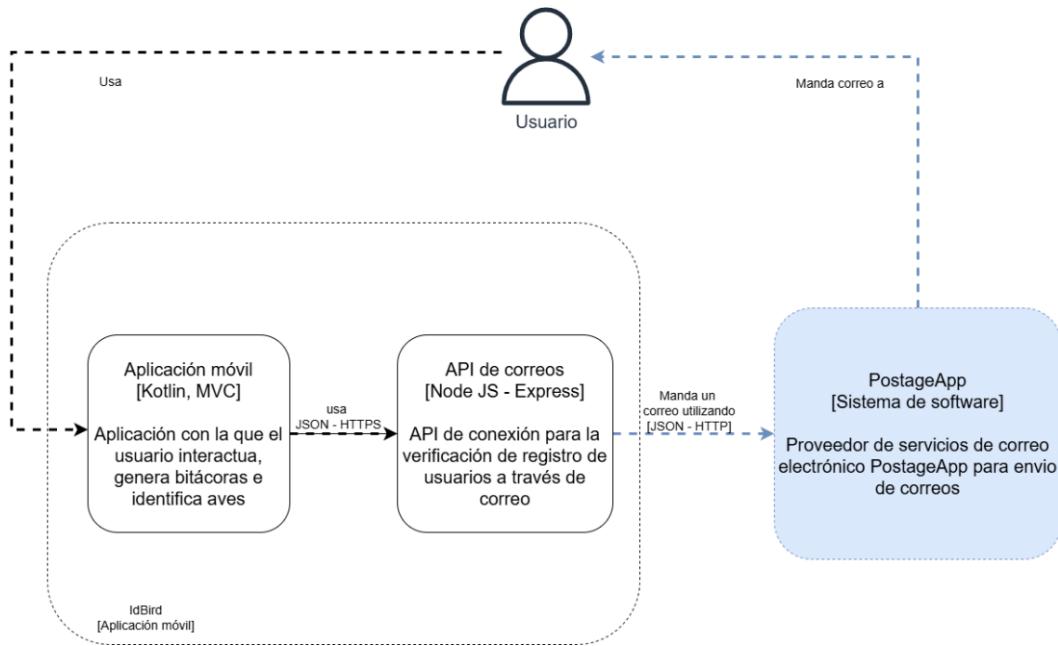


Figura 24 Diagrama de contenedores

Diagrama de Componentes:

El diagrama de componentes proporciona una visión detallada de los componentes internos de cada contenedor identificado en la vista de contenedores. La Figura 9 detalla cada uno de los componentes de manera visual para una mejor compresión. Para la aplicación móvil, se detallan los componentes MVC: Modelos, Vistas y Controladores. Cada uno de estos componentes maneja aspectos específicos de la aplicación, como la gestión de usuarios, bitácoras y muestras, así como la identificación de aves. La API de correos está dividida en componentes que manejan la autenticación y el envío de correos, utilizando la base de datos para gestionar la información del usuario. Esta vista permite comprender cómo se organiza el software internamente y cómo los componentes interactúan entre sí para proporcionar las funcionalidades del sistema.

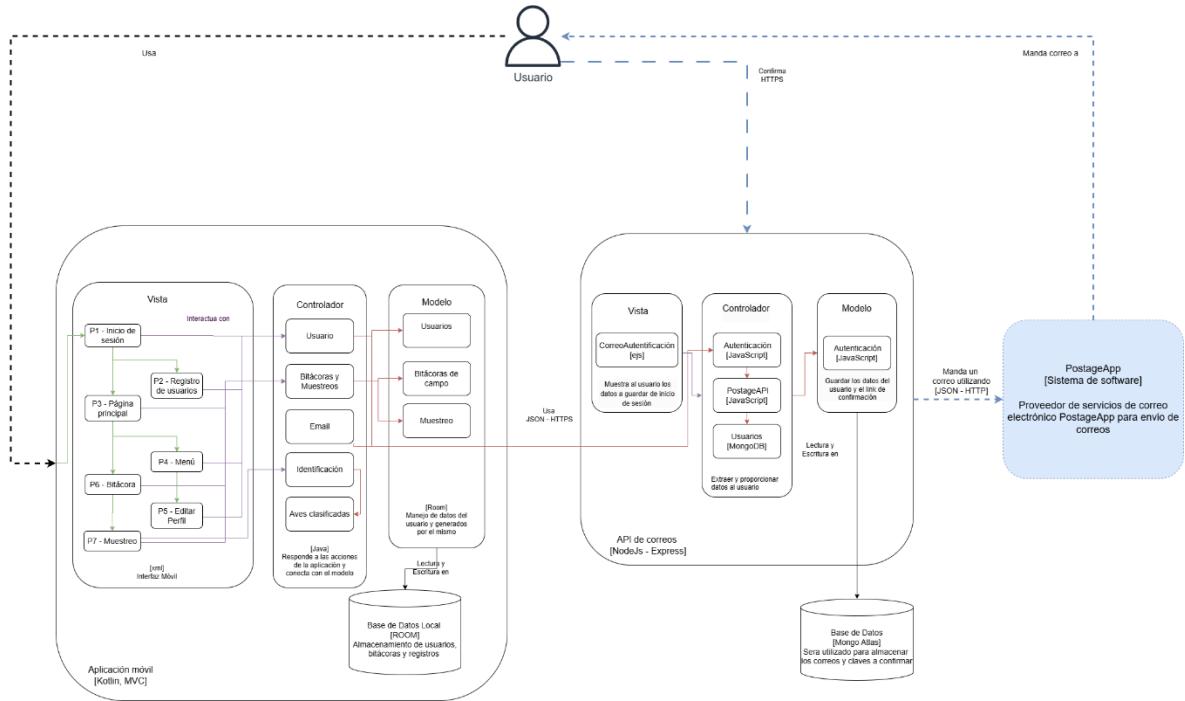


Figura 25 Diagrama de componentes

Diagrama de Clases:

El diagrama de clases detalla a nivel de código, mostrando las principales clases dentro de cada componente del sistema. La Figura 10 muestra cada una de las clases a usar en el proyecto y detalla sus funcionalidades necesarias. En la aplicación móvil, se destacan clases como Usuario, BitácorasMuestreos, Identificación, y AvesClasificadas. Cada clase tiene atributos y métodos específicos que permiten realizar operaciones como registrar usuarios, gestionar bitácoras y muestreos, y clasificar aves a partir de imágenes. En la API de correos, se detallan clases como Autenticación y PostageAPI, que manejan el registro de usuarios y el envío de correos de confirmación. Esta vista proporciona una visión detallada de la implementación y estructura del código, facilitando la comprensión del funcionamiento interno del sistema.

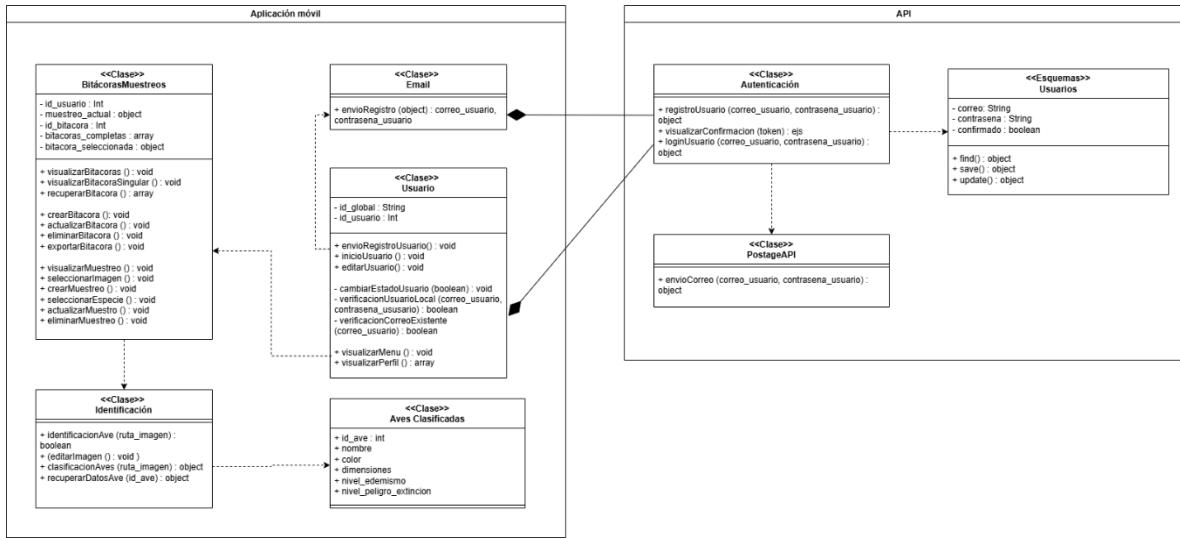


Figura 26 Diagrama de clases

Estos diagramas del arquitectura c4 proporcionan una representación clara y estructurada de la arquitectura del sistema "IdBird", desde la interacción a nivel de usuario hasta la implementación detallada del código. Cada nivel de abstracción ofrece información específica que ayuda a entender cómo se organiza y funciona el sistema en su totalidad.

b. Matriz de trazabilidad

La matriz de trazabilidad del proyecto es una herramienta esencial que permite rastrear y relacionar los diferentes elementos del proyecto, asegurando que todos los requerimientos están cubiertos por los componentes del sistema y que todas las funcionalidades están debidamente probadas. Este documento establece una conexión clara entre los objetivos del proyecto, los requerimientos, los diagramas de diseño, los componentes, los casos de uso y las pruebas realizadas.

El objetivo principal de la matriz de trazabilidad es garantizar que el sistema desarrollado cumple con todos los requerimientos definidos, desde la fase de diseño hasta la implementación y prueba. La Tabla IV presenta la matriz de trazabilidad, que documenta el seguimiento de los requisitos a lo largo del proyecto.

Tabla X Matriz de trazabilidad

Objetivo particular	Requerimiento	Diseño	Componente	Casos de uso	Prueba
Registro y Autenticación de Usuarios	RF-01	P1	Usuario Email Autenticación PostageAPI Usuarios	CU-01	EP-01
		P2			EP-02
		PE3		CU-02	EP-03
Gestión del Perfil de Usuario	RF-01	P6	Usuario	CU-03	EP-04
		P7			
Creación y Gestión de Bitácoras de Campo	RF-02	P3	Bitácoras y Muestreos	CU-04	EP-05
		P4		CU-05	EP-06
		P5		CU-06	EP-07
		P6		CU-11	EP-12
Manejo de Información de Muestreos	RF-03	P7	Bitácoras y Muestreos	CU-07	EP-08
		P8		CU-08	EP-09
		P9		CU-09	EP-10
Operaciones con los Muestreos	RF-03	P10	Bitácoras y Muestreos	CU-07	EP-08
		P11		CU-08	EP-09
		P12		CU-09	EP-10
Identificación y Clasificación de las Aves Muestreadas	RF-04	P13	Identificación	CU-10	EP-11
		P14			
Cumplimiento de Normativas Ambientales	RF-05	PE1	Bitácoras y Muestreos	CU-10	EP-11
		PE2		CU-11	EP-12
Cumplimiento de Normativas Ambientales	RF-02	P4	Bitácoras y Muestreos	CU-04	EP-05
	RF-03	P5		CU-07	EP-08

c. Diseño de la base de datos.

El diseño de la base de datos para el sistema "IdBird" se ha basado en un modelo entidad-relación (ER) y un modelo de base de datos que proporcionan una estructura clara para almacenar y gestionar la información de los usuarios, las bitácoras y los muestreos de aves.

Tablas de la Base de Datos:

El modelo físico de la base de datos se implementa en tablas que reflejan las entidades y relaciones del modelo ER.

- **Usuarios:** Esta tabla almacena la información personal de los usuarios, como nombre, apellido, correo y contraseña. Cada usuario tiene un identificador único (id_Usuarios), que se utiliza para asociar sus bitácoras.

- Bitácoras: La tabla de bitácoras almacena detalles de las observaciones realizadas por los usuarios, incluyendo la fecha, hora, lugar y ubicación de cada bitácora. Cada bitácora está vinculada a un usuario específico a través de Usuarios_id_Usuarios, garantizando que solo los usuarios propietarios puedan acceder y gestionar sus propias bitácoras.
- Muestreos: Esta tabla almacena información detallada de cada muestreo dentro de una bitácora, como el nombre del muestreo, el nombre coloquial del ave observada, coordenadas, color y dimensiones. Cada muestreo está asociado a una bitácora específica mediante Bitacoras_id_Bitacoras, asegurando una relación clara y estructurada entre los datos del muestreo y la bitácora correspondiente.

Como se observa en la Figura 11, se presentan las tablas descritas con sus respectivos atributos, además de sus relaciones entre cada una de ellas.

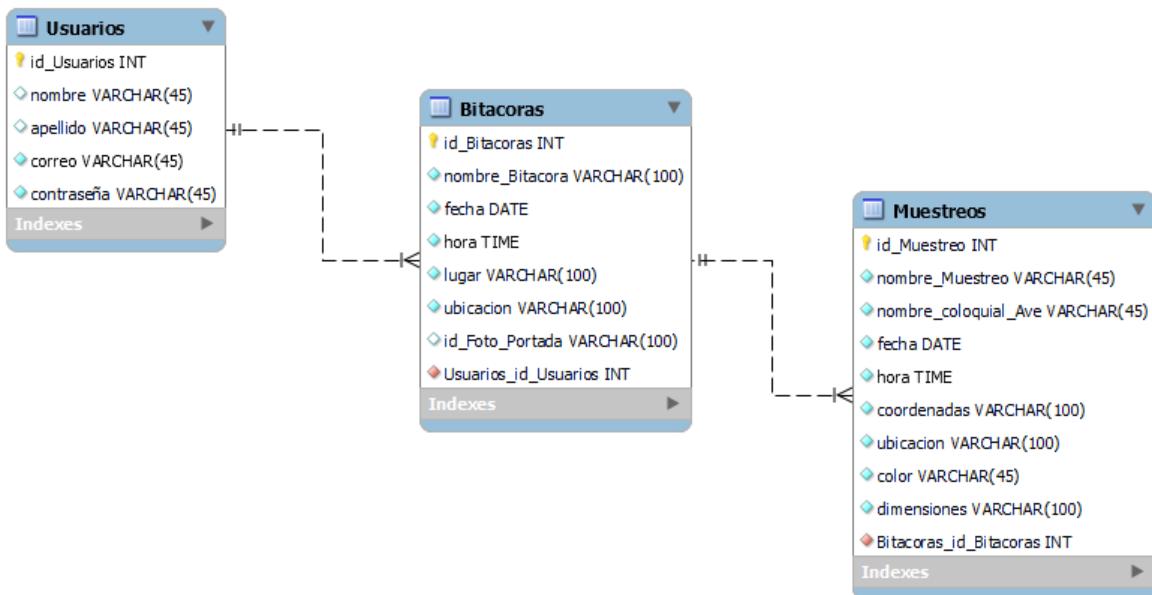


Figura 27 Diagrama de base de datos

Modelo Entidad-Relación (ER):

El modelo ER, ilustrado en la Figura 12, define las principales entidades del sistema: "Usuarios", "Bitácoras" y "Muestreos", y establece las relaciones entre ellas. Cada usuario puede tener múltiples bitácoras, y cada bitácora puede contener múltiples muestreos. Esto asegura que la información esté organizada de manera jerárquica y lógica, facilitando la gestión y consulta de datos relacionados. La relación uno a muchos entre "Usuarios" y "Bitácoras" y entre "Bitácoras" y "Muestreos" permite mantener la integridad referencial de los datos, asegurando que cada bitácora esté asociada a un usuario y cada muestreo a una bitácora específica (imagen entidad-relación).

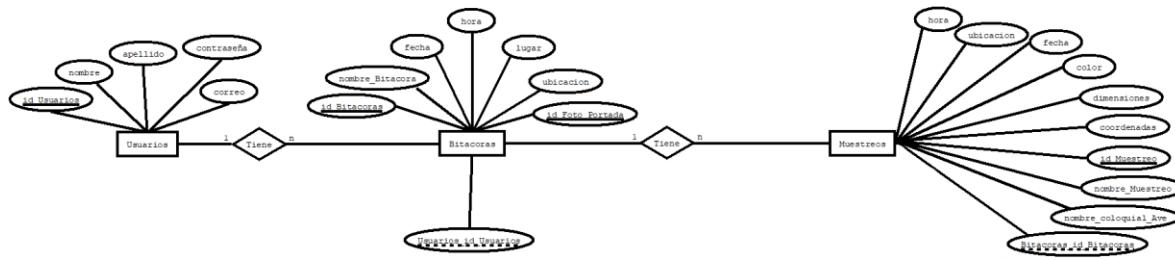


Figura 28 Diagrama de entidad – relación

d. Manejo de archivos.

En el desarrollo del proyecto "IdBird", se ha implementado un sistema de manejo de archivos para gestionar las bitácoras de campo, los muestreos y los archivos PDF generados por la aplicación. A continuación, se describe la estructura del contenido de los archivos y la ruta relativa de su ubicación dentro del sistema.

Estructura del Contenido de los Archivos:

La estructura de manejo de archivos está organizada en directorios y subdirectorios para facilitar el acceso y la gestión de los datos generados y utilizados por los usuarios. La estructura se muestra siguiente Figura:

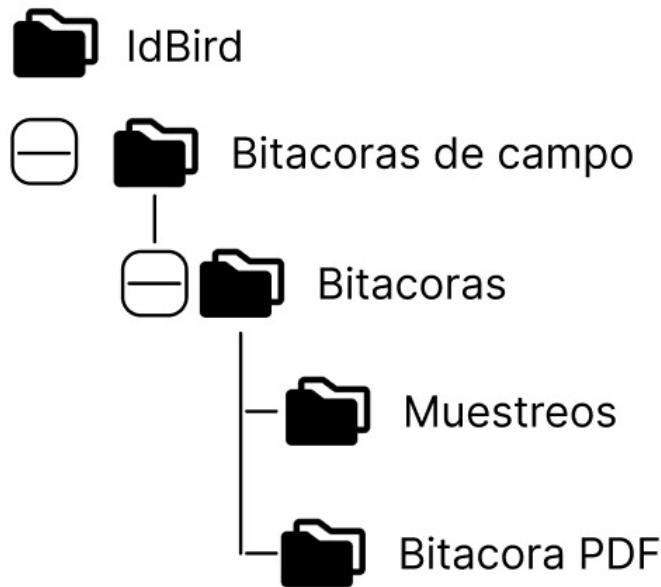


Figura 29 Diagrama de manejo de archivos

- **IdBird:** Directorio raíz de la aplicación que contiene todos los archivos relacionados con las bitácoras de campo.
- **Bitácoras de campo:** Directorio principal que agrupa todas las bitácoras creadas por los usuarios.
- **Bitácoras:** Subdirectorio que almacena todas las bitácoras individuales.
- **Muestreos:** Subdirectorio dentro de cada bitácora que almacena las imágenes subidas o tomadas por el usuario para el muestreo específico.
- **Bitácora PDF:** Subdirectorio dentro de cada bitácora que almacena los archivos PDF generados por la aplicación para cada bitácora.

Ruta Relativa de la Ubicación de los Archivos:

Las rutas relativas dentro del sistema de archivos de la aplicación son las siguientes:

1. Bitácoras de Campo:

- Ruta: ./IdBird/Bitacoras de Campo/Bitacoras/
- Este directorio contiene todas las bitácoras creadas por los usuarios.

2. Muestreos:

- Ruta: ./IdBird/Bitacoras de Campo/Bitacoras_N/Muestreos/

- Este subdirectorio almacena las imágenes relacionadas con los muestreos específicos realizados dentro de cada bitácora.

3. Bitácora PDF:

- Ruta: ./IdBird/Bitacoras de Campo/Bitacoras_N/Bitacora PDF/
- Este subdirectorio almacena los archivos PDF generados por la aplicación para cada bitácora, proporcionando una versión imprimible y portátil de las bitácoras de campo.

La organización de los archivos en esta estructura jerárquica permite un acceso eficiente y una gestión clara de los datos generados por los usuarios, asegurando que la información esté adecuadamente almacenada y fácilmente recuperable cuando sea necesario.

e. Diagramas de actividad.

Los diagramas de actividad proporcionan una visualización detallada de los flujos de trabajo y las interacciones del usuario con el sistema. Estos diagramas son esenciales para entender cómo se llevarán a cabo las principales funcionalidades del sistema, desde el registro de usuarios hasta la exportación de bitácoras en formato PDF, puesto que proporcionan una comprensión clara y estructurada de las interacciones del usuario con el sistema, asegurando que todos los procesos críticos estén bien definidos y gestionados de manera eficiente.

Al igual que todos los diagramas de diseño, se encuentran los 5 diagramas en el documento, se muestra uno de los diagramas para exemplificar.

Exportación de Bitácoras de Campo (RF-05)

El diagrama para la exportación de bitácoras, ilustrado en la Figura 14, muestra cómo los usuarios pueden seleccionar una bitácora y exportarla en formato PDF. Este flujo incluye la obtención de la información de la bitácora, la conversión a un formato de reporte y la generación del archivo PDF. La actividad finaliza con la visualización del reporte generado.

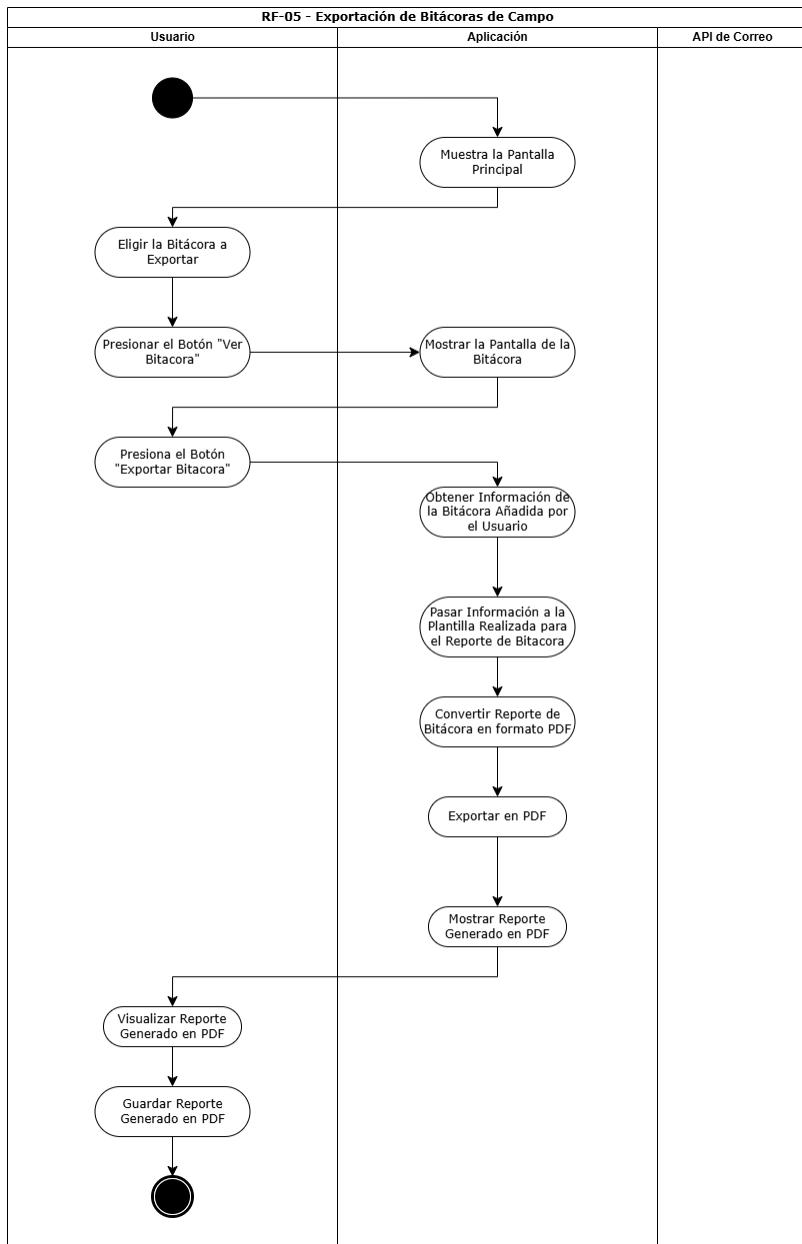


Figura 30 Diagrama de actividad de exportación de bitácoras de campo

f. Diagramas de secuencia.

Los diagramas de secuencia del proyecto proporcionan una representación detallada de las interacciones entre los diferentes componentes del sistema y los usuarios, describiendo el flujo de mensajes y las actividades involucradas en la realización de tareas específicas. Estos diagramas son esenciales para entender la dinámica del sistema y asegurar que todas las funcionalidades se implementen correctamente.

Los diagramas de secuencia cubren varios escenarios clave del sistema "IdBird", incluyendo el registro e inicio de sesión de usuarios, la actualización de perfiles, la creación y gestión de bitácoras y muestreros, la identificación de aves, y la exportación de bitácoras en formato PDF.

Un ejemplo de estos es el ilustrado en la Figura 15, el diagrama de inicio de sesión detalla el proceso completo desde que el usuario ingresa sus credenciales hasta que el sistema verifica la información y concede acceso. Incluyen la interacción con la API de autenticación y la actualización de los estados de usuario en la base de datos.

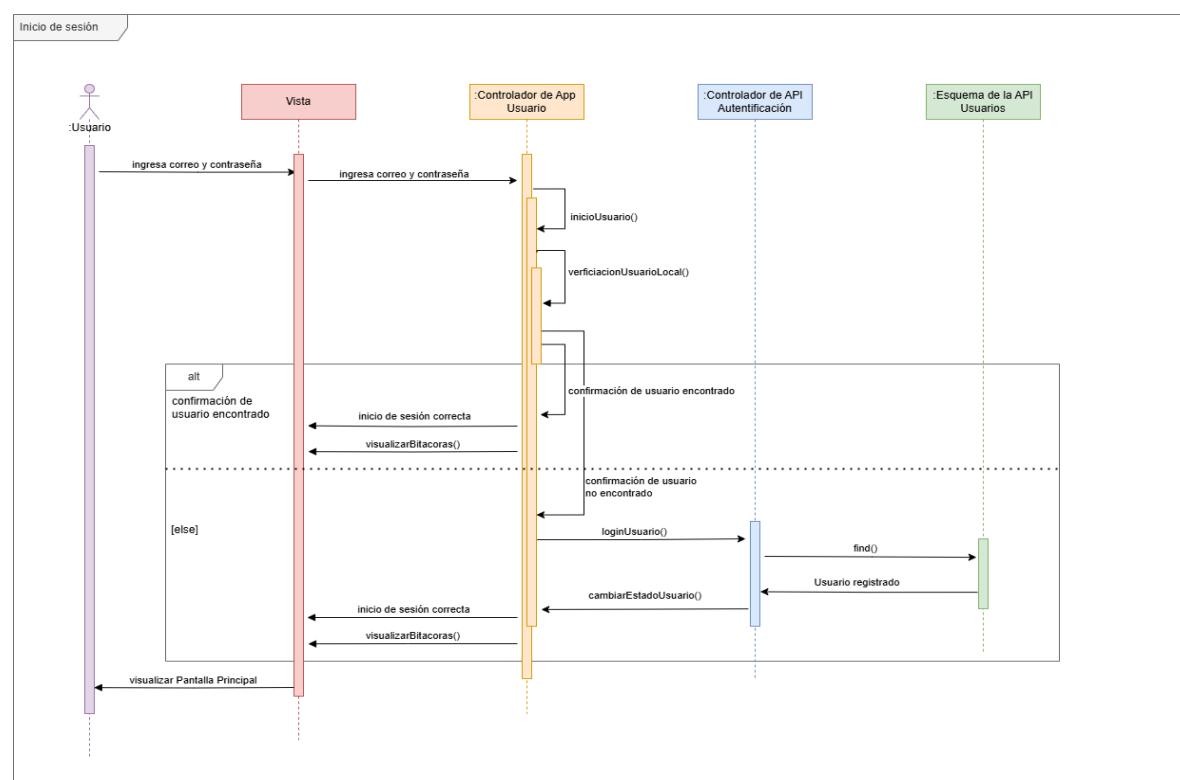


Figura 31 Diagrama de secuencia de inicio de sesión

g. Mockups.

Los mockups presentados para la aplicación IdBird abarcan diversas pantallas clave que forman parte de la funcionalidad integral del sistema. A continuación, se describe brevemente cada una:

1. **Logo de la Aplicación:** Representa la identidad visual de IdBird, diseñado para ser simple y reconocible.
2. **Pantalla de Inicio de Sesión:** Permite a los usuarios ingresar sus credenciales para acceder a la aplicación.
3. **Pantalla de Registro de Usuarios:** Proporciona un formulario para que nuevos usuarios se registren ingresando su nombre, apellidos, correo electrónico y contraseña.
4. **Pantalla Principal (Bitácoras de Campo):** Muestra una lista de bitácoras registradas por el usuario, con opciones para ver detalles, exportar en PDF y crear nuevas bitácoras.
5. **Pantalla de Bitácora:** Detalla los muestreos realizados dentro de una bitácora específica, permitiendo la visualización y creación de nuevos muestreos, así como la exportación de la bitácora en formato PDF.
6. **Pantalla de Muestreo:** Facilita la captura de fotografías para la identificación de aves, mostrando la información relacionada con la especie identificada.
7. **Alertas:** Elementos diseñados para mejorar la experiencia del usuario, notificando cambios o acciones realizadas dentro del sistema.
8. **Pantalla de Confirmación de Usuario Web:** Informa a los usuarios sobre la confirmación de su registro en el sistema después de verificar su correo electrónico.

Para más detalles y especificaciones técnicas de estos y otros mockups, se recomienda consultar el Documento de Diseño, donde se encuentran todos los detalles necesarios para una comprensión completa del diseño implementado a la aplicación. De momento se muestra parte de los mockups realizados en la Figura 16, que contiene el logo principal de la aplicación y las tres primeras pantallas.

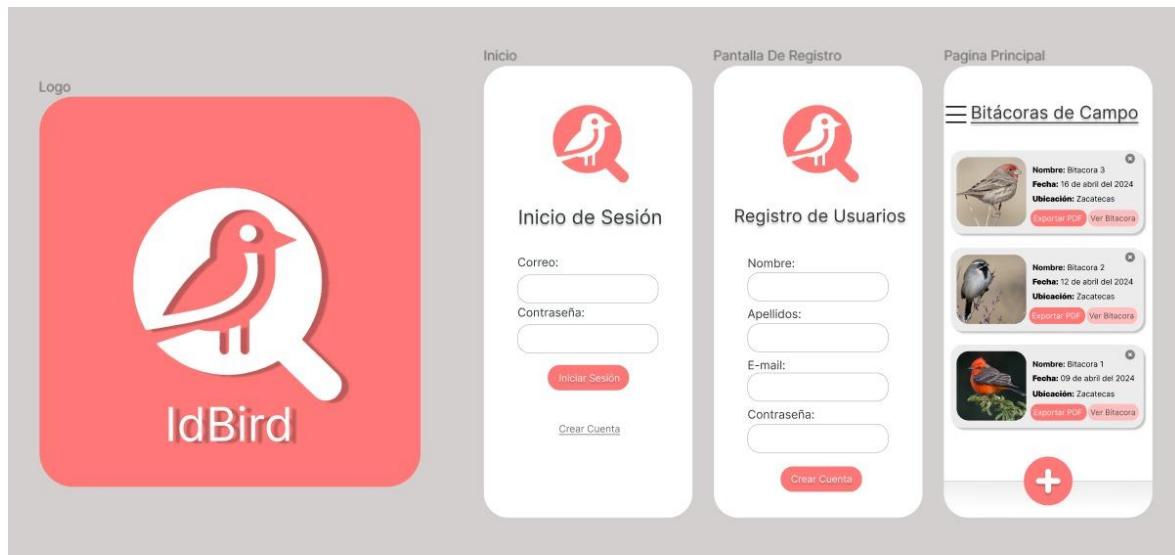


Figura 32 Diseño de prototipos

Todos los diagramas y mockups mostrados, así como todos aquellos creados que forman parte del diseño del sistema, se encuentran en el documento de diseño en el anexo E.

Análisis de resultados.

Durante la fase de Trabajo Terminal 1 (TT-I) del proyecto " Sistema para muestreo de aves en la ciudad de Zacatecas", las etapas iniciales del ciclo de vida del software para el desarrollo de un sistema, agregando la planificación del proyecto para empezar, se realizó el análisis de requerimientos y el diseño del sistema. A lo largo de estas etapas, se siguieron los planes establecidos y se trabaja conforme al ritmo de trabajo y tareas establecidas, sin embargo también se enfrentaron diversos contratiempos que fueron abordados mediante estrategias efectivas de mitigación y resolución de problemas.

Planificación de proyecto

Durante la fase de planificación, se llevó a cabo una investigación exhaustiva que resultó en la decisión de utilizar la metodología del modelo en V. Aunque inicialmente se había considerado un enfoque ágil debido a la modularidad posible del proyecto, se optó finalmente por el modelo en V debido a su similitud con el modelo en cascada, pero con la ventaja añadida de incluir pruebas continuas en cada una de sus etapas, garantizando un proceso integral y detallado y un software de alta calidad.

Por otro lado, la creación del cronograma resultó ser un proceso más prolongado de lo esperado al iniciar, ya que pasó por múltiples revisiones. En cada revisión, se añadieron diversas tareas adicionales para asegurar la finalización completa y de alta calidad de cada documento y requerimiento. Esta atención al detalle y la rigurosidad en la planificación fueron constantes a lo largo del desarrollo del proyecto para garantizar su éxito.

Análisis de Requerimientos

En la fase de análisis de requerimientos, el levantamiento de estos fue ágil, ya que las reuniones con el cliente siempre fueron breves y concisas. Sin embargo, lo que tomó más tiempo fue el análisis detallado de los requerimientos y la determinación de cómo se implementarían en el proyecto, incluyendo la redacción y aseguramiento de la congruencia de cada parte.

Otro aspecto que complicó el avance fue la determinación del funcionamiento planificado para cada una de las partes del proyecto.

Incluyendo identificar las características preferibles que debían tener las imágenes capturadas para que el modelo pudiera clasificar correctamente las especies de aves, realizando pruebas con diferentes dispositivos móviles de gammas medias a altas y comprobando la calidad de imágenes y la necesidad de distancia y zoom.

Asimismo, fue necesario decidir el lenguaje de programación para la aplicación móvil, asegurando su compatibilidad con los modelos de Redes Neuronales Convolucionales (CNN) y su popularidad en el mercado de aplicaciones móviles actuales; finalmente, se eligió Kotlin por su uso y compatibilidad usual en diversos proyectos distintos.

También se seleccionó un modelo CNN que fuera preciso y ligero para funcionar en un dispositivo móvil; optando por el modelo MobileNet de TensorFlow, previamente probado en un trabajo preliminar “Image Recognition System for Bird Sampling in the City of Zacatecas” [28], por su alta precisión y poca necesidad de almacenamiento y procesamiento ideales para dispositivos móviles.

Además, se planteó una forma enviar correos electrónicos a través de una API, según los requerimientos del cliente para la confirmación de registro de usuarios a través de correos, a través de la API de correos PostageApp.

Diseño del Sistema

Durante la fase de diseño del sistema, surgieron múltiples sobreestimaciones y subestimaciones que modificaban ligeramente los tiempos establecidos pero gracias a la organización se mantuvo el ritmo de trabajo estable y los tiempos sin muchos cambios.

La parte de la arquitectura, al estar basada en el arquitectura c4, requirió la elaboración de cuatro diferentes diagramas. Entre estos, los diagramas de componentes y de código fueron los más laboriosos, ya que fue en este momento cuando el equipo de trabajo determinó concretamente cómo funcionarían en conjunto las distintas partes de la aplicación.

Asimismo, los diagramas de actividades fueron más numerosos de lo esperado, lo que llevó tiempo adicional. Sin embargo, los diagramas de secuencia fueron los que más tiempo consumieron debido a la falta de familiaridad del equipo con su elaboración. A pesar de esto, estos diagramas fueron verificados múltiples veces para garantizar que se hicieran correctamente.

Por el contrario, los diagramas restantes, como los de casos de uso, de base de datos, manejo de archivos, entre otros, resultaron ser mucho más sencillos y rápidos de lo esperado, lo que permitió recuperar el tiempo sin mayores retrasos y avanzar conforme al cronograma establecido.

Contratiempos y Resoluciones

Inicialmente, se contemplaron alrededor de 11 riesgos posibles durante el desarrollo del proyecto. Afortunadamente, solo se presentaron 4 de estos riesgos, que al ser manejados con las medidas de mitigación y prevención adecuadas, no tuvieron un impacto grave en el proyecto.

Uno de los principales contratiempos fue la falta de asistencia en ciertos días de trabajo, lo que se mitigó replanificando las actividades y redistribuyendo las responsabilidades entre los miembros presentes. Esta flexibilidad en la gestión del tiempo permitió que el proyecto continuara avanzando sin interrupciones significativas.

Así mismo, el conflicto de puntos de vista dentro del equipo sobre detalles del diseño y técnicas de implementación se resolvió recurriendo a la opinión del director, el asesor y el cliente del proyecto, logrando un consenso que alineó los objetivos del equipo y mejoró la cohesión y cooperación.

Por ultimo, los contratiempos de sobreestimación y subestimación de los tiempos que requeriría cada actividad fue uno de los mas comunes pero sin grandes impactos ya que continuamente se realizaban planeaciones semanales y por etapas de las actividades necesarias y el tiempo disponible para cada una, por lo que se mantuvo el tiempo sin modificaciones que afectaran gravemente al proyecto.

Finalmente, el análisis de resultados muestra que, a pesar de los contratiempos enfrentados, el equipo del proyecto pudo completar las actividades planificadas con una eficiencia notable, reduciendo casi a la mitad el tiempo estimado inicialmente. La clave del éxito radicó en la replanificación dinámica, la colaboración efectiva y la capacitación continua, asegurando que todas las etapas del proyecto se completaran satisfactoriamente.

Conclusiones y Recomendaciones

El proyecto "IdBird" ha completado exitosamente su fase de Trabajo Terminal 1 (TT-I), abarcando la planificación del proyecto, el análisis de requerimientos y el diseño del sistema. Durante este primer periodo se logró una comprensión profunda de las necesidades del usuario y la dirección del proyecto, se definieron los objetivos y se establecieron las bases arquitectónicas para el desarrollo del sistema.

A pesar de enfrentar varios contratiempos, como la falta de asistencia y los conflictos de puntos de vista, el equipo demostró una notable capacidad de adaptación y resolución de problemas, logrando completar las tareas con una eficiencia que redujo casi a la mitad el tiempo estimado inicialmente. Este éxito sienta una adecuada base para la siguiente fase del proyecto, asegurando que el equipo esté bien preparado para abordar los desafíos de la siguiente etapa.

No obstante, se destacaron ciertas recomendaciones para un mejor trabajo futuro.

Mejora en la Gestión del Tiempo: Durante TT-I, la replanificación y redistribución de tareas fueron cruciales para mitigar la falta de asistencia. En TT-II, se recomienda implementar un sistema de seguimiento del tiempo más riguroso y proactivo. Esto incluiría un registro diario de horas y actividades realizadas, de igual forma ajustar las asignaciones de tareas según sea necesario, asegurando que el proyecto mantenga su curso sin interrupciones significativas.

Se propone el siguiente plan de proyecto para el desarrollo del periodo de trabajo terminal 2, ilustrado en la Figura 17, considerando las etapas de implementación, validación, despliegue del sistema y realización del reporte de TT2, además de horas de capacitación necesaria.

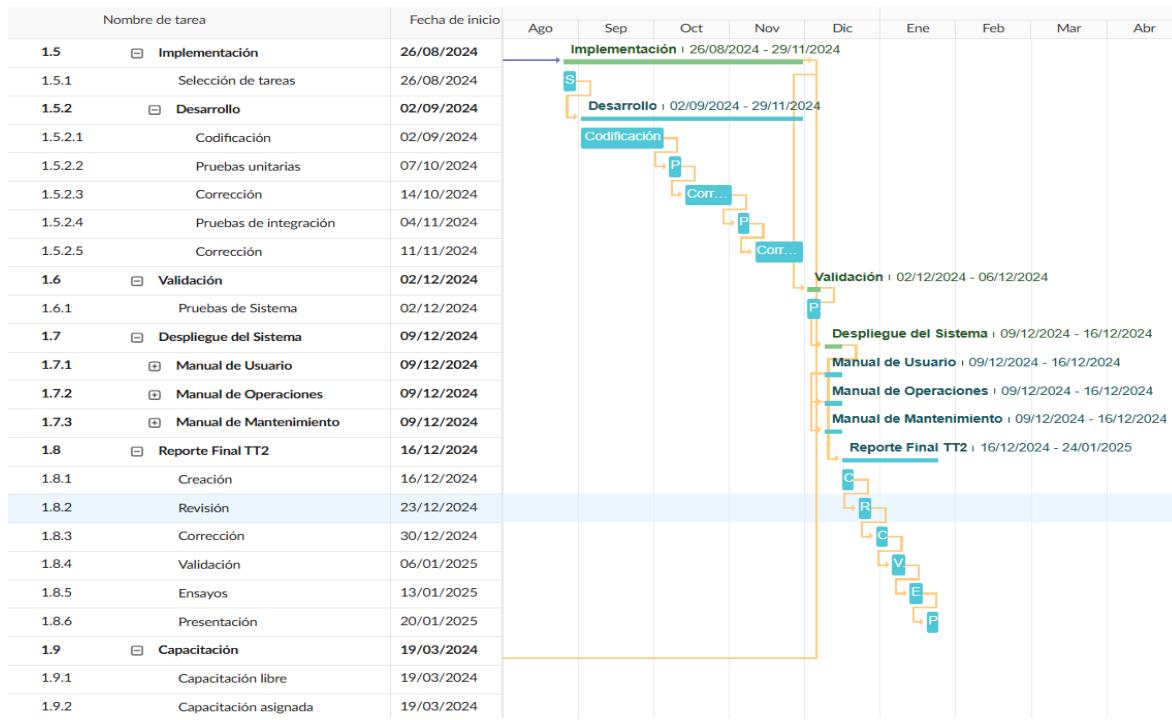


Figura 33 Cronograma propuesto para TT2

Capacitación Continua y Especializada: La capacitación en TT-I se realizó de manera continua y adaptativa, lo cual fue efectivo. En TT-II, se recomienda planificar sesiones de capacitación más estructuradas y específicas, centradas en temas claves que se identifiquen como necesarias para el desarrollo. Asegurando que todos los miembros del equipo tengan las habilidades necesarias para realizar cada una de las partes de implementación de manera eficiente.

Optimización de Recursos y Herramientas: Aprovechando las lecciones aprendidas en TT-I, se recomienda revisar y optimizar las herramientas y recursos utilizados durante el desarrollo. Evaluar la eficiencia de las herramientas de software, como los lenguajes de programación, modelos CNN para identificación y clasificación de imágenes, uso de softwares externos, etc.; metodologías de trabajo y prácticas de gestión de proyectos puede conducir a una mayor productividad y a la reducción de tiempos de desarrollo en TT-II.

Al implementar estas recomendaciones, el equipo de "Sistema para muestreo de aves en la ciudad de Zacatecas" puede mejorar aún más su eficiencia y efectividad, aspirando que la fase de TT-II se desarrolle de manera fluida y exitosa, cumpliendo con los objetivos establecidos y expectativas del proyecto.

Fuentes de consulta

- [1] A. G. Navarro-Sigüenza, M. F. Rebón-Gallardo, A. Gordillo-Martínez, A. T. Peterson, H. Berlanga-García, and L. A. Sánchez-González, "Biodiversidad de aves en México," Rev. Mex. Biodiv., vol. 85, no. Suppl, pp. S476-S495, 2014.
- [2] N. Pérez-Valadez, "Adiciones a la avifauna del estado de Zacatecas," Huitzil, vol. 17, no. 2, pp. 175-183, 2016.
- [3] F. Yang, N. Shen, and F. Xu, "Automatic Bird Species Recognition from Images with Feature Enhancement and Contrastive Learning," Appl. Sci., vol. 14, no. 10, p. 4278, 2024.
- [4] J. L. Navarro-Arteaga, J. L. Alanis-Mendez, M. Hernandez-Sanchez, and F. Limon-Salvador, "Bird inventory from Poza Rica, Veracruz, Mexico," Huitzil, vol. 22, no. 1, e604, 2021.
- [5] M. González-Jaramillo, E. Martínez, L. G. Esparza-Olgún, and J. L. Rangel-Salazar, "Actualización del inventario de la avifauna de la Reserva de la Biosfera de Calakmul, península de Yucatán, México: abundancia, estacionalidad y categoría de conservación," Huitzil, vol. 17, no. 1, 2016.
- [6] J. L. Navarro-Arteaga, J. L. Alanis-Mendez, M. Hernandez-Sanchez, and F. Limon-Salvador, "Bird inventory from Poza Rica, Veracruz, Mexico," Huitzil, vol. 22, no. 1, e604, 202
- [7] "New deep learning AI tool helps ecologists monitor rare birds through their songs," ScienceDaily, November 15, 2023.
- [8] F. Yang, N. Shen, and F. Xu, "Automatic Bird Species Recognition from Images with Feature Enhancement and Contrastive Learning," Appl. Sci., vol. 14, no. 10, p. 4278, 2024.
- [9] "Merlin Bird ID – Free, instant bird identification help and guide for thousands of birds," [Online]. Available: <https://merlin.allaboutbirds.org/>.
- [10] eBird Mobile - eBird," [Online]. Available: <https://ebird.org/about/ebird-mobile/>
- [11] "Merlin Bird ID – Free, instant bird identification help and guide for thousands of birds," [Online]. Available: <https://merlin.allaboutbirds.org/>
- [12] eBird Mobile - eBird," [Online]. Available: <https://ebird.org/about/ebird-mobile/>
- [13] "Birdsnap Dataset | Papers With Code," [Online]. Available: <https://paperswithcode.com/dataset/birdsnap>
- [14] "Geoportal del Sistema Nacional de Información sobre Biodiversidad," [Online]. Available: <http://www.conabio.gob.mx/informacion/gis/>
- [15] Information system | Definition, Examples, & Facts | Britannica", Britannica, 2024

- [16] “Características de las aves - Resumen,” ecologiaverde.com.
<https://www.ecologiaverde.com/caracteristicas-de-las-aves-1630.html>
- [17] “Características de la cámara del móvil: te explicamos todos los términos,” El Blog de BEEP Informática, Apr. 21, 2021. <https://blog.beep.es/camara-del-movil-caracteristicas-significado-terminos/>
- [18] “Procesado de imágenes digitales,” la.mathworks.com. <https://la.mathworks.com/discovery/digital-image-processing.html>
- [19] “Reconocimiento de patrones (Pattern Recognition),” la.mathworks.com.
<https://la.mathworks.com/discovery/pattern-recognition.html>
- [20] C. Kolosky, “What is an Inventory Management System?,” Online Database Software: Use Knack For Custom Online Databases & Systems, Oct. 16, 2023. <https://www.knack.com/blog/what-is-an-inventory-management-system/>
- [21] “Informática Básica: ¿Qué es una aplicación móvil?,” GCFGlobal.org.
[https://edu.gcfglobal.org/es/informatica-basica/que-es-una-aplicacion-movil/1/](https://edu.gcfglobal.org/es/informatica-basica/que-es-una-aplicacion-movil/)
- [22] R. S. Yadav, "Improvement in the V-Model," International Journal of Scientific & Engineering Research, vol. 3, no. 2, pp. 1-8, Feb. 2012. Available: <http://www.ijser.org>.
- [23] R. S. Yadav, "Improvement in the V-Model," International Journal of Scientific & Engineering Research, vol. 3, no. 2, pp. 1-8, Feb. 2012. Available: <http://www.ijser.org>.
- [24] C. C. Calderón Tapia, S. Hernández Ríos, H. M. Rincón Bedoya, L. V. Cruz Martínez, C. C. Murillo, and C. A. Gaitán Sánchez, "Protocolo para el Modelamiento de Arquitectura de Software," Gestión de Tecnologías de la Información, Código: GTI-PROT-03, Versión 1, pp. 1-15, Oct. 2021.
- [25] "MVC - MDN Web Docs Glossary: Definitions of Web-related terms," MDN Web Docs, 2024. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Glossary/MVC>. [Accessed: 04-Jun-2024]
- [26] "MVC - MDN Web Docs Glossary: Definitions of Web-related terms," MDN Web Docs, 2024. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Glossary/MVC>. [Accessed: 04-Jun-2024]
- [27] J. Vivanco, “El modelo C4 de documentación para la Arquitectura de Software,” Medium, Jun. 05, 2019. <https://medium.com/@javivivanco/el-modelo-c4-de-documentaci%C3%B3n-para-la-arquitectura-de-software-424704528390> (accessed Jun. 04, 2024).
- [28] V. S. Sánchez Lee, A. F. Félix Jiménez, I. I. Belmonte and E. F. Parra González, "Image Recognition System for Bird Sampling in the City of Zacatecas," 2023 12th International Conference On Software Process Improvement (CIMPS), Cuernavaca, Morelos, Mexico, 2023, pp. 126-135, doi: 10.1109/CIMPS61323.2023.10528852. keywords: {Image recognition;Biological system modeling;Urban areas;Neural networks;Birds;Software;Software reliability;birds;image processing;Zacatecas City;pattern recognition;convolutional neural networks}

Firmas.

En esta sección se mostrarán los nombres y las firmas de los alumnos responsables del desarrollo del proyecto de Trabajo Terminal.



Axel Frederick Félix Jiménez.



Vania Stephany Sánchez Lee.

Autorización.

Por medio del presente autorizo la impresión y distribución del presente reporte de avances de anteproyecto, toda vez que lo he leído, comprendido en su totalidad, y estoy de acuerdo con su contenido.

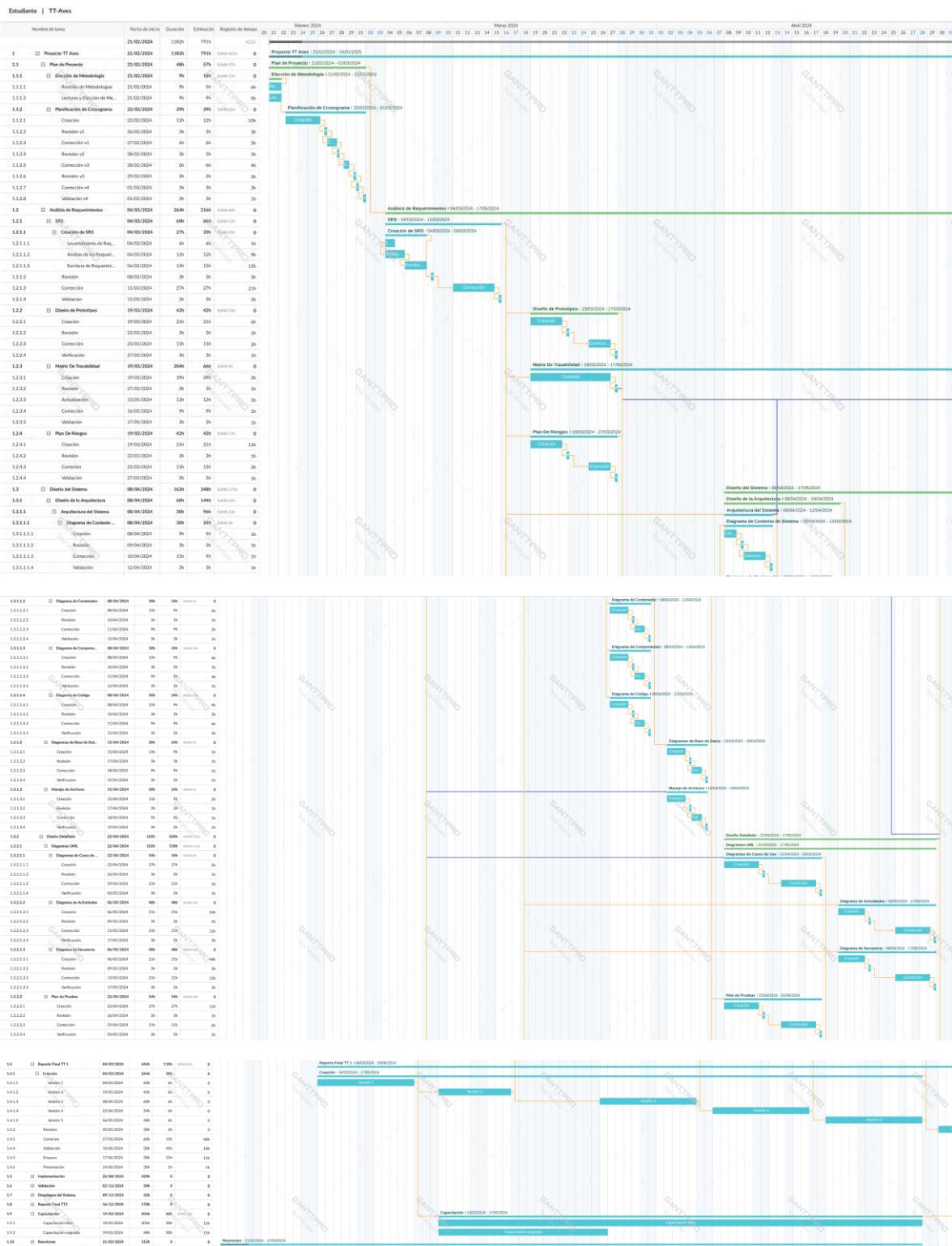
Atentamente;

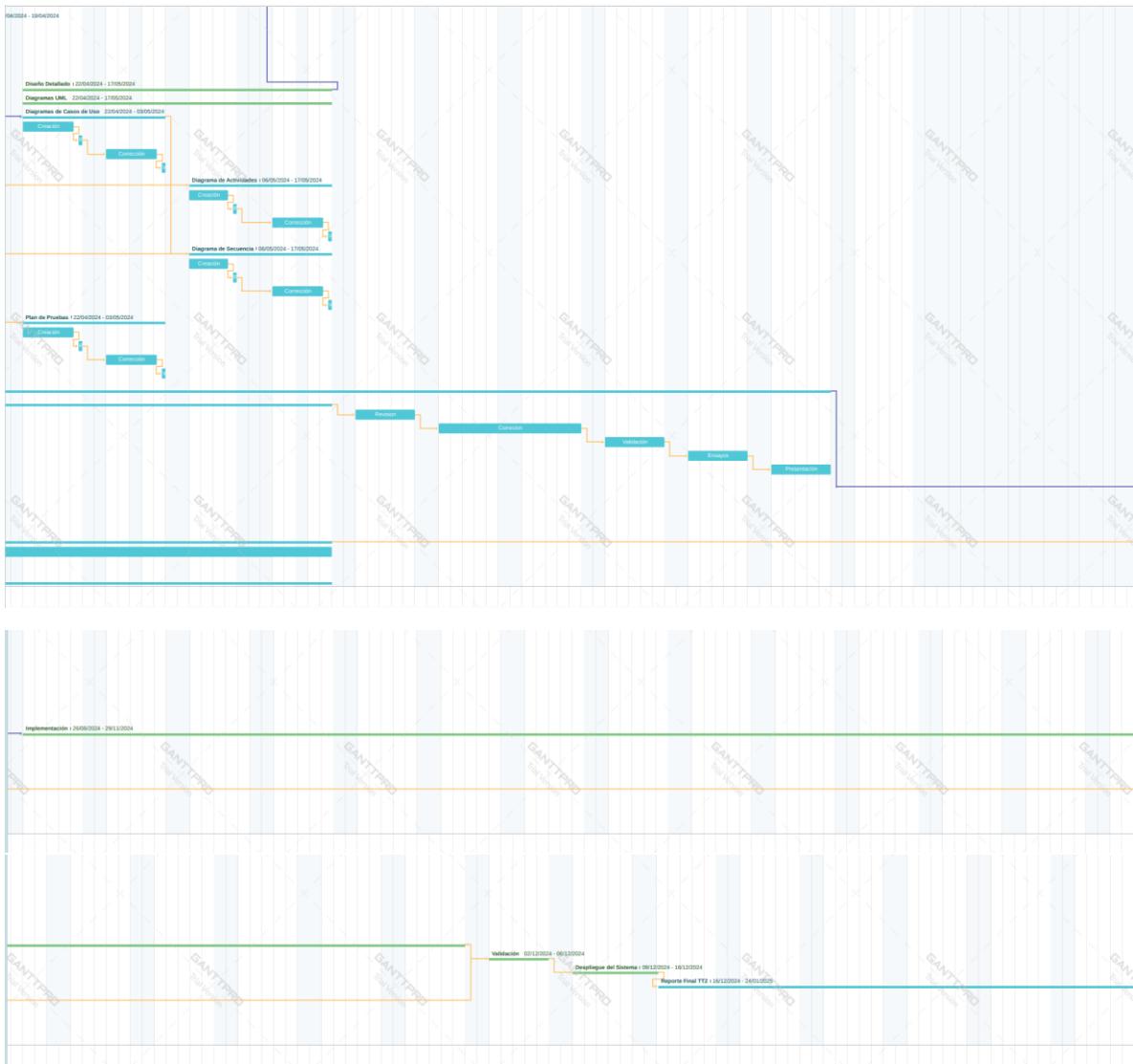
I.S.C Efraín Arredondo Morales

M.I.S Isaul Ibarra Belmonte.

Apéndices

Anexo A Plan de proyecto final completo







**Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria de
Ingeniería campus Zacatecas**

**Área de ubicación para el desarrollo del
trabajo**

Ingeniería en Sistemas Computacionales

Línea de investigación

Desarrollo de aplicaciones móviles

Título del proyecto de Trabajo Terminal

Sistema para muestreo de aves en la ciudad de
Zacatecas

Presenta(n):

Axel Frederick Félix Jiménez.
Vania Stephany Sánchez Lee.

Director:

ISC. Efraín Arredondo Morales.



Asesores:

M.S.I. Isaul Ibarra Belmonte

Zacatecas, Zacatecas a 05 de 03 de 2024

Índices

Índice de contenido

Descripción del proyecto.	1
Objetivo general del proyecto.	1
Objetivos particulares del proyecto.	1
Bibliografía.	2
Firmas.	3
Autorización.	3
Curriculum Vitae del director y los asesores del proyecto de TT.4	

Índice de tablas

Índice de figuras

Índice de gráficas

Descripción del proyecto.

El proyecto que se propone es el desarrollo de un sistema para muestreo de aves en la ciudad de Zacatecas, el cual utilizará una cámara de celular para identificar las características de las aves captadas por el usuario. El objetivo principal de este proyecto es mejorar el sistema de inventario de aves utilizado por los biólogos, proporcionando una herramienta fácil y accesible para la identificación y recopilación de datos de las aves presentes en la ciudad.

El sistema propuesto funcionará mediante la utilización de técnicas de procesamiento de imágenes y aprendizaje automático. Cuando el usuario apunte la cámara del celular a un ave en particular, la aplicación utilizará algoritmos de reconocimiento de patrones para identificar la especie de ave y recopilar datos relevantes sobre su tamaño, forma, coloración, entre otros aspectos. Estos datos serán almacenados en una base de datos y estarán disponibles para su posterior análisis por parte de los biólogos.

La aplicación también contará con una interfaz amigable e intuitiva que permitirá al usuario registrar la ubicación y hora de la observación, así como tomar notas adicionales sobre el comportamiento y hábitat de las aves observadas. Además, se considera la posibilidad de integrar una funcionalidad de georreferenciación para obtener datos precisos sobre la ubicación de las aves y su relación con el entorno.

Se espera que este sistema proporcione una herramienta valiosa para la identificación y seguimiento de las poblaciones de aves en la ciudad de Zacatecas. Además, se espera que la recopilación de datos a través de esta aplicación permita a los biólogos conocer mejor la diversidad de aves en la ciudad y sus patrones de distribución, lo que puede ser de gran ayuda en la toma de decisiones para la conservación de estas especies y sus hábitats.

Objetivo general del proyecto.

Llevar a cabo un muestreo no invasivo de las aves en la zona conurbada para generar un registro de las características de las aves analizadas y llevar dichos datos a la visualización del usuario.

Objetivos particulares del proyecto.

1. Registro y Autenticación de Usuarios:

Implementar un sistema de registro para los usuarios, requiriendo datos esenciales como nombre, apellido, correo electrónico y contraseña. Esto incluirá el envío de un correo electrónico de confirmación para verificar la cuenta del usuario y garantizar la autenticidad de la información proporcionada.

2. Gestión del Perfil de Usuario:

Desarrollar funcionalidades que permitan al usuario gestionar su perfil, incluyendo la capacidad de cambiar su contraseña, borrar su cuenta y consultar su perfil de usuario. Esto asegurará que los usuarios mantengan el control sobre su información personal y la seguridad de su cuenta.

3. Creación y Gestión de Bitácoras de Campo:

Facilitar la creación de bitácoras de campo por parte de los usuarios, permitiendo la inclusión de información detallada como el nombre de la bitácora, fecha, hora, lugar, ubicación, cantidad de muestreos y observaciones.

Proveer herramientas para la gestión de estas bitácoras, habilitando operaciones como crear, visualizar, editar y borrar, para que los usuarios puedan mantener un registro ordenado y actualizado de sus actividades de campo.

4. Manejo de Información de Muestreos:

Permitir el registro detallado de cada muestreo dentro de las bitácoras de campo, incluyendo datos específicos como el nombre del muestreo, el nombre coloquial del ave observada,

fecha, hora, coordenadas, ubicación, color y dimensiones. Esto contribuirá a una documentación exhaustiva de las observaciones realizadas durante las actividades de campo.

5. Operaciones con los Muestreos:

Implementar funcionalidades para la gestión de muestreos, ofreciendo a los usuarios la capacidad de crear, visualizar, editar y borrar registros de muestreo. Esto permitirá una administración eficaz de la información recopilada en las bitácoras de campo, facilitando el acceso y la modificación de los datos conforme sea necesario.

6. Identificación y Clasificación de las Aves Muestreadas:

Desarrollar mecanismos para identificar, registrar, analizar y clasificar las aves muestreadas. Esto incluye la integración de una base de datos que contenga información relevante sobre diversas especies de aves, permitiendo así una clasificación precisa basada en características observables como el color, dimensiones y ubicación.

7. Cumplimiento de Normativas Ambientales:

Incorporar un sistema de consulta y verificación que permita identificar si las aves muestreadas se encuentran en peligro de extinción, conforme a la normativa de la Procuraduría Federal de Protección al Ambiente (PROFEPA). Este sistema facilitará la responsabilidad ambiental y el cumplimiento de las leyes de protección de la fauna, asegurando que las actividades de muestreo se realicen dentro de un marco legal y ético.

8. Localización en Tiempo Real de los Registros:

Implementar una funcionalidad que permita localizar la posición exacta de cada registro de muestreo en tiempo real a través de coordenadas UTP (Universal Transverse Mercator). Esta característica mejorará significativamente la precisión y utilidad de los datos recopilados, permitiendo una referencia espacial clara de cada observación realizada en campo.

Marco metodológico.

Para abordar el desarrollo del Trabajo Terminal, se realizó un análisis exhaustivo entre tres modelos predominantes en el ámbito del desarrollo de software: el modelo en Cascada, el modelo V y el modelo Ágil. Esta revisión crítica no sólo contempló las características inherentes a cada modelo sino también su aplicabilidad al contexto específico del proyecto, con el objetivo de seleccionar la metodología que mejor se alinearán con los requerimientos, la naturaleza y los objetivos del Trabajo Terminal.

Inicialmente, se consideró el Modelo en Cascada, el cual destaca por su estructura secuencial, donde cada fase del desarrollo fluye hacia la siguiente, similar a una cascada[5]. Este modelo se caracteriza por su enfoque riguroso en la definición de requisitos antes de avanzar a las fases de diseño y desarrollo, y por realizar las pruebas una vez completado el desarrollo. Aunque este enfoque facilita la planificación y el control, se identificó que su rigidez y la dificultad para adaptarse a cambios en los requisitos lo hacían menos adecuado para nuestro proyecto, que anticipa la necesidad de flexibilidad y adaptación ante posibles modificaciones en las etapas de desarrollo.

Por otro lado, el Modelo Ágil es conocido por su flexibilidad y su capacidad para adaptarse a cambios en los requisitos a lo largo del desarrollo[6]. Este modelo promueve la entrega continua de software funcional, priorizando la satisfacción del cliente y la capacidad de respuesta ante los cambios. Sin embargo, pese a sus ventajas en términos de adaptabilidad y entrega rápida, requería de un equipo altamente capacitado y una entrega continua de prototipos desde el inicio del proyecto, se consideró menos idóneo para el proyecto a realizar debido a la naturaleza específica de los requisitos y el alcance del trabajo a desarrollar.

Finalmente, el Modelo V emergió como la opción más adecuada para el Trabajo Terminal. Este modelo representa una evolución del modelo en Cascada, incorporando la realización de pruebas en paralelo a cada etapa del desarrollo, lo cual permite una mayor involucración de los probadores desde el inicio del proyecto y facilita la adaptación a cambios en los requisitos[7]. Aunque es más rígido que el modelo Ágil, su estructura garantiza que cada fase

del desarrollo sea rigurosamente validada, lo cual es crucial para nuestro objetivo de desarrollar software de calidad.

El Modelo V se destaca por su enfoque en la calidad y la precisión desde las etapas iniciales del desarrollo, involucrando pruebas en cada fase para asegurar que el producto final cumpla con los estándares de calidad requeridos. Esta metodología es particularmente factible para nuestro proyecto, puesto que permite la detección temprana de errores y la validación continua del producto, aspectos que son fundamentales para alcanzar los objetivos de calidad y funcionalidad establecidos para el Trabajo Terminal.

Este enfoque nos permitirá proceder con el análisis y diseño de manera secuencial, haciendo referencia a la metodología Cascada, pero incorporando un elemento crítico del Modelo V: la ejecución de pruebas en cada etapa del desarrollo.

Desde la etapa de implementación, adoptaremos una práctica diligente de realizar pruebas unitarias y modulares. Esta estrategia es esencial para garantizar que cada componente del software funcione correctamente de manera aislada antes de proceder a su integración. Posteriormente, se llevarán a cabo las pruebas de integración y de sistema, siguiendo los lineamientos establecidos por el Modelo V. Este enfoque nos asegura no solo el cumplimiento de los requisitos específicos del proyecto sino también la calidad y funcionalidad del software desarrollado.

Además, para optimizar la organización, distribución y control de las tareas a lo largo del proyecto, implementaremos un tablero de tareas basado en la metodología Kanban. Esta herramienta será fundamental para visualizar el flujo de trabajo, facilitar la gestión de tareas y promover una mayor eficiencia en el proceso de desarrollo. El uso de este tablero de tareas nos permitirá priorizar tareas y ajustar la carga de trabajo según sea necesario, asegurando así una distribución equitativa de las tareas y una gestión efectiva del tiempo.

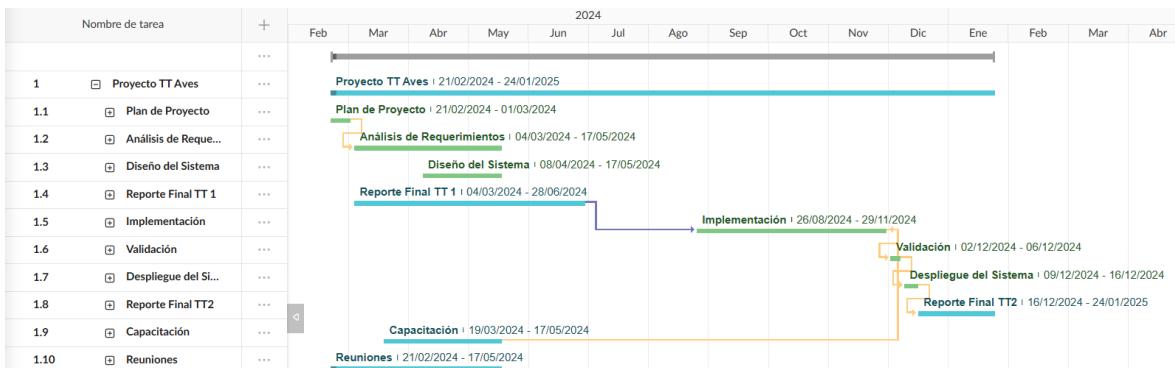
Al combinar la estructura secuencial y la rigurosidad en pruebas del Modelo V con la flexibilidad organizativa un tablero Kanban mejora nuestra capacidad para adaptarnos a cambios en los requisitos en los tiempos esenciales y nos permite mantener un enfoque constante en la calidad y la eficiencia.

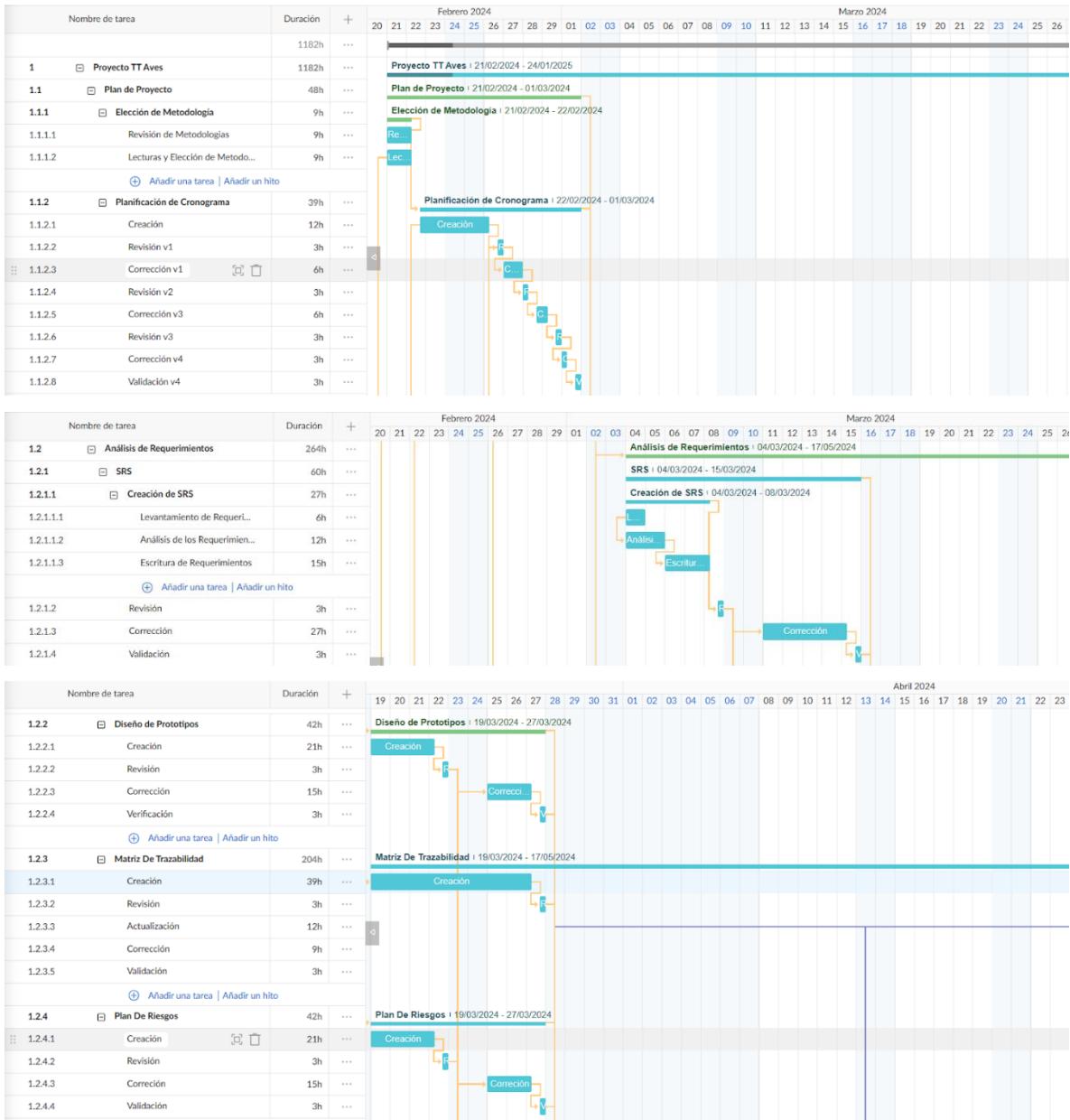
Características	Modelo en Cascada	Modelo V	Modelo Ágil
Descripción	Proceso de diseño secuencial en el que el progreso fluye como una cascada.	Extensión del modelo en cascada con pruebas en cada etapa.	Enfoque en la entrega rápida y continua de software de calidad, adaptándose a los cambios en los requisitos.
Características Principales	<ul style="list-style-type: none"> - Requiere requisitos claros antes de avanzar. - Las pruebas después del desarrollo completo. - Fases secuenciales sin superposiciones. - Cronograma específico para cada fase. - Documentación y pruebas al final de cada fase. 	<ul style="list-style-type: none"> - Involucra probadores desde la fase de requisitos. - Permite cambios en cualquier fase. - Más rígido y menos flexible. 	<ul style="list-style-type: none"> - Equipos adaptables a cambios. - Entrega frecuente de software funcional. - Prioriza satisfacción del cliente y entrega rápida.
Ventajas	<ul style="list-style-type: none"> - Estructura clara y lineal. - Facilita la planificación y control. 	<ul style="list-style-type: none"> - Probadores involucrados desde el principio. - Permite cambios en los requisitos en cualquier momento. 	<ul style="list-style-type: none"> - Flexibilidad para cambios en los requisitos. - Entrega rápida de productos en proyectos pequeños.
Desventajas	<ul style="list-style-type: none"> - Poca flexibilidad para cambios. - Los defectos suelen descubrirse tarde. 	<ul style="list-style-type: none"> - Requiere actualizaciones extensas de documentación con cambios. - No adecuado para proyectos a corto plazo. 	<ul style="list-style-type: none"> - Requiere equipo altamente capacitado y comprometido. - No ideal para proyectos con requisitos estables y claros.

Etapas de Desarrollo	1. Requisitos 2. Diseño 3. Implementación 4. Verificación 5. Mantenimiento	Desarrollo: 1. Requisitos 2. Diseño del sistema 3. Diseño arquitectónico 4. Implementación Pruebas: 1. Pruebas unitarias 2. Pruebas de integración 3. Pruebas de sistema 4. Pruebas de aceptación	1. Planificación 2. Diseño 3. Desarrollo 4. Pruebas 5. Revisión y Retrospectiva 6. Lanzamiento
----------------------	--	--	---

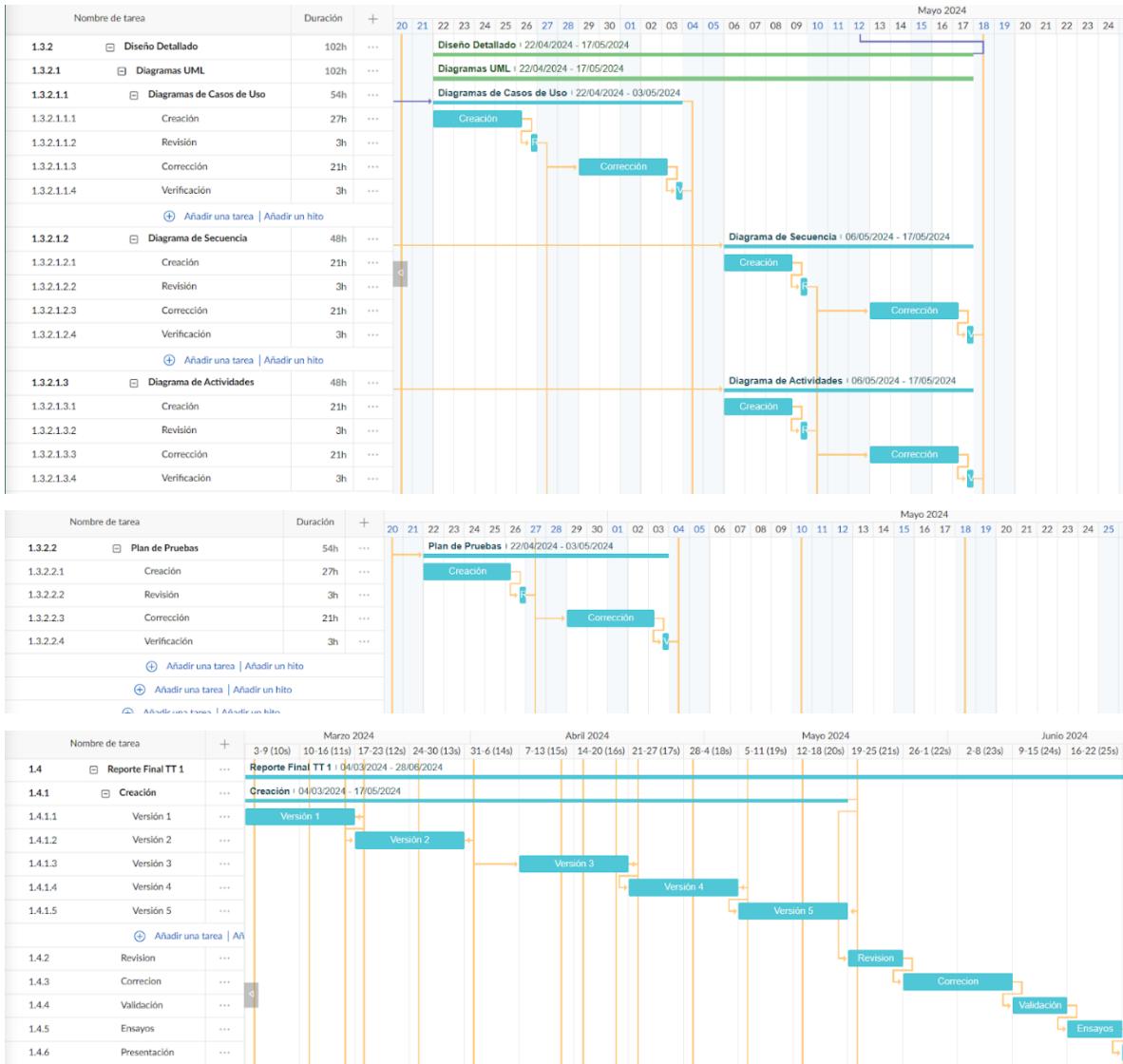
En conclusión, tras una cuidadosa deliberación y análisis de las ventajas y desventajas de cada modelo, el Modelo V fue seleccionado como la metodología óptima para el desarrollo del Trabajo Terminal. Esta decisión se fundamenta en su compatibilidad con los requisitos y objetivos del proyecto, así como en su capacidad para integrar la calidad en cada etapa del desarrollo, asegurando un producto final que no solo cumple con los objetivos garantizando la calidad del software.

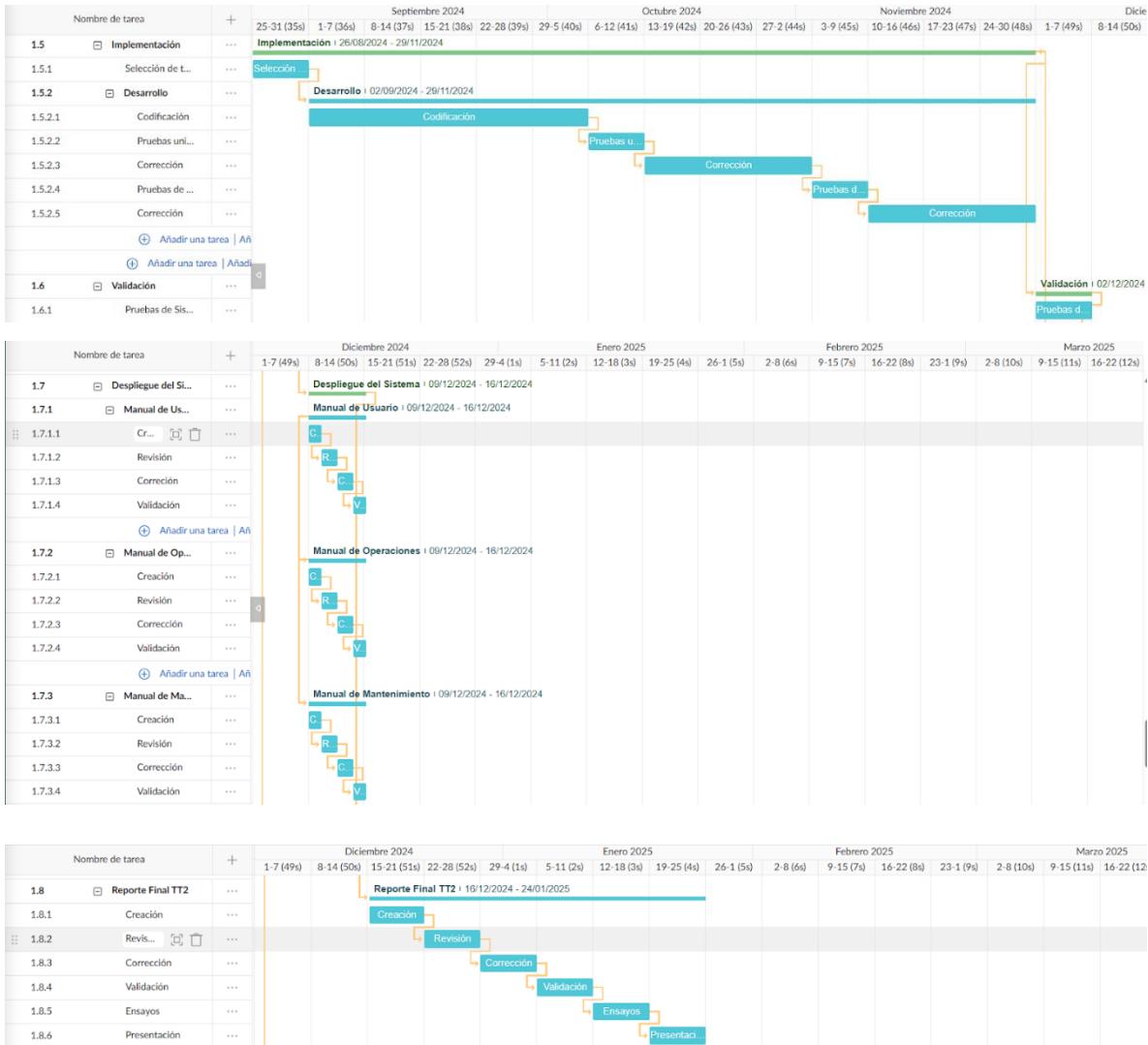
Cronograma de actividades.

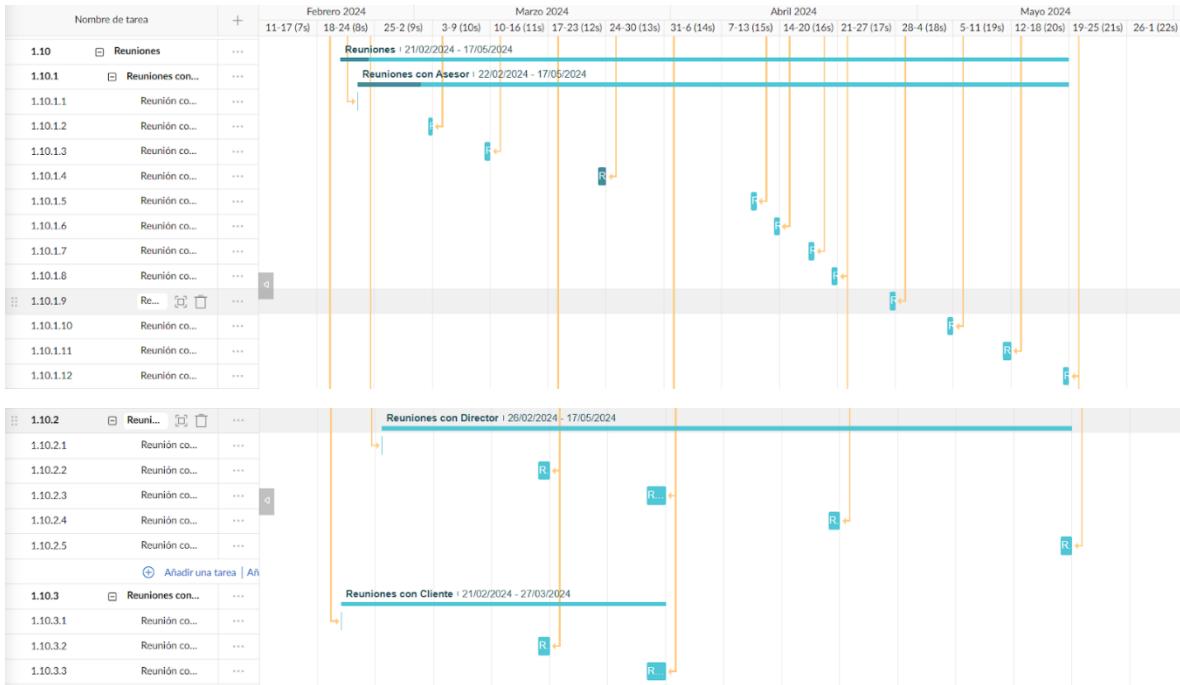




Nombre de tarea	Duración	+	Abril 2024
1.3 □ Diseño del Sistema	162h	...	Diseño del Sistema 08/04/2024 - 17/05/2024
1.3.1 □ Diseño de la Arquitectura	60h	...	Diseño de la Arquitectura 08/04/2024 - 19/04/2024
1.3.1.1 □ Arquitectura del Sistema	30h	...	Arquitectura del Sistema 09/04/2024 - 12/04/2024
1.3.1.1.1 □ Diagrama de Cont...	30h	...	Diagrama de Contexto de Sistema 08/04/2024 - 12/04/2024
1.3.1.1.1.1 Creación	9h	...	Creación
1.3.1.1.1.2 Revisión	3h	...	Revisión
1.3.1.1.1.3 Corrección	15h	...	Corrección
1.3.1.1.1.4 Validación	3h	...	Validación
1.3.1.1.2 □ Diagrama de Contenedor	30h	...	Diagrama de Contenedor 08/04/2024 - 12/04/2024
1.3.1.1.2.1 Creación	15h	...	Creación
1.3.1.1.2.2 Revisión	3h	...	Revisión
1.3.1.1.2.3 Corrección	9h	...	Corrección
1.3.1.1.2.4 Validación	3h	...	Validación
1.3.1.1.3 □ Diagrama de Componentes	30h	...	Diagrama de Componentes 08/04/2024 - 12/04/2024
1.3.1.1.3.1 Creación	15h	...	Creación
1.3.1.1.3.2 Revisión	3h	...	Revisión
1.3.1.1.3.3 Corrección	9h	...	Corrección
1.3.1.1.3.4 Validación	3h	...	Validación
1.3.1.1.4 □ Diagrama de Código	30h	...	Diagrama de Código 08/04/2024 - 12/04/2024
1.3.1.1.4.1 Creación	15h	...	Creación
1.3.1.1.4.2 Revisión	3h	...	Revisión
1.3.1.1.4.3 Corrección	9h	...	Corrección
1.3.1.1.4.4 Verificación	3h	...	Verificación
1.3.1.2 □ Diagramas de Base de Datos	30h	...	Diagramas de Base de Datos 15/04/2024 - 19/04/2024
1.3.1.2.1 Creación	15h	...	Creación
1.3.1.2.2 Revisión	3h	...	Revisión
1.3.1.2.3 Corrección	9h	...	Corrección
1.3.1.2.4 Verificación	3h	...	Verificación
1.3.1.3 □ Manejo de Archivos	30h	...	Manejo de Archivos 15/04/2024 - 19/04/2024
1.3.1.3.1 Creación	15h	...	Creación
1.3.1.3.2 Revisión	3h	...	Revisión
1.3.1.3.3 Corrección	9h	...	Corrección
1.3.1.3.4 Verificación	3h	...	Verificación







Bibliografía.

1. O. L. Lodoño Palacio, L. F. Maldonado Granados y L. C. Calderón Villafáñez, «Guía para constituir Estados del Arte.,» International Corporation of networks of Knowledge, p. 39, 2014.
2. R. Hernández Sampieri, C. Fernández-Collado y P. Baptista Lucio, Metodología de la Investigación, Ciudad de México: Mc. Graw Hill, 2006.
3. S. Balaji and M. Sundararajan, “International Journal of Information Technology and Business Management WATEERFALLVs V-MODEL Vs AGILE: a COMPARATIVE STUDY ON SDLC,” International Journal of Information Technology and Business Management, vol. 2, no. 1, Jun. 2012, Available: <https://mediaweb.saintleo.edu/Courses/COM430/M2Readings/WATEERFALLVs%20Vs%20AGILE%20A%20COMPARATIVE%20STUDY%20ON%20SDLC.pdf>

4. R. Shanker Yadav, "Improvement in the V-Model," International Journal of Scientific & Engineering Research, vol. 3, no. 2, Feb. 2012, Available: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=626c60ffcaf6c0d4b4a18a6f84c1651dc3bff31c>
5. International Journal of Information Technology and Business Management. (2012). Waterfall Vs V-Model Vs Agile: A Comparative Study on SDLC. JITBM & ARF. <https://www.jitbm.com>
6. International Journal of Information Technology and Business Management. (2012). Waterfall Vs V-Model Vs Agile: A Comparative Study on SDLC. JITBM & ARF. <https://www.jitbm.com>
7. International Journal of Information Technology and Business Management. (2012). Waterfall Vs V-Model Vs Agile: A Comparative Study on SDLC. JITBM & ARF. <https://www.jitbm.com>

Firmas.

En esta sección se mostrarán los nombres y las firmas de los alumnos responsables del desarrollo del proyecto de Trabajo Terminal.

Axel Frederick Félix Jiménez. Vania Stephany Sánchez Lee.

Autorización.

Por medio del presente autorizo la impresión y distribución del marco metodológico y cronograma de actividades, toda vez que lo he leído, comprendido en su totalidad, y estar de acuerdo con su desarrollo.

Atentamente;

I.S.C. Efrain Arredondo Morales

M.I.S. Isaul Belmonte Ibarra

Contenido

<u>1. Introducción.</u>	<u>2</u>
<u>1.1 Propósito.</u>	<u>2</u>
<u>1.2 Alcance.</u>	<u>3</u>
<u>1.3 Definiciones, acrónimos y abreviaturas.</u>	<u>4</u>
<u>1.4 Referencias.</u>	<u>6</u>
<u>1.5 Vista general.</u>	<u>6</u>
<u>2. Descripción General.</u>	<u>7</u>
<u>2.1. Perspectiva del producto.</u>	<u>7</u>
<u>2.2. Funcionalidad del producto.</u>	<u>8</u>
<u>2.3. Características del usuario.</u>	<u>9</u>
<u>2.4. Restricciones generales.</u>	<u>9</u>
<u>2.5. Presunciones y dependencias.</u>	<u>11</u>
<u>3. Especificación de requerimientos.</u>	<u>12</u>
<u>3.1. Requerimientos Funcionales.</u>	<u>12</u>
<u>4.1. Requerimientos de desempeño.</u>	<u>20</u>
<u>4.2. Requerimientos de la base de datos lógica.</u>	<u>21</u>
<u>4.3. Restricciones de diseño.</u>	<u>21</u>
<u>a. Cumplimiento de estándares.</u>	<u>21</u>
<u>4.4. Atributos</u>	<u>22</u>
<u>a. Portabilidad.</u>	<u>22</u>
<u>4.5. Requisitos de Interfaces externas.</u>	<u>22</u>
<u>5. Anexos.</u>	<u>23</u>

Introducción.

En el presente documento se explicarán y analizarán los requisitos del proyecto “Sistema para muestreo de aves en la ciudad de Zacatecas”, desarrollado por los alumnos Axel Frederick Félix Jiménez y Vania Stephany Sánchez Lee, ambos pertenecientes del Instituto Politécnico Nacional.

El propósito de este SRS es proporcionar una guía clara para todos los participantes del proyecto, asegurando que los desarrolladores, diseñadores y stakeholders compartan una visión unificada y objetivos claros hacia la implementación exitosa del sistema.

Con un enfoque en mejorar la identificación y registro de aves a través de tecnología avanzada de procesamiento de imágenes en dispositivos móviles, esta aplicación busca ser

una herramienta fundamental para biólogos y entusiastas de aves, contribuyendo a la conservación y educación ambiental en Zacatecas.

1.1 Propósito.

Este documento tiene como propósito dar a conocer el funcionamiento general del Proyecto Sistema para muestreo de aves en la ciudad de Zacatecas que está dirigido al cliente MHPE-TE Hector Alejandro Acuña Cid. Este documento detalla las funcionalidades clave, restricciones y especificaciones técnicas necesarias.

1.2 Alcance.

La aplicación, llamada “IdBird”, permitirá a los usuarios utilizar la cámara de su celular para identificar aves en su entorno natural o campo abierto en un escenario ideal por definir acorde a las limitaciones del dispositivo listados en la Tabla 1. Utilizará técnicas de procesamiento de imágenes para recopilar datos relevantes sobre las aves observadas, incluyendo especie, tamaño, forma y coloración. Estos datos serán almacenados en el dispositivo para su posterior análisis. Así mismo la aplicación ofrecerá a los usuarios la capacidad de registrar detalles específicos sobre sus observaciones de aves, como la ubicación, hora, comportamiento y hábitat. Se explorará la integración de funcionalidades de georreferenciación para mejorar la precisión de los datos de ubicación.

Para acceder a las funcionalidades de la aplicación, los usuarios deberán de registrarse proporcionando información personal básica y verificar su cuenta a través de un correo electrónico de confirmación por medio de una conexión a internet. La gestión de la cuenta de usuario permitirá cambiar la contraseña, borrar la cuenta y consultar el perfil del dispositivo.

La aplicación también incluirá la funcionalidad de bitácoras de campo, permitiendo a los usuarios crear registros detallados de sus actividades de observación de aves. Cada bitácora incluirá información como el nombre de la bitácora, fecha, hora, lugar, ubicación, cantidad de muestreos y observaciones generales. Los usuarios tendrán la capacidad de crear, visualizar, editar y borrar bitácoras.

Para cada muestra individual dentro de una bitácora, se mostrarán datos como el nombre del muestreo, el nombre coloquial del ave observada, fecha, hora, ubicación, color y dimensiones del ave. Estas funcionalidades permiten a los usuarios y biólogos recopilar y analizar datos valiosos sobre las aves, contribuyendo así a la investigación y conservación de estas especies.

1.3 Definiciones, acrónimos y abreviaturas.

Sistema para muestreo de aves en la ciudad de Zacatecas: El proyecto de software enfocado en utilizar la cámara de un celular para identificar características específicas de aves con el propósito de mejorar el sistema de inventario de aves para biólogos y entusiastas en la ciudad de Zacatecas.

- Cámara de celular: Dispositivo integrado en un teléfono móvil diseñado para capturar imágenes o videos.
- Inventario de aves: Registro completo de las especies de aves presentes en una determinada área geográfica.
- Biología: Ciencia que estudia los seres vivos y sus procesos vitales.
- Eficiencia: Capacidad del sistema para cumplir con sus objetivos de manera efectiva y con recursos mínimos.
- Accesibilidad: Característica del sistema que permite su fácil utilización y disponibilidad para un amplio grupo de usuarios.
- Conservación de la biodiversidad: Prácticas y políticas destinadas a proteger y preservar la variedad de vida en la Tierra.
- Aves: Clase de animales vertebrados ovíparos, de respiración aérea, con plumas y cuerpos adaptados para el vuelo.
- Región: Área geográfica específica, en este contexto, la ciudad de Zacatecas.
- Interpretación apropiada: Comprender el contenido y significado del documento de manera correcta y efectiva.
- Acrónimo: Sigla formada por las letras iniciales de varias palabras, que se utiliza como una abreviatura.
- Abreviatura: Forma reducida de una palabra o frase que se utiliza para ahorrar espacio o tiempo al escribir o hablar.
- Algoritmo: Conjunto finito de instrucciones bien definidas y ordenadas que permiten realizar una actividad o resolver un problema.
- Metodología: Conjunto de métodos, técnicas y procedimientos utilizados para realizar una investigación o llevar a cabo un proyecto de manera sistemática y eficiente.
- Proyecto de software: Esfuerzo planificado y organizado para desarrollar un sistema o aplicación de software, que incluye actividades como análisis, diseño, implementación y pruebas.
- Ciclo de vida del software: Serie de etapas por las que pasa un sistema de software desde su concepción hasta su retirada del uso, que incluye fases como planificación, desarrollo, pruebas, implementación y mantenimiento.
- Prototipo de software: Versión preliminar de un sistema de software que se desarrolla para probar y validar conceptos, funcionalidades o requerimientos antes de la implementación completa del sistema.

1.4 Referencias.

Sehgal, A., & Kehtarnavaz, N. (2019). Guidelines and Benchmarks for Deployment of Deep Learning Models on Smartphones as Real-Time Apps. *Machine Learning And Knowledge Extraction*, 1(1), 450-465. <https://doi.org/10.3390/make1010027>

International Journal of Scientific & Engineering Research. (2012). Improvement in the V-Model. <http://www.ijser.org>. [Yadav, R. S.]

Balaji, S., & Murugaiyan, M. S. (2012). WATEERFALLVs V-MODEL Vs AGILE: A COMPARATIVE STUDY ON SDLC. *International Journal of Information Technology and Business Management*, 29(1), 1-5.

1.5 Vista general.

El documento está dividido en 4 secciones:

- La sección 1 se enfoca en la explicación, objetivos, metas y descripción del documento.
- La sección 2 está orientada, como su nombre lo indica, a la descripción general del sistema, donde la información está orientada al cliente/usuario potencial.
- La sección 3 trata sobre los requisitos específicos. Se emplean términos técnicos orientados principalmente a los desarrolladores y programadores.

2. Descripción General.

2.1 Perspectiva del producto.

La aplicación a desarrollar al tratarse de un sistema de identificación de aves nativas de Zacatecas utilizando una cámara en un teléfono móvil es un producto único e independiente a otros realizados previamente.

A pesar de su originalidad, existen aplicaciones como BirdNet y INaturalist, por mencionar algunas semejantes, que también se enfocan en la identificación y registro de aves. BirdNet es conocida por su capacidad para identificar aves a través de sonidos y realizar registros de avistamientos, mientras que INaturalist destaca por su funcionalidad de registro de avistamientos de numerosas especies.

El sistema propuesto se enfoca en el análisis de imágenes para la captura y clasificación de 6 especies de aves que componen la biodiversidad zacatecana, las especies elegidas son especies endémicas de México y que habitan en la ciudad de Zacatecas las cuales son el carpintero de Arizona (*Dryobates arizonae*), el papamoscas pinero (*Empidonax affinis*), el carbonero mexicano (*Poecile sclateri*), el zacatonero serrano (*Oriturus superciliosus*), el chipe gorra canela común (*Basileuterus rufifrons*) y el papamoscas cardenalito (*Pyrocephalus rubinus*).

Las aves mencionadas fueron seleccionadas por sus características frente a otras especies que las distingue visualmente de mejor manera de las diferentes especies facilitando.

La limitación del número elegido de especies de aves mejora la precisión de la identificación de aves, además de facilitar el registro y análisis de observaciones detalladas, permitiendo a los usuarios crear bitácoras de campo digitales para organizar sus descubrimientos.

Además de su utilidad práctica, la aplicación actúa como una herramienta para biólogos, conservacionistas y aficionados. Su diseño intuitivo asegura una experiencia de usuario amigable, lo que la hace accesible para todos, independientemente de su nivel de conocimiento previo sobre aves o habilidades tecnológicas.

Diferenciándose por su enfoque localizado en la fauna de Zacatecas, integración de georreferenciación para mejorar la precisión de los datos y el uso de tecnología de procesamiento de imágenes. La aplicación proporciona una solución efectiva para la identificación y registro de aves además de contribuir a la investigación, educación y conservación de estas especies.

Se anexa un diagrama que muestra los componentes principales del sistema. Anexo A de Anexos.

2.2 Funcionalidad del producto.

La aplicación en el dispositivo debe ser capaz de realizar las siguientes funciones de manera local

- a. Registro de usuarios e inicio de sesión.
- b. Actualización y eliminación de usuarios.
- c. Creación de bitácoras de campo.
- d. Creación de registros de muestreo.
- e. Identificación de especie de ave a través de una imagen.
- f. Captura de tiempo y ubicación del muestreo.

2.3 Características del usuario.

El sistema se ha desarrollado con el objetivo de satisfacer las necesidades de dos grupos principales de usuarios: biólogos y entusiastas de la observación de aves. Ambos grupos poseen conocimientos sobre aves y sus respectivos hábitats, buscando en nuestra herramienta un aliado esencial para facilitar las actividades de muestreo y el registro preciso de avistamientos.

2.4 Restricciones generales.

El desarrollo y operación de la aplicación "IdBird" están sujetos a una serie de restricciones generales que impactan directamente tanto en la fase de diseño como en la implementación y uso final del producto. Estas restricciones incluyen:

- Se limitará el reconocimiento a 6 especies de aves que se encuentren dentro del alcance de detección.
- El funcionamiento y almacenamiento de la aplicación será exclusivamente local, con la excepción de la verificación de correos.
- El procesamiento de imágenes se realizará de manera local en el dispositivo.
- Se emplea un modelo de reconocimiento de aves que sea compatible tanto con el lenguaje de programación de la aplicación Kotlin, la biblioteca Tensorflow Lite para

el desarrollo del modelo de reconocimiento de imágenes, así como con dispositivos de gama media a alta.

- La selección de especies se centrará en las aves endémicas de México y que habitan en la ciudad de Zacatecas, descritas previamente.
- La aplicación está diseñada específicamente para usuarios con formación en biología que se dedican a la investigación aviar.
- Se generarán bitácoras de los muestreos de aves realizados.

- **Limitaciones de hardware:** La aplicación debe estar diseñada para operar eficientemente dentro de las limitaciones de procesamiento y almacenamiento de dispositivos móviles de gama media alta o mayores.

Tabla 1 Tabla de las características de un teléfono móvil de gama media-alta

Aspectos Diferenciadores de un Teléfono Móvil de Gama Media-Alta	
Procesador	Snapdragon serie 700, Mediatek Dimensity serie 800/900, o equivalentes. Octa-core, hasta 2.4 GHz, tecnología de 7nm o 6 nm para eficiencia y potencia.
Memoria y Almacenamiento:	4-8 GB Ram
Almacenamiento Interno:	128-256 Gb
Cámaras	Características: Gran angular de 13 MP (f/1.8) con magnificación máxima de 0.21x, Ultra gran angular de 2 MP (f/2.2) con magnificación máxima de 0.21x, Lente de profundidad de 2 MP (f/2.4) Funciones: Enfoque automático, flash LED, geoetiquetado, HDR.

- **Criticidad de la aplicación:** Dada la importancia del sistema para la investigación y conservación de aves, es fundamental garantizar su estabilidad y confiabilidad determinando una precisión mínima requerida para la clasificación correcta de las especies. Las imágenes serán preprocesadas y luego analizadas por un modelo de redes neuronales convolucionales (CNN) para clasificar e identificar las especies de aves correspondientes.
- **Consideración de seguridad:** Se busca proteger la privacidad y seguridad de los datos de nuestros usuarios. La información personal, incluidos nombre, correo electrónico y datos de ubicación, se recopila exclusivamente para el funcionamiento del sistema y no se utilizará para otros fines sin el consentimiento explícito del usuario.

2.5 Presunciones y dependencias.

La app se crea exclusivamente para dispositivos con sistema Android versión mínima 10, por lo que los usuarios necesitan dispositivos compatibles para acceder a todas las funciones de la app.

A pesar de que la app está diseñada para operar mayormente de forma independiente sin necesitar una conexión constante a internet, se espera que los usuarios puedan conectarse a internet para completar la creación de su cuenta y la verificación por correo electrónico.

La app depende de la precisión del GPS del dispositivo para ubicar con exactitud las aves observadas.

Se considera que los dispositivos de los usuarios disponen de cámaras de calidad suficiente (ver Tabla 1) para tomar fotos claras y detalladas de las aves, y que tienen la capacidad de procesamiento y almacenamiento suficiente para la app y los datos que genera.

3. Especificación de requerimientos.

Tabla 2 Tablas de requerimientos

ID	Nombre del requerimiento	Tipo de requerimiento
RF-01	Registro, Inicio y Actualización de Usuarios	Funcional
RF-02	Registro, Actualización y Eliminación de Bitácoras de Campo	Funcional
RF-03	Registro, Actualización y Eliminación de Muestreos.	Funcional
RF-04	Identificación de Aves	Funcional
RF-05	Exportación de Bitácoras de Campo	Funcional
RNFD-01	Preprocesamiento de Imágenes	No funcional de desempeño
RNFD-02	Localización en tiempo real	No funcional de desempeño

3.1 Requerimientos Funcionales.

RF-01 Registro y Actualización de Usuarios:

El sistema debe permitir el registro de usuarios, con su respectiva confirmación por correo electrónico, y la actualización de los atributos nombre, apellido, correo electrónico y contraseña.

Tabla 3 Tabla de CU-01 del requerimiento RF-01

ID de requerimientos:	RF-01	ID de CU:	CU-01
		Prioridad	Media
Nombre:	Inicio de sesión	Estado	Pendiente
Descripción:	Permitir a los usuarios iniciar sesión en la aplicación proporcionando su correo y contraseña de la aplicación, previamente registrados.		

Necesidades que resuelve:	Permite el acceso personalizado al sistema para que el registro de observaciones sea por investigador.
Métrica de satisfacción:	Ingreso a la aplicación y visualizar sus bitácoras de campo correspondientes.
Pre-condiciones:	Acceso a internet en un primer inicio de sesión. Haber registrado una cuenta en la aplicación.
Post-condiciones:	Éxito: Puede acceder al sistema. Fallo: El usuario no puede activar la cuenta debido a problemas con el correo de verificación.
Flujo Normal:	1. El usuario ingresa a la aplicación. 2. El usuario ingresa su correo y contraseña. 3. El usuario ingresa a su cuenta en la aplicación en la opción de “Iniciar sesión”.
Flujos Alternativos:	FA1: La contraseña es errónea. El usuario vuelve a ingresar su correo y contraseña.
Flujos de Excepción:	FE1: Usuario no registrado El sistema notifica al usuario que el correo que intenta ingresar no está registrado.
Notas o Problemas:	Asegurar la compatibilidad con diferentes proveedores de correo.

Tabla 3 Tabla de CU-02 del requerimiento RF-01

ID de requerimientos:	RF-01	ID de CU:	CU-02
		Prioridad	Media
Nombre:	Registro de usuario	Estado	Pendiente
Descripción:	Permitir a los usuarios crear una cuenta proporcionando un correo electrónico, nombre de usuario y contraseña. Un correo de verificación será enviado para activar la cuenta.		
Necesidades que resuelve:	Permite el acceso personalizado al sistema para que el registro de observaciones sea por investigador.		
Métrica de satisfacción:	Usuario registrado y cuenta activada tras confirmación por correo.		
Pre-condiciones:	Acceso a internet y disponibilidad de un correo electrónico válido.		
Post-condiciones:	Éxito: Usuario activa su cuenta y puede acceder al sistema. Fallo: El usuario no puede activar la cuenta debido a problemas con el correo de verificación.		
Flujo Normal:	4. El sistema muestra la opción "Registrarse" e "Iniciar sesión" en la pantalla inicial. 5. El usuario selecciona la opción de "Crear cuenta". 6. El usuario proporciona nombre, apellido, correo, y contraseña. 7. El sistema envía un correo de verificación. 8. El usuario verifica la cuenta a través del correo. 9. El usuario ingresa a su cuenta en la aplicación en la opción de "Iniciar sesión".		
Flujos Alternativos:	FA1: Correo ya registrado. El sistema avisa al usuario que el correo ya tiene una cuenta registrada.		
Flujos de Excepción:	FE1: El correo de verificación no se recibe.		

	El usuario solicita el reenvío del correo.
Notas o Problemas:	Asegurar la compatibilidad con diferentes proveedores de correo.

Tabla 4 Tabla de CU-03 del requerimiento RF-01

ID de requerimiento:	RF-01	ID de CU:	CU-03
		Prioridad	Baja
Nombre:	Actualización de usuario	Estado	Pendiente
Descripción:	Permitir a los usuarios editar sus campos de perfil y guardar los cambios de su cuenta.		
Necesidades que resuelve:	Permite el cambio de los campos del perfil del usuario para su personalización.		
Métrica de satisfacción:	Perfil de usuario actualizado con los nuevos cambios.		
Pre-condiciones:	El usuario debe estar registrado y haber iniciado sesión en el sistema.		
Post-condiciones:	Éxito: El usuario puede ver los campos actualizados Fallo: El usuario no puede ver los cambios realizados y ve los datos anteriores a la actualización.		
Flujo Normal:	1. El usuario da click en el ícono de menú. 2. El usuario selecciona la opción de “Editar perfil”. 3. El usuario llena los campos que desea cambiar. 4. El usuario guarda los cambios y el sistema los muestra en el perfil.		
Flujos Alternativos:	FA1: -El usuario no edita ninguno de los campos. El usuario regresa a la pantalla principal sin editar ningún campo ni guardar los cambios.		
Flujos de Excepción:	FE1: El correo de verificación no se recibe. El sistema no muestra ningún cambio.		
Notas o Problemas:			

RF-02 Registro, Actualización y Eliminación de Bitácoras de Campo:

El sistema debe permitir el registro, actualización y eliminación de las bitácoras de campos, que contienen información con los campos nombre de la bitácora, fecha, hora, lugar, ubicación, cantidad de muestras, observaciones y los muestras correspondientes.

Tabla 5 Tabla de CU-04 del requerimiento RF-02

ID de requerimiento:	RF-02	ID de CU:	CU-04
		Prioridad:	Alta
Nombre:	Registro de bitácoras de campo	Estado	Pendiente
Descripción:	Permite a los usuarios crear bitácoras de campo dentro de la aplicación para registrar múltiples observaciones de aves en una sola sesión. Cada bitácora puede incluir detalles como la fecha, ubicación, y notas generales, así como múltiples registros de observación de aves específicas.		
Necesidades que resuelve:	Facilita a los usuarios la organización de sus observaciones de aves basadas en sesiones de observación específicas, para su referencia futura.		
Métrica de satisfacción:	El usuario puede completar y guardar un registro de observación sin errores o dificultades.		

Pre-condiciones:	El usuario debe estar registrado y haber iniciado sesión en el sistema.
Post-condiciones:	<p>Éxito: La bitácora de campo se guarda con éxito en el sistema con todas las observaciones de aves pertinentes.</p> <p>Fallo: La bitácora no se guarda debido a un error del sistema o a una entrada de usuario incorrecta.</p>
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona la opción "Crear Bitácora". 2. Completa los detalles generales de la bitácora . 3. Añade muestreos de observación de aves a la bitácora. (CU-06) 4. Guarda la bitácora de campo completa.
Flujos Alternativos:	<p>FA1: El usuario decide crear una bitácora vacía.</p> <p>El usuario realiza los pasos 1 y 2, guarda la bitácora sin agregar muestreos.</p>
Flujos de Excepción:	<p>FE1: Error al guardar la bitácora de campo.</p> <p>El sistema informa al usuario y sugiere intentar de nuevo.</p>
Notas o Problemas:	Evaluando el uso de geolocalización para facilitar el registro de ubicaciones.

Tabla 6 Tabla de CU-05 del requerimiento RF-02

ID de requerimiento:	RF-02	ID de CU:	CU-05
		Prioridad:	Media
Nombre:	Actualización de bitácoras de campo	Estado	Pendiente
Descripción:	Permite a los usuarios actualizar las bitácoras de campo dentro de la aplicación para editar detalles como la fecha, ubicación, y notas generales.		
Necesidades que resuelve:	Facilita a los usuarios la organización de sus observaciones de aves basadas en sesiones de observación específicas, para su referencia futura.		
Métrica de satisfacción:	El usuario puede observar los cambios de la actualización de la información de una bitácora de campo.		
Pre-condiciones:	El usuario tiene bitácoras de campos previamente creadas.		
Post-condiciones:	<p>Éxito: La bitácora de campo se guarda con éxito en el sistema con los cambios realizados.</p> <p>Fallo: La bitácora no se guarda debido a un error del sistema o a una entrada de usuario incorrecta y se mantienen los datos anteriores a la actualización.</p>		
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona la bitácora de campo que desea editar. 2. El usuario selecciona y edita los campos de información que desea. 3. El usuario elimina o agrega muestreos. 4. El usuario da click en el botón de guardar. 		
Flujos Alternativos:	<p>FA1: El usuario solo edita los campos de información de la bitácora, sin modificar los muestreos.</p> <p>El usuario realiza los pasos 1, 2 y 4, guarda la bitácora sin modificar los muestreos.</p>		
Flujos de Excepción:	<p>FE1: Error al guardar la bitácora de campo.</p> <p>El sistema no muestra ningún cambio.</p>		
Notas o Problemas:			

Tabla 7 Tabla de CU-06 del requerimiento RF-02

ID de requerimiento:	RF-02	ID de CU:	CU-06
		Prioridad:	Media
Nombre:	Eliminación de bitácoras de campo	Estado	Pendiente
Descripción:	Permite a los usuarios eliminar las bitácoras de campo previamente creadas.		
Necesidades que resuelve:	Facilita a los usuarios la organización de sus observaciones de aves basadas en sesiones de observación específicas, para su referencia futura.		
Métrica de satisfacción:	El usuario elimina una bitácora de campo exitosamente.		
Pre-condiciones:	El usuario tiene bitácoras de campos previamente creadas.		
Post-condiciones:	Éxito: La bitácora de campo se elimina con éxito en el sistema. Fallo: La bitácora no se guarda debido a un error del sistema o a una entrada de usuario incorrecta y se mantienen los datos anteriores a la actualización.		
Flujo Normal:	1. El usuario identifica la bitácora de campo que desea eliminar. 2. El usuario pulsa el botón de eliminar en la esquina superior derecha. 3. El sistema muestra una ventana de confirmación. 4. El usuario confirma la eliminación de la bitácora.		
Flujos Alternativos:	FA1: El usuario se arrepiente y no desea borrar la bitácora. 4.1 El usuario cancela la eliminación de la bitácora.		
Flujos de Excepción:	FE1: Error al eliminar la bitácora de campo. El sistema no muestra ningún cambio.		
Notas o Problemas:			

RF-03 Registro, Actualización y Eliminación de Muestreos:

El sistema debe permitir el registro, actualización y eliminación de los muestreos, que contienen información con los campos nombre del muestreo, nombre coloquial del ave, fecha, hora, coordenadas, ubicación, color, dimensiones y grado de peligro en extinción (grado de endemismo).

Tabla 8 Tabla de CU-07 del requerimiento RF-03

ID de requerimiento:	RF-03	ID de CU:	CU-07
		Prioridad:	Alta
Nombre:	Registro de Muestreo	Estado	Pendiente
Descripción:	Permite a los usuarios registrar detalles específicos de sus observaciones de aves, incluyendo especie, fecha, hora, ubicación, y notas adicionales.		
Necesidades que resuelve:	Proporciona una herramienta para que los usuarios documenten y guarden un registro de sus observaciones de aves.		
Métrica de satisfacción:	El usuario puede completar y guardar un muestreo sin errores o dificultades.		
Pre-condiciones:	El usuario debe de haber seleccionado una bitácora de campo donde ingresar el muestreo. El usuario debe haber identificado una ave y estar registrado en el sistema.		
Post-condiciones:	Éxito: El muestreo se guarda en la base de datos del usuario. Fallo: El muestreo no se guarda debido a un error del sistema.		

Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona "Crear muestreo". 2. El usuario da click en el botón de cámara, el sistema le muestra la opción de usar la cámara o abrir la galería para seleccionar la imagen a identificar. 3. El usuario toma la foto del ave o se selecciona una existente en el dispositivo y el sistema la identifica. (CU-04) 4. El sistema llena los campos del ave automáticamente. 5. El usuario completa el formulario de observación con los campos faltantes. 6. El sistema guarda la observación en la bitácora de campo correspondiente.
Flujos Alternativos:	<p>FA1: El usuario desea cambiar los datos proporcionados por el sistema. El usuario edita los campos permitidos, como nombre del muestreo, color, dimensiones.</p>
Flujos de Excepción:	<p>FE1: Error al guardar la observación. El sistema informa al usuario y sugiere intentarlo nuevamente.</p>
Notas o Problemas:	Uso de la cámara y galería del teléfono del usuario.

Tabla 8 Tabla de CU-08 del requerimiento RF-03

ID de requerimiento:	RF-03	ID de CU:	CU-08
		Prioridad:	Media
Nombre:	Actualización de muestreo	Estado	Pendiente
Descripción:	Permite a los usuarios editar detalles específicos de sus observaciones de aves.		
Necesidades que resuelve:	Proporciona una herramienta para que los usuarios actualicen y guarden un muestreo previo.		
Métrica de satisfacción:	El usuario puede realizar cambios y actualizar un muestreo sin errores o dificultades.		
Pre-condiciones:	<p>El usuario debe de haber seleccionado una bitácora de campo donde ingresar el muestreo.</p> <p>El usuario debe de haber seleccionado un muestreo existente.</p>		
Post-condiciones:	<p>Éxito: El muestreo se actualiza y se muestran los cambios.</p> <p>Fallo: El muestreo no se guarda debido a un error del sistema.</p>		
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona el muestreo que desea editar. 2. El usuario selecciona y edita los campos de información que desea. 3. El usuario da click en el botón de guardar. 		
Flujos Alternativos:	<p>FA1: El usuario no desea cambiar ningún campo.</p> <p>1.1 El usuario sólo visualiza el muestreo realizado.</p> <p>1.2 El usuario regresa a la pantalla anterior de “Bitácora”.</p>		
Flujos de Excepción:	<p>FE1: Error al actualizar el muestreo.</p> <p>El sistema informa al usuario y sugiere intentarlo nuevamente.</p>		
Notas o Problemas:			

Tabla 8 Tabla de CU-09 del requerimiento RF-03

ID de requerimiento:	RF-03	ID de CU:	CU-09
		Prioridad:	Media
Nombre:	Eliminación de muestreo	Estado	Pendiente

Descripción:	Permite a los usuarios eliminar las bitácoras de campo previamente creadas.
Necesidades que resuelve:	Facilita a los usuarios la organización de sus muestreos para su referencia futura.
Métrica de satisfacción:	El usuario elimina un muestreo exitosamente.
Pre-condiciones:	El usuario tiene bitácoras de campos previamente creadas. El usuario tiene bitácoras de campos previamente creadas dentro de la bitácora seleccionada.
Post-condiciones:	Éxito: La bitácora de campo se elimina con éxito en el sistema. Fallo: La bitácora no se guarda debido a un error del sistema o a una entrada de usuario incorrecta y se mantienen los datos anteriores a la actualización.
Flujo Normal:	1. El usuario identifica el muestreo que desea eliminar. 2. El usuario pulsa el botón de eliminar en la esquina superior derecha. 3. El sistema muestra una ventana de confirmación. 4. El usuario confirma la eliminación del muestreo.
Flujos Alternativos:	FA1: El usuario se arrepiente y no desea borrar el muestreo. 4.1 El usuario cancela la eliminación del muestreo.
Flujos de Excepción:	FE1: Error al eliminar la bitácora de campo. El sistema no muestra ningún cambio.
Notas o Problemas:	

RF-04 Identificación de Aves:

El sistema permite a los usuarios tomar una foto o seleccionar una existente en su teléfono y el sistema procesa la imagen para su clasificación e identificar la especie.

Tabla 9 Tabla de requerimiento RF-04

ID de requerimiento:	RF-04	ID de CU:	CU-10
		Prioridad:	Media
Nombre:	Identificación de Aves	Estado	Pendiente
Descripción:	Los usuarios pueden tomar o cargar una foto de un ave, y el sistema procesa la imagen para identificar la especie.		
Necesidades que resuelve:	Facilita la identificación rápida y precisa de especies de aves para los usuarios.		
Métrica de satisfacción:	El sistema identifica correctamente la especie de ave de las 6 especies disponibles.		
Pre-condiciones:	El usuario tiene acceso a un dispositivo con cámara o imágenes previamente capturadas de aves. El usuario ha seleccionado una bitácora previa y seleccionó la opción de “Aregar muestreo”		
Post-condiciones:	Éxito: El sistema muestra el nombre y datos relevantes de la especie identificada. Fallo: El sistema no logra identificar la especie de la imagen proporcionada.		
Flujo Normal:	1. El usuario toma una foto o carga una existente. 2. El sistema procesa la imagen y muestra los resultados. 3. El usuario revisa la información proporcionada.		

Flujos Alternativos:	FA1: La imagen no es clara. El sistema solicita al usuario tomar o cargar otra imagen.
Flujos de Excepción:	FE1: Fallo en el procesamiento de la imagen. El usuario es notificado del error y puede intentarlo nuevamente.
Notas o Problemas:	Considerar la integración de una guía para tomar fotos adecuadas para la identificación.o.

RF-05 Exportación de Bitácoras de Campos:

El sistema permite a los usuarios tomar una foto o seleccionar una existente en su teléfono y el sistema procesa la imagen para su clasificación e identificar la especie.

Tabla 10 Tabla de requerimiento RF-05

ID de requerimiento:	RF-05	ID de CU:	CU-11
		Prioridad:	Alta
Nombre:	Exportación de Bitácoras de Campo	Estado	Pendiente
Descripción:	Permite a los usuarios crear y mantener bitácoras de campo que puedan incluir varios muestreos de observación de aves, cada uno con detalles específicos como la especie, fecha, hora y ubicación, así mismo se le permite al usuario exportar dicha bitácora en PDF.		
Necesidades que resuelve:	Facilita a los usuarios el seguimiento detallado de sus actividades de observación de aves, permitiendo un registro organizado de múltiples muestreos en una sola bitácora, así mismo facilitando la obtención de la bitácora la cual se puede almacenar o exportar en formato PDF.		
Métrica de satisfacción:	El usuario puede exportar la bitácora deseada en formato PDF sin dificultades.		
Pre-condiciones:	El usuario debe de tener en una bitácora previamente creada y llenada.		
Post-condiciones:	Éxito: La bitácora de campo se exporta correctamente en formato PDF, con todos los registros de observación incluidos. Fallo: La bitácora no se exporta debido a errores del sistema o entrada inválida del usuario.		
Flujo Normal:	1. El usuario selecciona una bitácora preexistente. 2. El usuario selecciona la opción "Exportar Bitácora". 3. El sistema genera y descarga la bitácora en formato PDF.		
Flujos Alternativos:	FA1: El usuario exporta la bitácora deseada desde la pantalla "Bitácoras de campo" 1.1 El usuario identifica la bitácora que desea exportar. 1.2 El usuario da click en el botón de "Exportar PDF". 1.3 El sistema genera y descarga la bitácora en formato PDF.		
Flujos de Excepción:	FE1: Error al exportar la bitácora en formato PDF. 3.1 El sistema notifica al usuario del error. 3.2 Se sugiere al usuario intentar la exportación nuevamente.		
Notas o Problemas:	Asegurar que la funcionalidad de exportación en PDF. Considerar la compatibilidad del formato PDF con diferentes dispositivos y lectores PDF.		

4.1 Requerimientos de desempeño.

RNFD-01. Preprocesamiento de Imágenes: Antes de la clasificación de aves, las imágenes capturadas serán sometidas a un proceso de preprocesamiento para mejorar su calidad y facilitar la identificación precisa de especies.

RNFD-02 Localización en Tiempo Real: Obtención de las coordenadas de ubicación del usuario a través de las funciones de GPS del teléfono móvil, permitiendo una referencia espacial de cada observación realizada en campo.

4.2 Requerimientos de la base de datos lógica.

Esta sección describe los requerimientos lógicos para la base de datos que respalda la aplicación "IdBird", enfocándose en cómo se manejará, almacenará, y accederá a la información dentro del sistema.

Tipos de Información:

- Datos de usuario: Incluyen información de registro y perfil, como correo electrónico, nombre y apellido del usuario y contraseña. Accedidos cada vez que un usuario inicia sesión o actualiza su perfil.
- Bitácoras de campo: Agrupaciones de observaciones relacionadas conformadas por múltiples muestras. Actualizadas o consultadas por el usuario que desee revisar o exportar sus registros acumulados.
- Muestreo de aves: Datos recopilados por el usuario, incluyendo especie de ave, fecha, hora, ubicación y notas adicionales. Consultados regularmente por el usuario para añadir nuevas entradas o revisar observaciones pasadas.

4.3 Restricciones de diseño.

a. Cumplimiento de estándares.

Formatos de reporte.

El sistema generará bitácoras exportadas en PDF que cumplirán con el estándar PDF/A y el formato indicado para el registro de aves, para asegurar la conservación a largo plazo de los documentos electrónicos y una visualización formal de los muestras registrados en cada una de las bitácoras.

Se anexa el formato para el registro de aves. Anexo B de Anexos.

4.4 Atributos

a. Portabilidad.

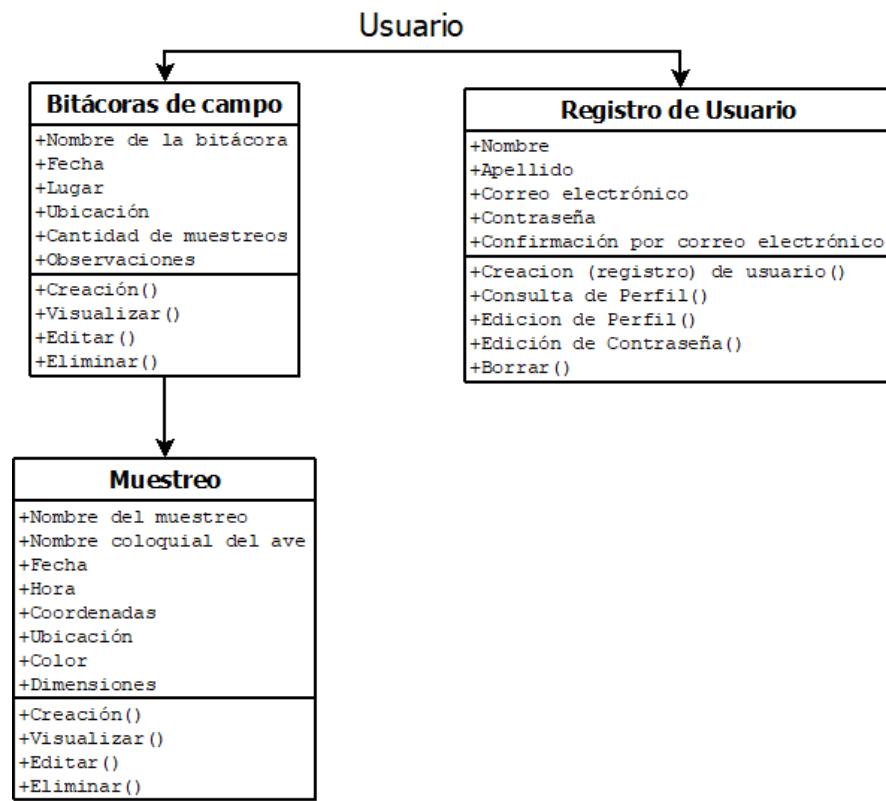
- Uso de Lenguajes Portátiles: El sistema será desarrollado empleando lenguajes de programación portátiles y de amplio soporte, tales como Dart, para el desarrollo de la aplicación, y se hará uso del framework Flutter, para garantizar la compatibilidad con múltiples dispositivos móviles y el funcionamiento correcto de la aplicación y el modelo de reconocimiento de imágenes.
- Base de datos: La base de datos se implementará de manera local, lo que permitirá que la aplicación sea funcional en dispositivos de gama media-alta, incluso sin necesidad de una conexión a internet.

4.2 Requisitos de Interfaces externas.

- Enviar correo electrónico para confirmación de cuenta: El sistema debe ser capaz de interactuar con una API externa para enviar correos electrónicos de confirmación cuando un usuario crea una cuenta en la aplicación móvil.
- Uso de cámara y galería del teléfono: El sistema debe ser capaz de abrir la cámara y la galería del teléfono para obtener la imagen del ave para su clasificación posterior.

5. Anexos.

A. Diagrama del sistema como descripción general.



B. Formato para el registro de aves mediante el método de conteo por puntos

Formato para el registro de aves mediante el método de conteo por puntos.

Estado _____ Municipio _____ Fecha _____
Localidad _____
Hora de inicio del muestreo _____ Hora de término del muestreo _____
Tipo de hábitat _____ Vegetación dominante _____
Nombre del monitor _____ Página ____ de ____

Anexo D Documento de plan de riesgos

CONTROL DE VERSIONES					
Autor(es)	Fecha de modificación	Versión	Descripción del cambio	Revisó	Estado
VSSL	15/04/2024	1.0	Creación del Documento	IIB	PENDIENTE

Propósito

Definir un marco metodológico para la correcta evaluación de los riesgos que se pueden encontrar dentro de un proyecto, en el contexto de Trabajo Terminal I y II.

De la evaluación de los riesgos

Se deben llenar 4 tablas que nos ayudarán a medir la probabilidad y nivel de riesgo de sucesos que pueden ocurrir a lo largo del desarrollo del proyecto e incluso una vez terminado.

Dichas tablas contendrán los niveles de probabilidad, los niveles de impacto, el nivel de riesgo y una tabla en la cual se registrarán los posibles riesgos que amenacen el proyecto.

Niveles de probabilidad

Los niveles de probabilidad deberán expresar el nivel que se define para la ocurrencia de un suceso, para los proyectos de Trabajo Terminal de la UPIIZ, se sugiere utilizar la siguiente tabla:

Nivel	Probabilidad	Descripción
1	Raro	Solo ocurrirá en casos excepcionales
2	Improbable	Puede ocurrir en algún momento pero las condiciones del proyecto no dan pie a que suceda
3	Possible	Podría ocurrir en algún momento del proyecto
4	Probable	Es probable que ocurra en la mayoría de las circunstancias del proyecto
5	Casi Seguro	Se espera que ocurra para todas las posibles circunstancias

Niveles de impacto

El nivel de impacto, como su nombre lo indica nos permite identificar que tanto impactaría en el proyecto, la ocurrencia de algún suceso riesgoso para el proyecto, para los proyectos de Trabajo Terminal de la UPIIZ, se sugiere utilizar la siguiente tabla:

Nivel	Impacto	Descripción
1	Insignificante	Si el hecho se llega a presentar no afecta la realización del proyecto
2	Menor	Si el hecho se llega a presentar el impacto no es significativo para la realización del proyecto no, genera una desviación significativa

3	Moderado	Si el hecho se llega a presentar el impacto es aun controlable y no afecta de manera grave la realización del proyecto.
4	Mayor	Si el hecho se llega a presentar el impacto es mucho mayor e implica cambios significativos en la realización del proyecto.
5	Catastrófico	Si el hecho se llega a presentar el impacto es grave y compromete la realización del proyecto.

Nivel de riesgo

Una vez definidos los niveles de probabilidad, y los niveles de impacto debemos calcular el nivel del riesgo, para ello se debe realizar una multiplicación simple de los niveles anteriores, con ello evaluaremos los riesgos que detectemos dentro de nuestro proyecto, siempre hay que considerar que a menor probabilidad e impacto, menor será el nivel del riesgo, a mayor probabilidad e impacto, mayor será el nivel de riesgo.

Probabilidad	Impacto				
	Insignificante (1)	Menor (2)	Moderado (3)	Mayor (4)	Catastrófico (5)
Raro (1)	1	2	3	4	5
Improbable (2)	2	4	6	8	10
Possible (3)	3	6	9	12	15
Probable (4)	4	8	12	16	20
Casi Seguro (5)	5	10	15	20	25

De esta manera obtendremos la siguiente matriz de nivel de riesgo

Nivel de riesgo	Probabilidad X Impacto
Muy Alto	> = 20
Alto	De 15 a 19
Medio	De 9 a 14
Bajo	De 6 a 8
Muy bajo	< = 5

Matriz de riesgos

Id riesgo	Descripción	Fase afectada	Causa del riesgo	Probabilidad	Impacto	Nivel del riesgo	Estrategia de prevención	Estrategia de Mitigación
R-001	Falta de algún integrante en alguna reunión de equipo de trabajo	Análisis Diseño Implementación Pruebas Despliegue del sistema	Falta por una enfermedad o emergencia personal	Possible 2	Menor 2	Bajo 6	Comprobar continuamente el estado de salud de los integrantes y horarios o .	Planeación y división de las actividades necesarias .

							eventualidades personales	Tomar en cuenta los tiempos necesarios El otro integrante del equipo debe avanzar
R-002	Conflictos personales entre el equipo de trabajo	Análisis Diseño Implementación Pruebas Despliegue del sistema	Problemas personales que puedan afectar el trabajo en equipo	Possible 3	Moderado 3	Medio 9	Comprobar una buena relación entre los miembros del equipo continuamente	División del trabajo de manera individual Resolución de los conflictos de manera personal
R-003	Conflicto de puntos de vista del proyecto en el equipo de trabajo	Análisis Diseño Implementación Pruebas Despliegue del sistema	Diferentes puntos de vista e ideas para el proyecto	Possible 3	Insignificante 1	Muy bajo 3	Planear las actividades previamente y llegar a un consenso	Pedir punto de vista a otro integrante del equipo, como asesor o director.
R-004	Subestimar la cantidad de tiempo requerida para una actividad	Análisis Diseño Implementación Pruebas Despliegue del sistema	Una actividad lleva más tiempo del planeado	Possible 3	Moderado 3	Medio 9	Medir los tiempos necesarios en cada actividad Considerar los tiempos disponibles	Acortar el alcance de la actividad Si es está disponible tomar más tiempo Apoyarse de otro integrante del equipo
R-005	Sobreestimar la cantidad de tiempo requerida para una actividad	Análisis Diseño Implementación Pruebas Despliegue del sistema	Una actividad lleva menos tiempo del planeado	Improbable 2	Moderado 3	Bajo 6	Medir los tiempos necesarios en cada actividad Considerar los tiempos disponibles	Usar el tiempo disponible para avanzar en actividades faltantes o pendientes
R-006	Es necesario agregar	Análisis Diseño	No se contempló alguna	Raro 1	Mayor 4	Muy bajo 4	Comprobar con clientes y asesores si	Adaptar la funcionalidad

	funcionalidades al proyecto	Implementación	funcionalidad anteriormente				el proyecto va siendo completo	necesaria al proyecto actual sin comprometerlo
R-007	El dispositivo móvil de pruebas no soporta la aplicación o alguna parte del proyecto	Implementación Pruebas Despliegue del sistema	El procesamiento del dispositivo móvil no soporta los modelos CNN	Improbable 2	Mayor 4	Bajo 8	Comprobar los requerimientos de hardware y software para la aplicación y sus modelos	Adaptar los modelos CNN y la aplicación para que cumpla los requerimientos del dispositivo móvil de pruebas Utilizar el segundo dispositivo móvil de pruebas
R-008	Ya no se cuenta con el dispositivo móvil de pruebas	Implementación Pruebas Despliegue del sistema	El dispositivo móvil de pruebas se avería, extravía o tiene fallas internas	Improbable 2	Mayor 4	Bajo 8	Cuidado de los dispositivos de pruebas Comprobación de disponibilidad de dispositivos de reemplazo	Intentar recuperar el dispositivo en buenas condiciones Utilizar el segundo dispositivo móvil de pruebas Utilizar un tercer dispositivo con las características necesarias
R-009	La cantidad de imágenes para el entrenamiento de los modelos es mínima.	Análisis Implementación	Las imágenes con las características necesarias son menores a las esperadas	Possible 3	Moderado 3	Medio 9	Investigación en múltiples plataformas y fuentes para obtener imágenes que cumplan con las	Modificar las imágenes existentes para multiplicar la cantidad de imágenes.

							características Toma continua a lo largo del proyecto de fotos que cumplan con las características.	Entrenar los modelos con las imágenes existentes, aunque comprometen la exactitud de los modelos.
R-010	Se necesita un modelo CNN extra al planeado	Análisis Diseño Implementación	Se necesita programar y entrenar un modelo de identificación de objetos antes del modelo de clasificación de especies	Casi seguro 5	Menor 2	Medio 10	Se contempla desde la etapa de análisis la necesidad de más herramientas o modelos	Se implementan las herramientas y modelos previamente investigados
R-011	Algún componente del proyecto no es compatible con los dispositivos móviles	Análisis Diseño Implementación Pruebas	La aplicación, el modelo, la API o algún otro componente no funciona con la versión de android o alguna característica del dispositivo	Improbable 2	Mayor 4	Bajo 8	Se analizan los requisitos de los componentes y se comparan con las características de los dispositivos móviles comprobando la compatibilidad	Se utiliza un software para adaptarlo según las necesidades Se implementa alguna variación compatible Se utiliza alguna tecnología alternativa

Anexo E Documento de diseño

CONTROL DE VERSIONES					
Autor(es)	Fecha de modificación	Versión	Descripción del cambio	Revisó	Estado
AFFJ	28/04/2024	1.0	Creación del Documento	VSSL	APROBADO
VSSL	31/05/2024	1.1	Corrección del Documento		PENDIENTE

Propósito

Detallar el cómo los requisitos de software deben ser implementados y determinar si se han abordado todos los requisitos necesarios, también proporciona al equipo de desarrollo una orientación general sobre la arquitectura del proyecto de software y los modelos a seguir, facilita a identificar los elementos de diseño que no están alineados con un requerimiento, para determinar si el elemento de diseño es necesario; se convierte en la base para limitar los cambios en el alcance de un proyecto, ya que sirve como una guía durante toda la vida del sistema para los miembros del proyecto. El documento debe ser tan detallado como sea posible, a fin de mantener el equipo de desarrollo de software centrado y alineado al objetivo del sistema.

8

Se mostrará la arquitectura del proyecto de trabajo terminal I, para lo cual se tomará en cuenta la definición de arquitectura del software que la IEEE std. 1471-2000, y que a la letra dice: *La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente, y los principios que orientan su diseño y evolución.*

La arquitectura de software se refiere a la “estructura general del software y las formas en la que la estructura proporciona una integridad conceptual para un sistema”. En su forma más simple, la arquitectura es la estructura u organización de los componentes del programa (módulos), la manera en que estos componentes interactúan; así como, la estructura de los datos que utilizan los componentes. La especificación de la arquitectura del software es importante porque:

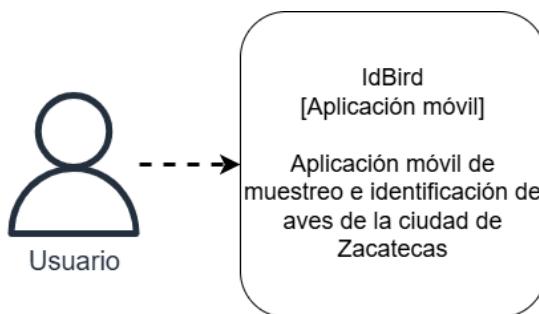
- Las representaciones de la arquitectura del software permiten la comunicación entre todas las partes integrantes o participantes interesadas en el desarrollo de un sistema de cómputo.
- Destaca las decisiones iniciales relacionadas con el diseño que tendrán un impacto profundo en todo el trabajo de la Ingeniería de Software.
- Constituye un modelo relativamente pequeño e intelectualmente comprensible de cómo está estructurado el sistema y cómo trabajan juntos sus componentes.

La arquitectura 4C (Contexto, Contenedores, Componentes, Clases) se utiliza en este proyecto debido a su capacidad para ofrecer una visión clara y comprensible del sistema en diferentes niveles de detalle. Cada uno de los niveles de la arquitectura 4C proporciona una perspectiva distinta y esencial del sistema, lo que facilita su comprensión, desarrollo y mantenimiento.

El uso de la arquitectura 4C permite una comprensión progresiva y detallada del sistema, desde una vista general del contexto hasta la implementación específica de clases. Esto no solo mejora la comunicación entre los diferentes miembros del equipo, sino que también asegura que el sistema esté bien organizado y sea mantenable a largo plazo. La clara separación y definición de responsabilidades en cada nivel de la arquitectura 4C también ayuda a identificar y resolver problemas de manera más eficiente, mejorando la calidad y la fiabilidad del sistema.

Diagrama de Contexto:

El diagrama de contexto nos permite identificar y definir las interacciones del sistema con los actores externos. Esto es crucial para entender cómo los usuarios y otros sistemas interactúan con IdBird, proporcionando una visión global del entorno del sistema.



El diagrama presentado es un diagrama de contexto que describe la relación entre el usuario y la aplicación móvil IdBird, enfocada en el muestreo e identificación de aves en la ciudad de Zacatecas.

Descripción del Diagrama de Contexto

Usuario:

Representado por un ícono de persona, el usuario es cualquier individuo que interactúa con la aplicación móvil IdBird. Este puede ser un observador de aves, investigador, estudiante o cualquier persona interesada en registrar y conocer más sobre las aves de Zacatecas.

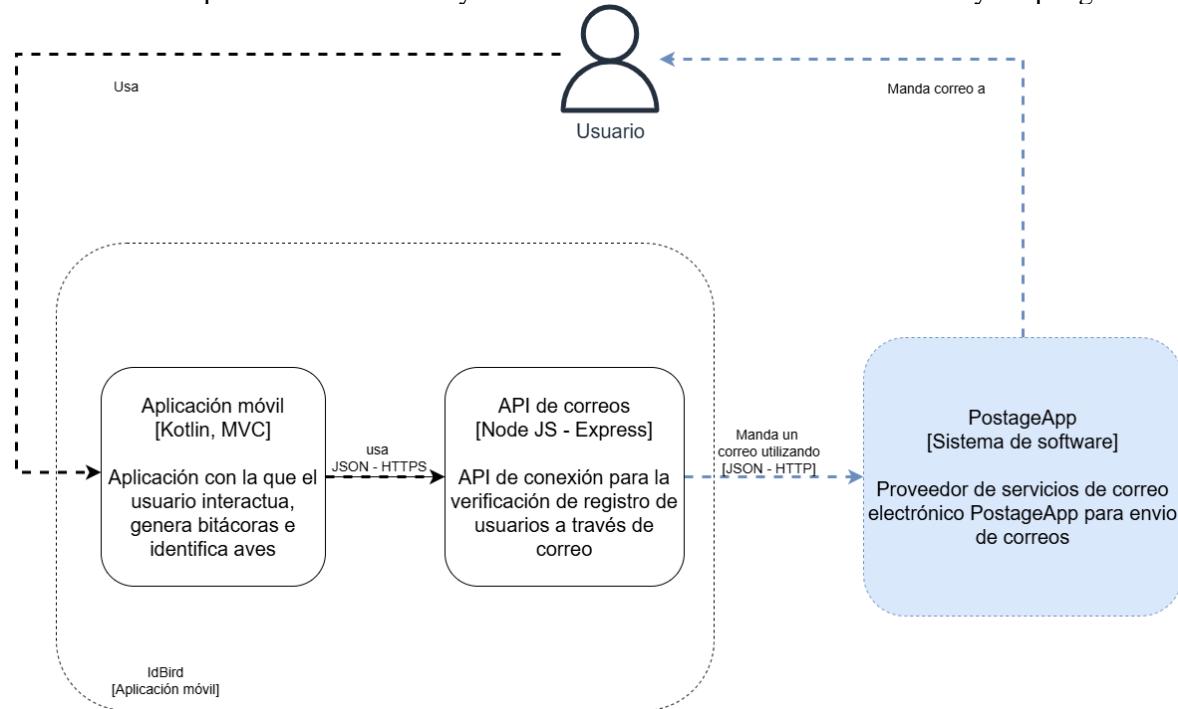
IdBird (Aplicación móvil):

Representada como un sistema dentro de un recuadro, IdBird es una aplicación móvil desarrollada específicamente para facilitar el muestreo e identificación de aves en Zacatecas. La aplicación

proporciona herramientas para registrar avistamientos de aves, tomar fotografías, ingresar datos de observación y obtener identificaciones precisas de las especies observadas.

Diagrama de contenedores:

El diagrama de contenedores descompone el sistema en sus principales bloques de construcción, como aplicaciones web, bases de datos, y servicios backend. Esto ayuda a visualizar la infraestructura del sistema y cómo los diferentes contenedores se comunican entre sí, permitiendo una gestión efectiva de la arquitectura del sistema y facilitando decisiones sobre escalabilidad y despliegue.



El diagrama presentado es un diagrama de contenedor que ilustra la arquitectura de alto nivel de la aplicación móvil IdBird y sus interacciones con otros sistemas. Este diagrama muestra cómo diferentes componentes del sistema están organizados e interactúan entre sí y con los usuarios.

Descripción del Diagrama de Contenedor

Usuario:

Representado por un ícono de persona, el usuario interactúa con la aplicación móvil IdBird. Los usuarios pueden ser observadores de aves, investigadores, estudiantes, o cualquier persona interesada en el muestreo y la identificación de aves.

Aplicación móvil IdBird (Kotlin, MVC):

Este contenedor representa la aplicación móvil desarrollada en Kotlin utilizando el patrón de arquitectura MVC (Modelo-Vista-Controlador). Es la interfaz principal con la que el usuario interactúa para generar bitácoras e identificar aves. La aplicación permite a los usuarios registrar observaciones, tomar fotos de aves y obtener identificaciones.

API de correos (Node JS - Express):

Este contenedor es una API desarrollada en Node.js con el framework Express. Su función principal es gestionar la verificación del registro de usuarios a través del envío de correos electrónicos. La aplicación móvil IdBird se comunica con esta API para enviar solicitudes de verificación de registro cuando un usuario se registra en la aplicación.

La aplicación móvil IdBird se comunica con la API de correos mediante solicitudes JSON a través de HTTPS. Esta interacción es crucial para las funcionalidades de registro y verificación de usuarios, donde la aplicación necesita enviar correos electrónicos de confirmación a los nuevos usuarios.

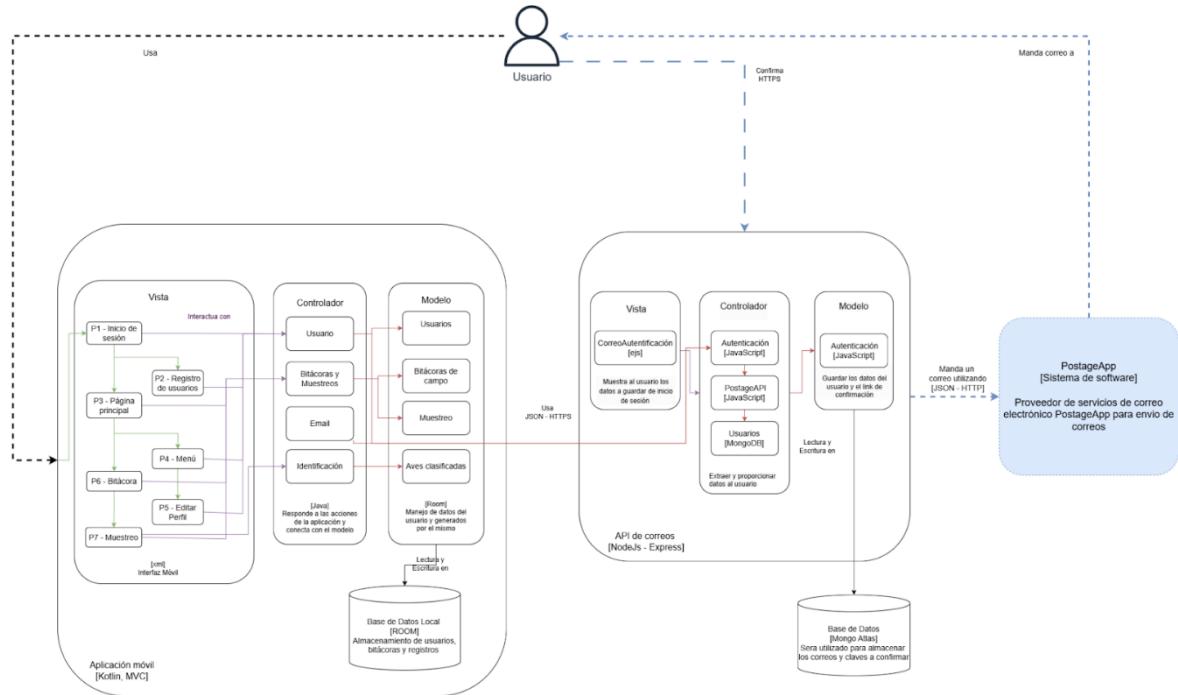
PostageApp (Sistema de software):

PostageApp es un proveedor de servicios de correo electrónico que la API de correos utiliza para enviar correos electrónicos. La API de correos envía las solicitudes de envío de correos a PostageApp utilizando el formato JSON a través del protocolo HTTPS. PostageApp luego maneja el envío real de los correos electrónicos a los usuarios.

La API de correos envía solicitudes de envío de correos a PostageApp utilizando JSON a través de HTTPS. PostageApp, como proveedor de servicios de correo electrónico, procesa estas solicitudes y envía los correos electrónicos a los usuarios, completando así el proceso de verificación de registro.

Diagrama de componentes:

Dentro de cada contenedor, el diagrama de componentes desglosa el sistema en componentes más pequeños y específicos que realizan funciones particulares. Esta perspectiva es esencial para el desarrollo y la asignación de responsabilidades a diferentes partes del sistema, asegurando que cada componente esté bien definido y que su función esté claramente entendida.



El diagrama de componentes del proyecto IdBird muestra la estructura interna del sistema, dividiendo el sistema en componentes específicos que realizan funciones particulares. A continuación se describe cada parte del diagrama y sus interacciones:

Aplicación Móvil (Kotlin, MVC):

1. Vista:

- P1 - Inicio de sesión: Pantalla donde el usuario ingresa sus credenciales para acceder a la aplicación.
- P2 - Registro de usuarios: Pantalla donde el usuario puede registrarse proporcionando la información requerida.
- P3 - Página principal: Pantalla principal que muestra las opciones disponibles para el usuario.
- P4 - Menú: Menú de navegación que permite al usuario acceder a diferentes secciones de la aplicación.
- P5 - Editar Perfil: Pantalla donde el usuario puede actualizar su información personal.
- P6 - Bitácora: Pantalla que muestra las bitácoras registradas por el usuario.
- P7 - Muestreo: Pantalla que muestra los muestreos realizados y permite agregar nuevos muestreos.

2. Controlador:

- Usuario: Gestiona las acciones relacionadas con el perfil del usuario.
- Bitácoras y Muestreos: Gestiona las acciones relacionadas con la creación, actualización y eliminación de bitácoras y muestreos.
- Email: Maneja el envío de correos electrónicos para confirmaciones y notificaciones.
- Identificación: Gestiona el proceso de identificación de aves a partir de imágenes.
- Aves Clasificadas: Maneja la información de las aves que han sido clasificadas.

3. Modelo:

- Usuarios: Gestiona los datos del usuario, como credenciales y perfil.
- Bitácoras de Campo: Gestiona la información de las bitácoras creadas por el usuario.
- Muestreo: Gestiona los datos relacionados con los muestreos realizados por el usuario.
- Base de Datos Local (ROOM): Almacena localmente la información de usuarios, bitácoras y muestreos.

API de Correos (NodeJS - Express):

1. Vista:

- CorreoAutentificación (ejs): Muestra datos relacionados con la confirmación de correo electrónico y autenticación de usuario.

2. Controlador:

- Autenticación (JavaScript): Gestiona la autenticación de usuarios, incluyendo registro e inicio de sesión.
- PostageAPI (JavaScript): Maneja el envío de correos electrónicos utilizando el servicio PostageApp.
- Usuarios (MongoDB): Gestiona la información de los usuarios en la base de datos global.

3. Modelo:

- Autenticación (JavaScript): Gestiona la lógica relacionada con la autenticación de usuarios.
- Usuarios (MongoDB): Almacena la información de los usuarios y las claves de confirmación.

PostageApp (Sistema de software):

1. Proveedores de servicios de correo electrónico: Maneja el envío de correos electrónicos utilizando JSON - HTTP para la comunicación.

Flujo de Interacción

1. Usuario y Aplicación Móvil:

- El usuario interactúa con las pantallas de la aplicación móvil para iniciar sesión, registrar usuarios, ver y editar bitácoras y muestreos, y actualizar su perfil.
- La vista de la aplicación maneja estas interacciones y envía las solicitudes correspondientes a los controladores.

2. Controladores de la Aplicación Móvil:

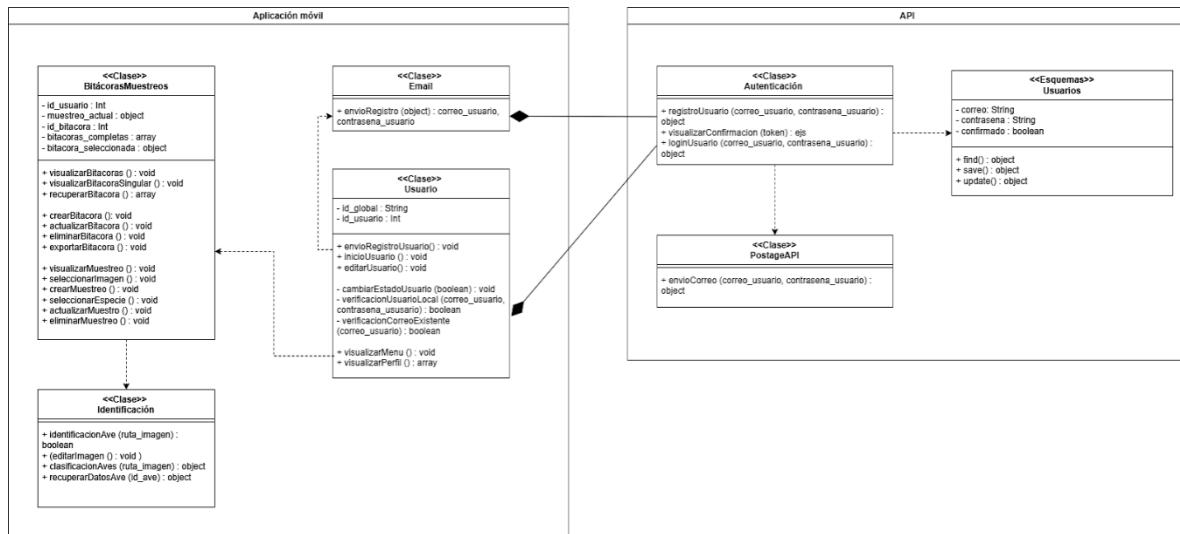
- Los controladores responden a las acciones del usuario y actualizan el modelo según sea necesario.
- Los controladores también interactúan con el API de correos para enviar correos de confirmación y notificaciones.

API de Correos y PostageApp:

- El API de correos gestiona las solicitudes de autenticación y el envío de correos electrónicos.
- El PostageApp recibe las solicitudes del API de correos y envía los correos electrónicos a los usuarios.

Diagrama de clases:

El diagrama de clases ofrece un nivel detallado, mostrando la estructura interna de los componentes en términos de clases y sus relaciones. Este detalle es fundamental para los desarrolladores, ya que proporciona un mapa claro de la implementación del código, facilitando el desarrollo y el mantenimiento del software.



El diagrama de arquitectura de clases ilustra la estructura y las interacciones entre las clases en la aplicación móvil y su API para la detección de aves. Aquí se proporciona una descripción resumida y detallada de las principales clases y sus relaciones.

Aplicación Móvil

La aplicación móvil está compuesta por varias clases que gestionan diferentes aspectos de la funcionalidad de la aplicación.

La clase BitácorasMuestreos es fundamental, ya que maneja la visualización, creación, actualización y eliminación de bitácoras y muestreos. Esta clase contiene atributos que incluyen el identificador del usuario, el muestreo actual, el identificador de la bitácora y arrays para almacenar bitácoras completas y la bitácora seleccionada. Sus métodos permiten al usuario visualizar todas las bitácoras, crear nuevas, actualizar las existentes, y exportarlas en formato PDF. Además, facilita la gestión de los muestreos al permitir seleccionar imágenes y especies, así como actualizar y eliminar muestreos.

La clase Usuario maneja la información y las acciones relacionadas con los usuarios. Posee atributos como el identificador global y el identificador del usuario, y métodos para registrar usuarios, iniciar sesión, editar información, verificar la existencia de correos y cambiar el estado del usuario. También incluye métodos para mostrar el menú y el perfil del usuario.

Para la gestión del correo electrónico, la clase Email envía registros de usuario, conectando con la API para completar el proceso de autenticación. En cuanto a la identificación de aves, la clase Identificación proporciona métodos para identificar aves a partir de imágenes, editar imágenes y clasificar aves, además de recuperar datos específicos de aves identificadas.

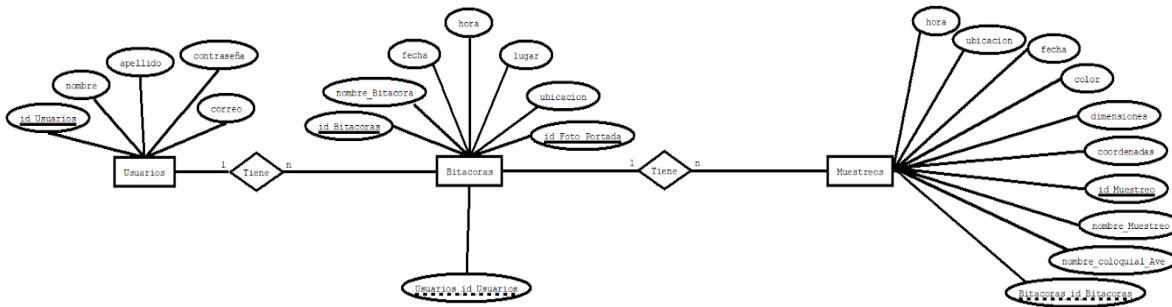
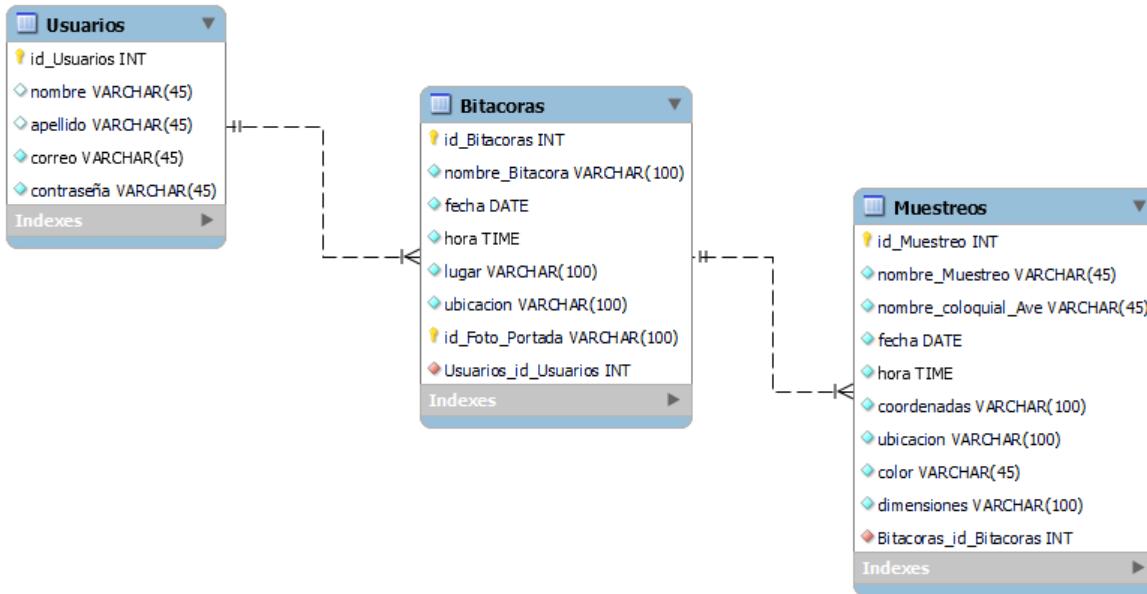
Finalmente, la clase Aves Clasificadas almacena información sobre las aves, incluyendo su identificador, nombre, color, dimensiones, nivel de endemismo y nivel de peligro de extinción.

API

La API se compone principalmente de las clases Autenticación y PostageAPI, junto con un esquema para los usuarios. La clase Autenticación gestiona el registro y el inicio de sesión de los usuarios, y muestra la confirmación del registro mediante un token. La clase PostageAPI es responsable del envío de correos electrónicos durante el proceso de registro y autenticación.

El Esquema de Usuarios define la estructura de los datos de usuario, incluyendo el correo, contraseña y el estado de confirmación. Los métodos en este esquema permiten encontrar, guardar y actualizar los datos de los usuarios.

Diseño de la base de datos.



El diagrama presentado es un modelo de base de datos relacional que estructura la información necesaria para una aplicación de detección de aves. Este modelo incluye tres tablas principales: Usuarios, Bitacoras y Muestreos, y define las relaciones entre ellas para asegurar la integridad y la eficiencia en la gestión de datos.

La tabla Usuarios contiene información básica sobre los usuarios de la aplicación, incluyendo su identificador único (id_Usuarios), nombre, apellido, correo electrónico y contraseña. Cada usuario puede crear múltiples bitácoras, lo que se refleja en la relación uno a muchos entre las tablas Usuarios y Bitacoras.

La tabla Bitacoras almacena registros de las observaciones de aves. Cada bitácora tiene un identificador único (id_Bitacoras) y está vinculada a un usuario específico mediante la clave foránea Usuarios_id_Usuarios. Además, la tabla Bitacoras incluye detalles como el nombre de la bitácora, la fecha y hora de la observación, el lugar y la ubicación específica de la misma, y un identificador para la foto de portada. La relación entre Usuarios y Bitacoras asegura que cada bitácora esté vinculada a un solo usuario, permitiendo que un usuario tenga múltiples bitácoras.

La tabla Muestreos guarda información detallada sobre cada muestreo individual realizado dentro de una bitácora. Cada muestreo tiene su propio identificador único (id_Muestreo) y está asociado con una bitácora específica a través de la clave foránea Bitacoras_id_Bitacoras. Los campos en la tabla Muestreos incluyen el nombre del muestreo, el nombre coloquial del ave identificada, la fecha y hora del muestreo, las coordenadas geográficas del lugar, la ubicación específica, el color y las dimensiones del ave observada. La relación entre Bitácoras y Muestreos garantiza que cada muestreo pertenezca a una sola bitácora, permitiendo que una bitácora contenga múltiples muestreos.

Manejo de archivos.



El diagrama representa la estructura de directorios y el manejo de archivos de la aplicación IdBird, que organiza las bitácoras de campo y sus respectivos contenidos de manera eficiente. A continuación, se detalla la estructura y la funcionalidad de cada uno de los directorios:

Estructura de Directorios

1. Directorio principal: IdBird

Este es el directorio raíz que contiene toda la información relacionada con la aplicación IdBird.

2. Subdirectorio: Bitácoras de campo

Dentro de este subdirectorio se encuentran todas las bitácoras creadas por los usuarios.

3. Sub-subdirectorio: Bitácoras

Ruta: ./IdBird/Bitácoras de Campo/Bitácoras/

Contiene todas las bitácoras creadas por los usuarios, organizadas de manera que se pueda acceder fácilmente a cada registro.

4. Sub-subdirectorio: Muestreos

Ruta: ./IdBird/Bitacoras de Campo/Bitacoras_N/Muestreos/

Este directorio almacena las imágenes subidas o tomadas por el usuario para un muestreo específico. Cada bitácora puede tener múltiples muestras asociados, y este directorio ayuda a mantener organizadas las imágenes relacionadas con cada uno.

5. Sub-subdirectorio: Bitácora PDF

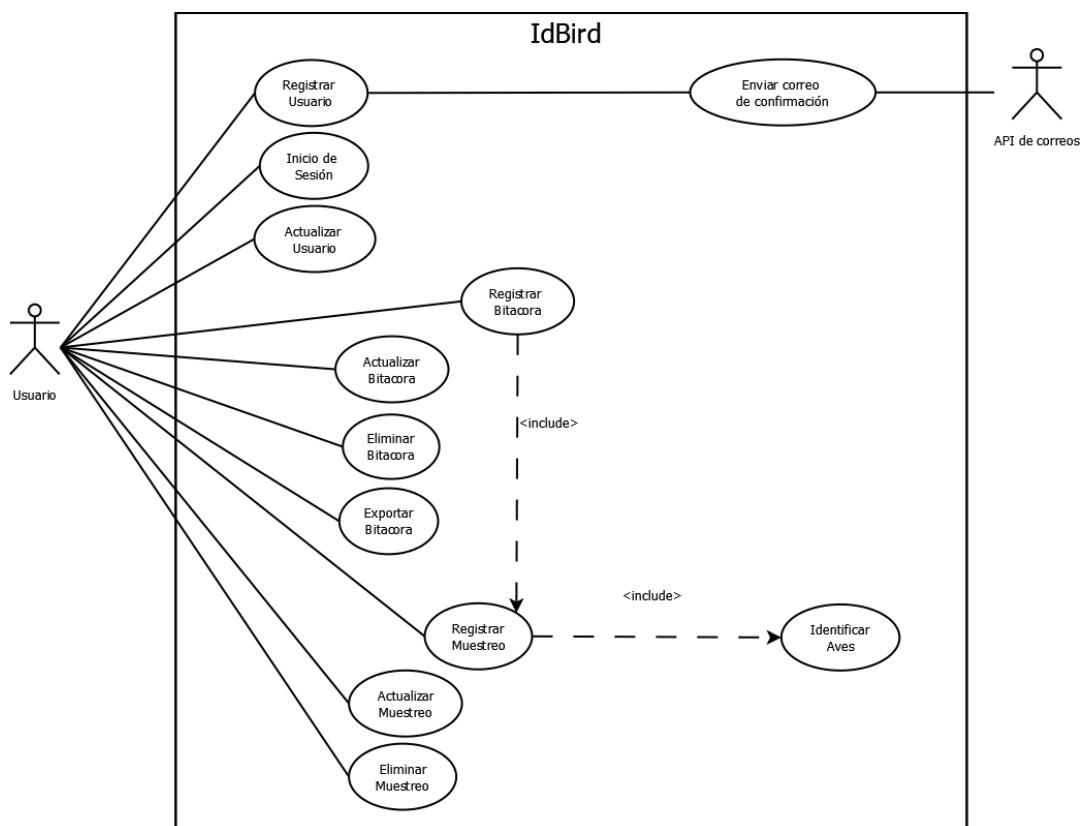
Ruta: ./IdBird/Bitacoras de Campo/Bitacoras_N/Bitacora PDF/

Almacena los archivos PDF de las bitácoras generados por la aplicación. Estos PDFs son versiones exportables de las bitácoras, facilitando su revisión y compartición. Almacenar los PDFs en un directorio separado asegura que los documentos sean fáciles de localizar y gestionar.

Diagramas UML

Diagramas UML de comportamiento

Diagrama de caso de uso:



El diagrama presentado es un diagrama de casos de uso que ilustra las interacciones entre el usuario y la aplicación móvil IdBird, así como la integración con el servicio PostageApp para el envío de correos de confirmación. Este diagrama es útil para comprender las funcionalidades clave de la aplicación desde la perspectiva del usuario.

Descripción del Diagrama de Casos de Uso

Actores:

- Usuario: Representado por un ícono de persona, el usuario interactúa con la aplicación móvil IdBird para realizar diversas tareas relacionadas con el registro, gestión de bitácoras y muestreos, e identificación de aves.
- PostageApp: Representado por otro ícono de persona, PostageApp es el sistema que se encarga de enviar correos electrónicos de confirmación. Este actor interactúa con la aplicación IdBird para completar el proceso de registro del usuario.

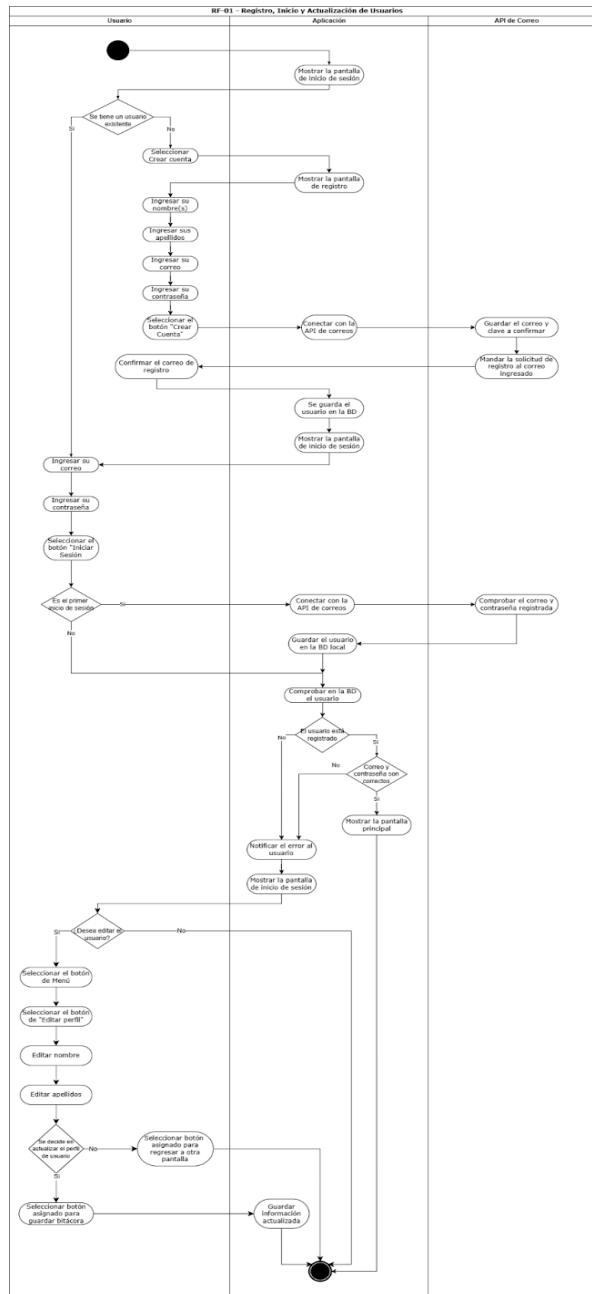
Casos de Uso:

- Registrar Usuario: El usuario se registra en la aplicación IdBird. Este caso de uso incluye el envío de un correo de confirmación a través de PostageApp.
- Inicio de Sesión: El usuario inicia sesión en la aplicación.
- Actualizar Usuario: El usuario actualiza su información personal en la aplicación.
- Registrar Bitácora: El usuario crea una nueva bitácora para registrar observaciones de aves. Este caso de uso incluye la posibilidad de registrar muestreos y, eventualmente, identificar aves.
- Actualizar Bitácora: El usuario actualiza la información de una bitácora existente.
- Eliminar Bitácora: El usuario elimina una bitácora existente.
- Exportar Bitácora: El usuario exporta una bitácora en formato PDF.

- Registrar Muestreo: El usuario registra un nuevo muestreo dentro de una bitácora. Este caso de uso incluye la identificación de aves, lo que permite al usuario registrar detalles específicos sobre las aves observadas.
- Actualizar Muestreo: El usuario actualiza la información de un muestreo existente.
- Eliminar Muestreo: El usuario elimina un muestreo existente.
- Identificar Aves: El usuario utiliza la funcionalidad de la aplicación para identificar aves a partir de imágenes y otros datos.

Diagramas de actividades:

RF-01 Registro y Actualización de Usuarios



El diagrama de actividad presentado describe el proceso de registro, inicio de sesión y actualización de usuarios (RF-01) en la aplicación móvil IdBird, detallando las interacciones entre el usuario, la aplicación y la API de correos.

Descripción del Diagrama de Actividad

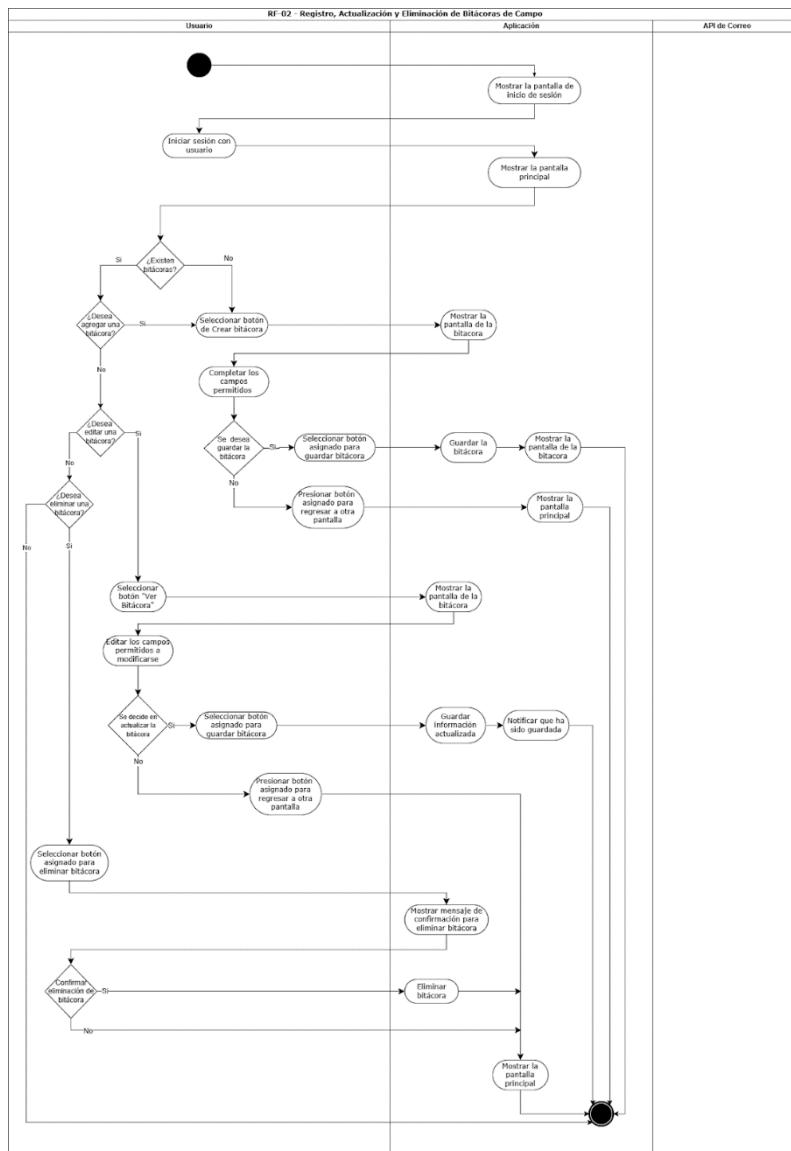
1. Inicio del Proceso:

- El proceso comienza con el usuario accediendo a la pantalla de inicio de sesión de la aplicación.

2. Verificación de Usuario Existente:
 - La aplicación verifica si el usuario ya existe. Si el usuario no existe, se muestra la pantalla de registro.
 - El usuario ingresa su nombre, apellidos, correo y contraseña, y selecciona el botón "Crear Cuenta".
3. Conexión con la API de Correos:
 - La aplicación se conecta con la API de correos para guardar el correo y la clave a confirmar.
 - La API de correos envía una solicitud de registro al correo ingresado por el usuario.
 - El usuario debe confirmar su correo de registro.
4. Guardar Usuario en la Base de Datos:
 - Una vez confirmado el correo, el usuario se guarda en la base de datos y se muestra nuevamente la pantalla de inicio de sesión.
5. Inicio de Sesión:
 - El usuario ingresa su correo y contraseña, y selecciona el botón "Iniciar Sesión".
 - Si es el primer inicio de sesión, la aplicación se conecta con la API de correos para comprobar el correo y contraseña registrada.
 - El usuario se guarda en la base de datos local.
6. Verificación de Usuario y Credenciales:
 - La aplicación comprueba en la base de datos si el usuario está registrado.
 - Si el usuario no está registrado o las credenciales son incorrectas, se notifica el error al usuario y se muestra nuevamente la pantalla de inicio de sesión.
 - Si las credenciales son correctas, se muestra la pantalla principal de la aplicación.
7. Actualización de Usuario:
 - Desde la pantalla principal, el usuario puede optar por actualizar su perfil seleccionando el botón de Menú y luego el botón de "Editar perfil".
 - El usuario edita su nombre y apellidos, y decide si desea actualizar su perfil.
 - Si se decide actualizar, el usuario selecciona el botón asignado para guardar la información actualizada.
8. Guardar Información Actualizada:

- La información actualizada del usuario se guarda en la base de datos, completando así el proceso de actualización.

RF-02 Registro, Actualización y Eliminación de Bitácoras de Campo



El diagrama de actividad presentado describe el proceso de registro, actualización y eliminación de bitácoras de campo (RF-02) en la aplicación móvil IdBird. Este diagrama detalla los pasos que siguen los usuarios para gestionar sus bitácoras de campo y las interacciones entre el usuario y la aplicación.

Descripción del Diagrama de Actividad

1. Inicio del Proceso:

- El proceso comienza con el usuario iniciando sesión en la aplicación.

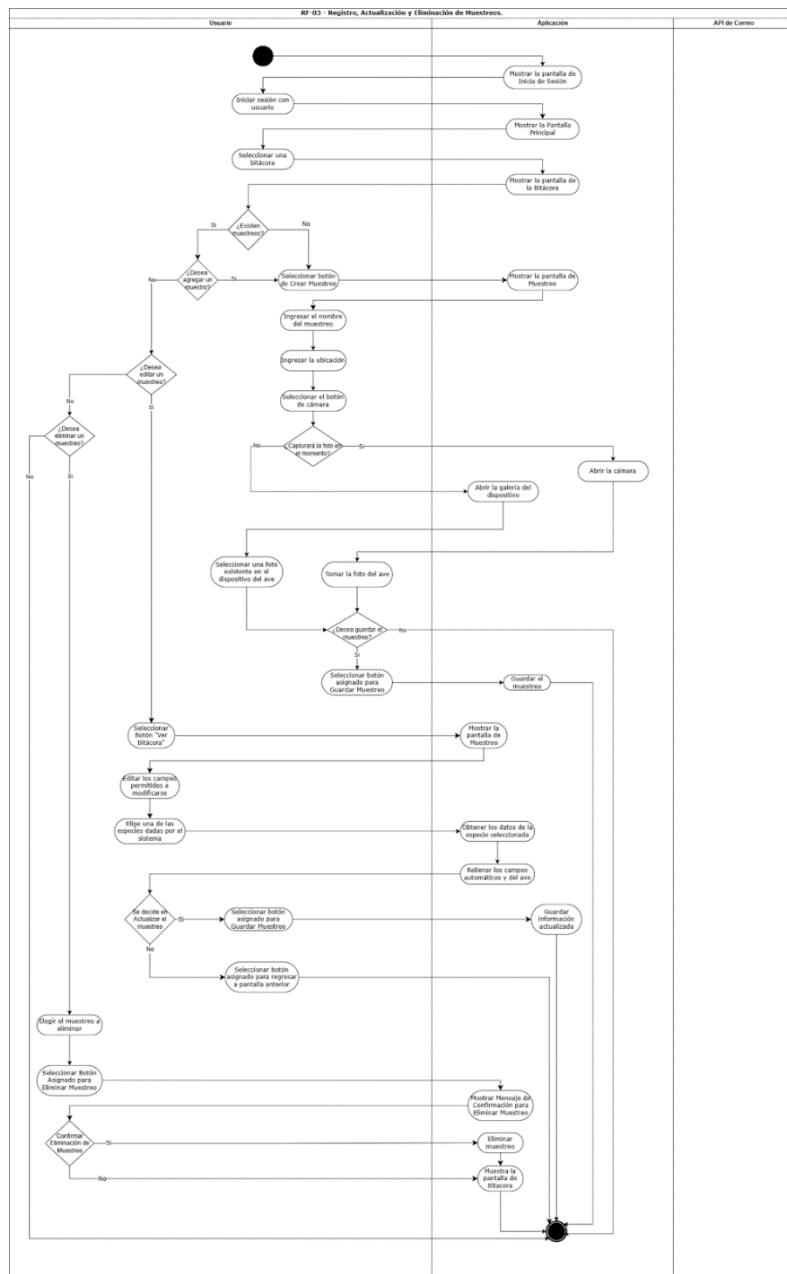
2. Verificación de Bitácoras Existentes:
 - Una vez que el usuario ha iniciado sesión, la aplicación muestra la pantalla principal.
 - Se verifica si el usuario tiene bitácoras existentes.

3. Registro de Bitácora:
 - Si el usuario desea agregar una nueva bitácora, selecciona el botón de "Crear Bitácora".
 - La aplicación muestra la pantalla de la bitácora, donde el usuario completa los campos permitidos.
 - Si el usuario decide guardar la bitácora, selecciona el botón asignado para guardar.
 - La bitácora se guarda y la aplicación muestra nuevamente la pantalla de la bitácora y luego la pantalla principal.

4. Actualización de Bitácora:
 - Si el usuario desea editar una bitácora existente, selecciona el botón de "Ver Bitácora".
 - La aplicación muestra la pantalla de la bitácora, donde el usuario puede editar los campos permitidos.
 - Si el usuario decide actualizar la bitácora, selecciona el botón asignado para guardar.
 - La información actualizada de la bitácora se guarda y se notifica al usuario que ha sido guardada.

5. Eliminación de Bitácora:
 - Si el usuario desea eliminar una bitácora existente, selecciona el botón asignado para eliminar.
 - La aplicación muestra un mensaje de confirmación para la eliminación de la bitácora.
 - Si el usuario confirma la eliminación, la bitácora se elimina y se muestra nuevamente la pantalla principal.

RF-03 Registro, Actualización y Eliminación de Muestreos



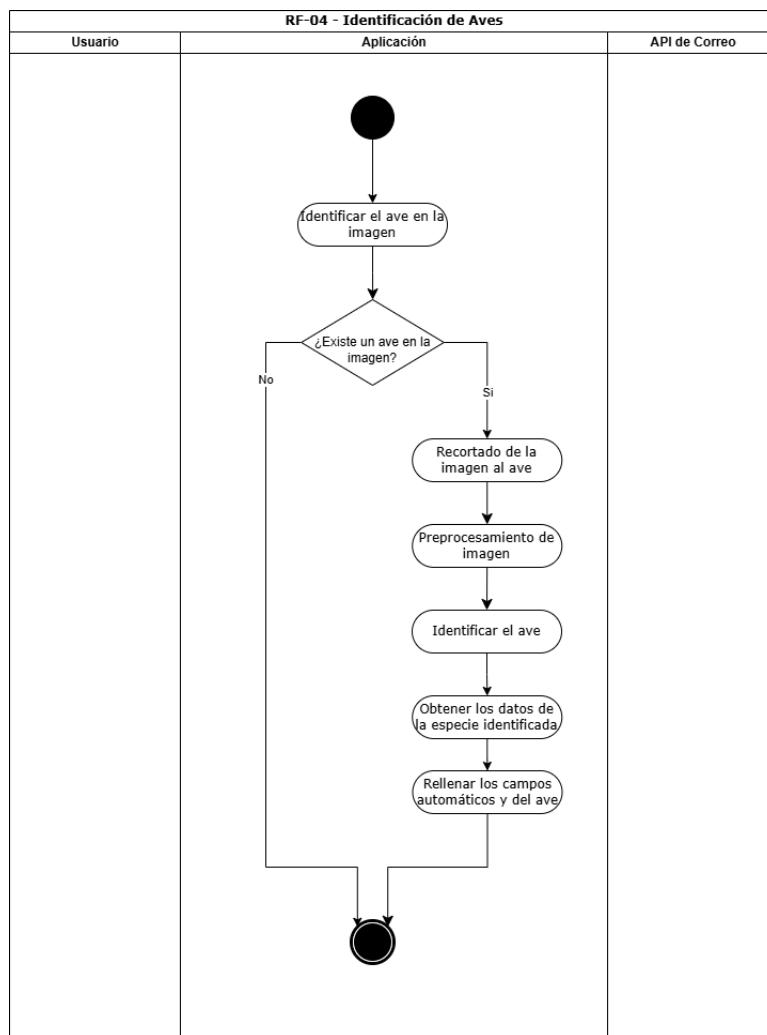
El diagrama de actividad presentado describe el proceso de registro, actualización y eliminación de muestras (RF-03) en la aplicación móvil IdBird. Este diagrama detalla los pasos que siguen los usuarios para gestionar los muestras asociados a las bitácoras de campo y las interacciones entre el usuario y la aplicación.

Descripción del Diagrama de Actividad

- 1. Inicio del Proceso:**
 - El proceso comienza con el usuario iniciando sesión en la aplicación.
- 2. Selección de Bitácora:**
 - Después de iniciar sesión, el usuario selecciona una bitácora desde la pantalla principal.
 - La aplicación muestra la pantalla de la bitácora seleccionada.
- 3. Verificación de Muestreos Existentes:**
 - La aplicación verifica si existen muestreos asociados a la bitácora seleccionada.
- 4. Registro de Muestreo:**
 - Si el usuario desea agregar un nuevo muestreo, selecciona el botón de "Crear Muestreo".
 - La aplicación muestra la pantalla de muestreo, donde el usuario ingresa el nombre del muestreo y la ubicación.
 - El usuario selecciona el botón de cámara para capturar una foto del ave. Puede elegir entre abrir la cámara o seleccionar una foto existente de la galería del dispositivo.
 - Si el usuario toma una nueva foto, puede optar por guardarla o repetir la toma.
 - Una vez satisfecho con la foto, el usuario selecciona el botón para guardar el muestreo.
 - La aplicación guarda el muestreo y muestra nuevamente la pantalla de muestreo, luego la pantalla de la bitácora.
- 5. Actualización de Muestreo:**
 - Si el usuario desea editar un muestreo existente, selecciona el botón de "Ver Bitácora" y luego el muestreo que desea editar.
 - La aplicación muestra la pantalla del muestreo, donde el usuario puede editar los campos permitidos.
 - El usuario elige una de las especies dadas por el sistema, y la aplicación obtiene los datos de la especie seleccionada, llenando automáticamente los campos del ave.
 - Si el usuario decide actualizar el muestreo, selecciona el botón para guardar los cambios.
 - La información actualizada se guarda y se notifica al usuario que ha sido guardada.
- 6. Eliminación de Muestreo:**

- Si el usuario desea eliminar un muestreo existente, selecciona el muestreo a eliminar y luego el botón de "Eliminar Muestreo".
- La aplicación muestra un mensaje de confirmación para la eliminación del muestreo.
- Si el usuario confirma la eliminación, el muestreo se elimina y la aplicación muestra nuevamente la pantalla de la bitácora.

RF-04 Identificación de Aves



El diagrama de actividad presentado describe el proceso de identificación de aves (RF-04) en la aplicación móvil IdBird. Este diagrama detalla los pasos que sigue la aplicación para identificar un ave a partir de una imagen proporcionada por el usuario.

Descripción del Diagrama de Actividad

1. Inicio del Proceso:

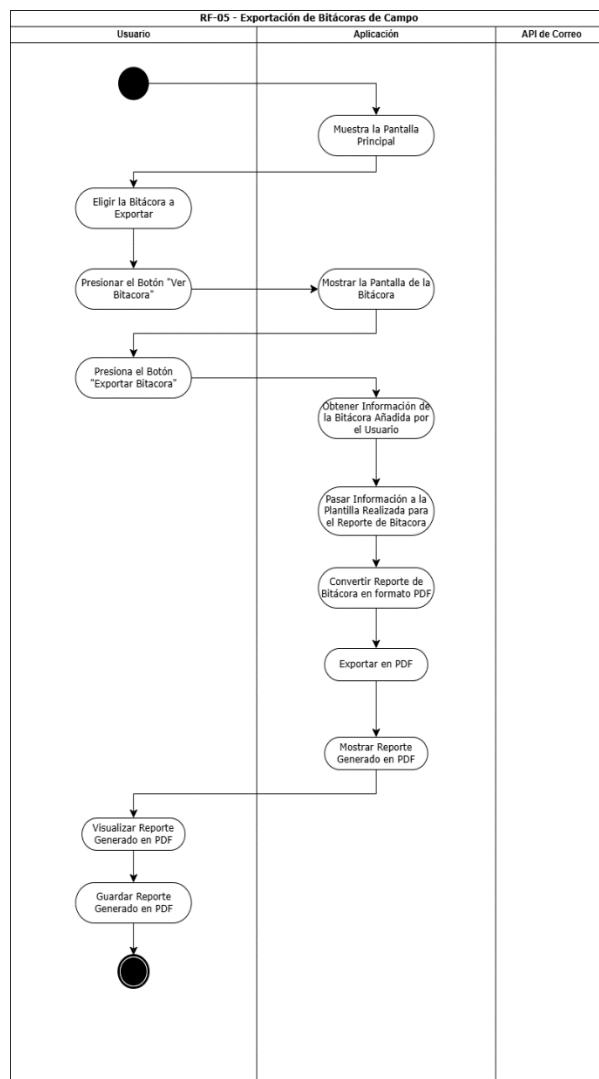
- El proceso comienza cuando el usuario inicia la función de identificación de aves en la aplicación.

2. Identificar el Ave en la Imagen:

- El usuario sube una imagen del ave que desea identificar.
- La aplicación pregunta si existe un ave en la imagen.

3. Verificación de Presencia de Ave:
 - Si no se detecta un ave en la imagen, el proceso termina en este punto.
 - Si se detecta un ave en la imagen, la aplicación procede con los siguientes pasos.
4. Recortado de la Imagen al Ave:
 - La aplicación recorta la imagen para enfocar al ave, eliminando cualquier fondo innecesario.
5. Preprocesamiento de Imagen:
 - La aplicación realiza un preprocesamiento de la imagen para mejorar la calidad y asegurar que los detalles relevantes del ave sean claros para el sistema de identificación.
6. Identificación del Ave:
 - La aplicación utiliza técnicas de reconocimiento de imágenes para identificar el ave en la imagen.
7. Obtener los Datos de la Especie Identificada:
 - Una vez identificada el ave, la aplicación obtiene datos específicos sobre la especie, como el nombre científico, características, y cualquier otra información relevante.
8. Rellenar los Campos Automáticos y del Ave:
 - La aplicación rellena automáticamente los campos con los datos obtenidos de la especie identificada.

RF-05 Exportación de Bitácoras de Campo



El diagrama de actividad presentado describe el proceso de exportación de bitácoras de campo (RF-05) en la aplicación móvil IdBird. Este diagrama detalla los pasos que sigue el usuario y la aplicación para convertir una bitácora en un archivo PDF y permitir su descarga.

Descripción del Diagrama de Actividad

1. Inicio del Proceso:

- El proceso comienza cuando el usuario accede a la aplicación móvil IdBird.

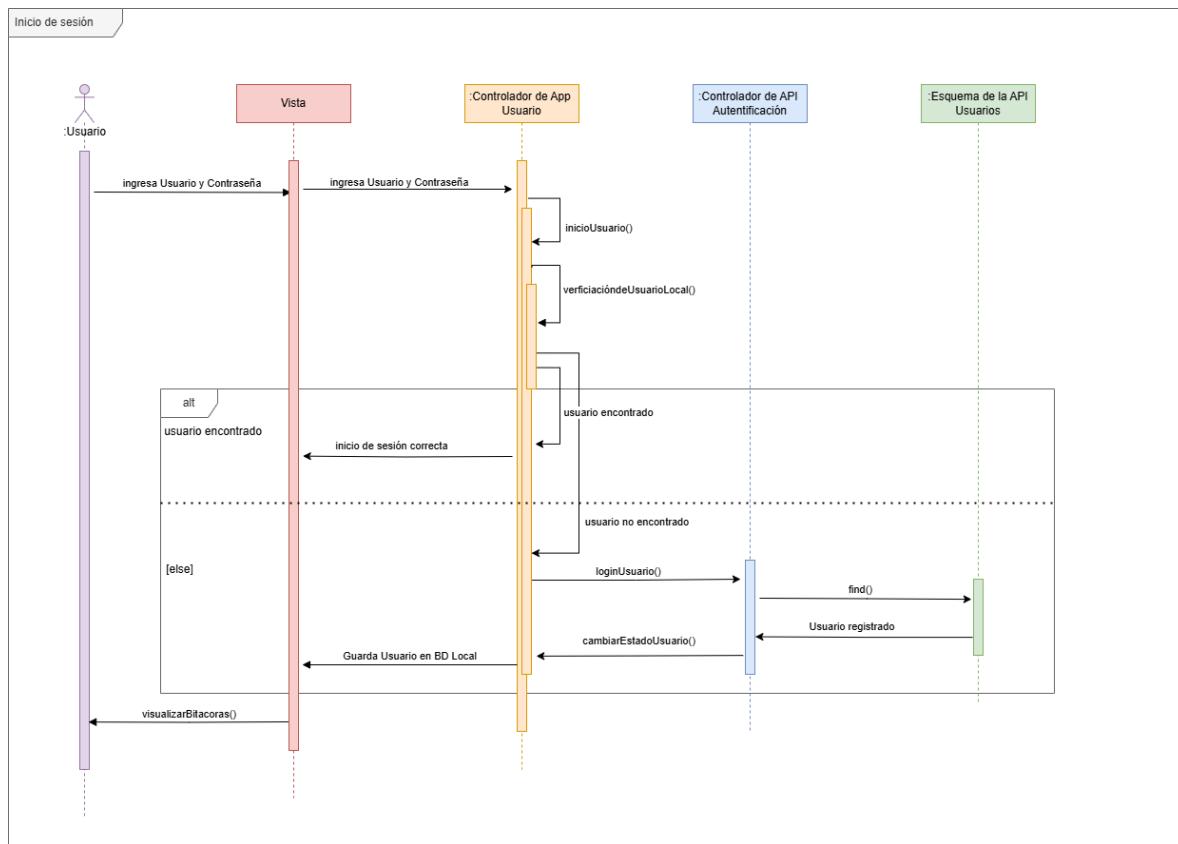
2. Selección de Bitácora:

- El usuario elige la bitácora que desea exportar desde la pantalla principal.
- El usuario presiona el botón "Ver Bitácora" para visualizar los detalles de la bitácora seleccionada.

3. Exportación de Bitácora:
 - El usuario presiona el botón "Exportar Bitácora" para iniciar el proceso de exportación.
 - La aplicación obtiene la información de la bitácora añadida por el usuario.
4. Generación del Reporte en PDF:
 - La aplicación pasa la información de la bitácora a una plantilla predefinida para generar el reporte de la bitácora.
 - El reporte de la bitácora se convierte en formato PDF.
 - La aplicación exporta el reporte en formato PDF.
5. Visualización y Guardado del PDF:
 - La aplicación muestra el reporte generado en formato PDF.
 - El usuario puede visualizar el reporte generado y guardararlo en su dispositivo.

Diagrama de secuencia:

DS-01 Inicio de Sesión



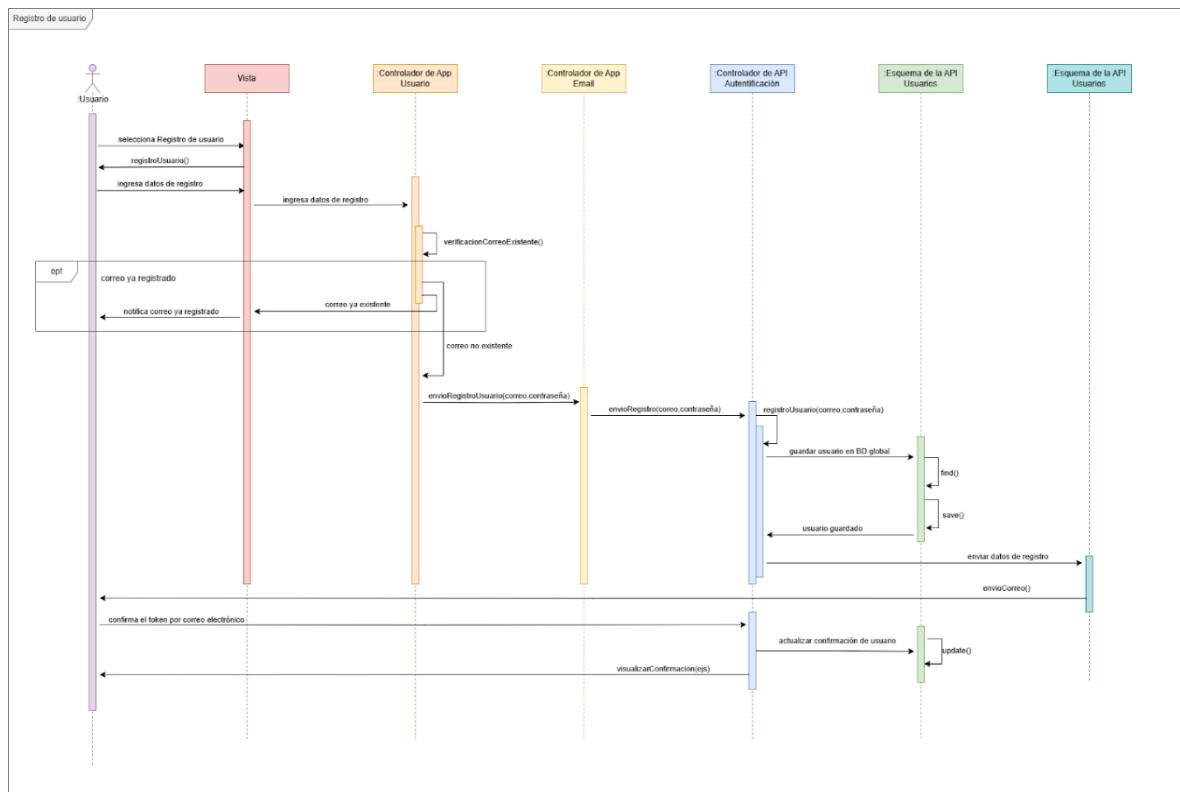
El diagrama de secuencia presentado describe el proceso de inicio de sesión (DS01) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación, el controlador de la aplicación, el controlador de la API de autenticación y el esquema de la API de usuarios durante el proceso de inicio de sesión.

Descripción del Diagrama de Secuencia

1. Ingreso de Credenciales por el Usuario.
 - El usuario ingresa su correo y contraseña en la vista de la aplicación.
 - La vista de la aplicación envía las credenciales al controlador de la aplicación (Usuario).
2. Verificación Local del Usuario:

- El controlador de la aplicación llama al método inicioUsuario() para iniciar el proceso de verificación.
 - Luego, el controlador de la aplicación utiliza el método verificacionUsuarioLocal() para verificar si el usuario está registrado localmente.
3. Confirmación de Usuario Encontrado Localmente:
- Si el usuario es encontrado localmente, se confirma la verificación.
 - El controlador de la aplicación responde con un mensaje de "inicio de sesión correcta" y llama al método visualizarBitacoras() para mostrar las bitácoras del usuario.
 - La vista de la aplicación muestra la pantalla principal.
4. Verificación Externa del Usuario:
- Si el usuario no es encontrado localmente, el controlador de la aplicación llama al método loginUsuario() en el controlador de la API de autenticación.
 - El controlador de la API de autenticación utiliza el método find() del esquema de la API de usuarios para buscar el usuario en la base de datos global.
5. Confirmación de Usuario Encontrado Externamente:
- Si el usuario es encontrado en la base de datos global, se confirma la verificación.
 - El controlador de la API de autenticación responde con el estado del usuario y llama al método cambiarEstadoUsuario() para actualizar el estado del usuario.
 - El controlador de la aplicación responde con un mensaje de "inicio de sesión correcta" y llama al método visualizarBitacoras() para mostrar las bitácoras del usuario.
 - La vista de la aplicación muestra la pantalla principal.

DS-02 Registro de usuario



El diagrama de secuencia presentado describe el proceso de registro de usuario (DS-02) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación, el controlador de la aplicación, el controlador de la API de autenticación, el esquema de la API de usuarios y el controlador de la API de correos durante el proceso de registro de un nuevo usuario.

Descripción del Diagrama de Secuencia

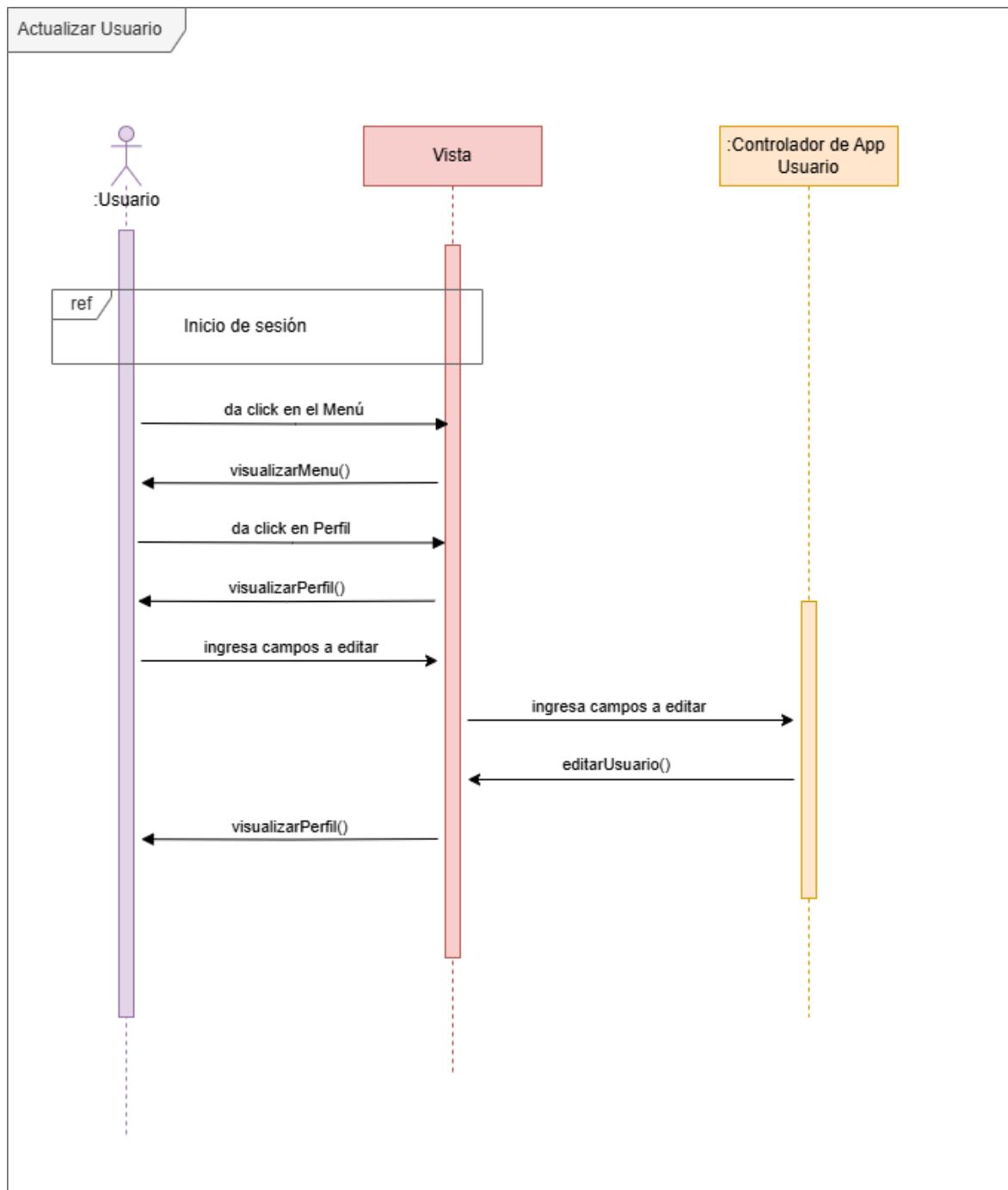
1. Selección de Registro de Usuario:
 - El usuario selecciona la opción de "Registro de usuario" en la vista de la aplicación.
 - La vista de la aplicación muestra la pantalla de registro.

2. Ingreso de Datos de Registro:
 - El usuario ingresa los datos de registro, como nombre, correo y contraseña.
 - La vista de la aplicación envía estos datos al controlador de la aplicación (Usuario).

3. Verificación de Correo Existente:
 - El controlador de la aplicación verifica si el correo ya está registrado usando el método `verificacionCorreoExistente()`.

- Si el correo ya está registrado, la aplicación notifica al usuario y el proceso termina aquí.
 - Si el correo no está registrado, el controlador de la aplicación envía los datos de registro al controlador de la aplicación de correo (Email).
4. Envío de Registro a la API de Autenticación:
- El controlador de la aplicación de correo envía los datos de registro (correo y contraseña) a la API de autenticación.
 - La API de autenticación llama al método registroUsuario(correo, contraseña) para registrar al usuario en la base de datos global.
5. Guardar Usuario en la Base de Datos Global:
- La API de autenticación utiliza el esquema de la API de usuarios para guardar el nuevo usuario en la base de datos global.
 - El esquema de la API de usuarios llama al método save() para almacenar los datos del usuario.
 - Una vez guardado el usuario, la API de autenticación envía los datos de registro al controlador de la API de correos.
6. Envío de Correo de Confirmación:
- El controlador de la API de correos envía un correo de confirmación al nuevo usuario utilizando el método envioCorreo().
 - El usuario recibe un correo con un token de confirmación y confirma su registro.
7. Confirmación de Registro:
- El usuario confirma el token recibido por correo electrónico.
 - La aplicación visualiza la página de confirmación de correo.
 - La API de autenticación actualiza la confirmación del usuario en la base de datos global utilizando el método update().

DS-03 Actualizar Usuario



El diagrama de secuencia presentado describe el proceso de actualización de usuario (DS-03) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación y el controlador de la aplicación durante el proceso de actualización de la información del perfil del usuario.

[Descripción del Diagrama de Secuencia](#)

1. Inicio de Sesión:

 - El diagrama asume que el usuario ya ha iniciado sesión en la aplicación.
2. Navegación al Menú de Perfil:

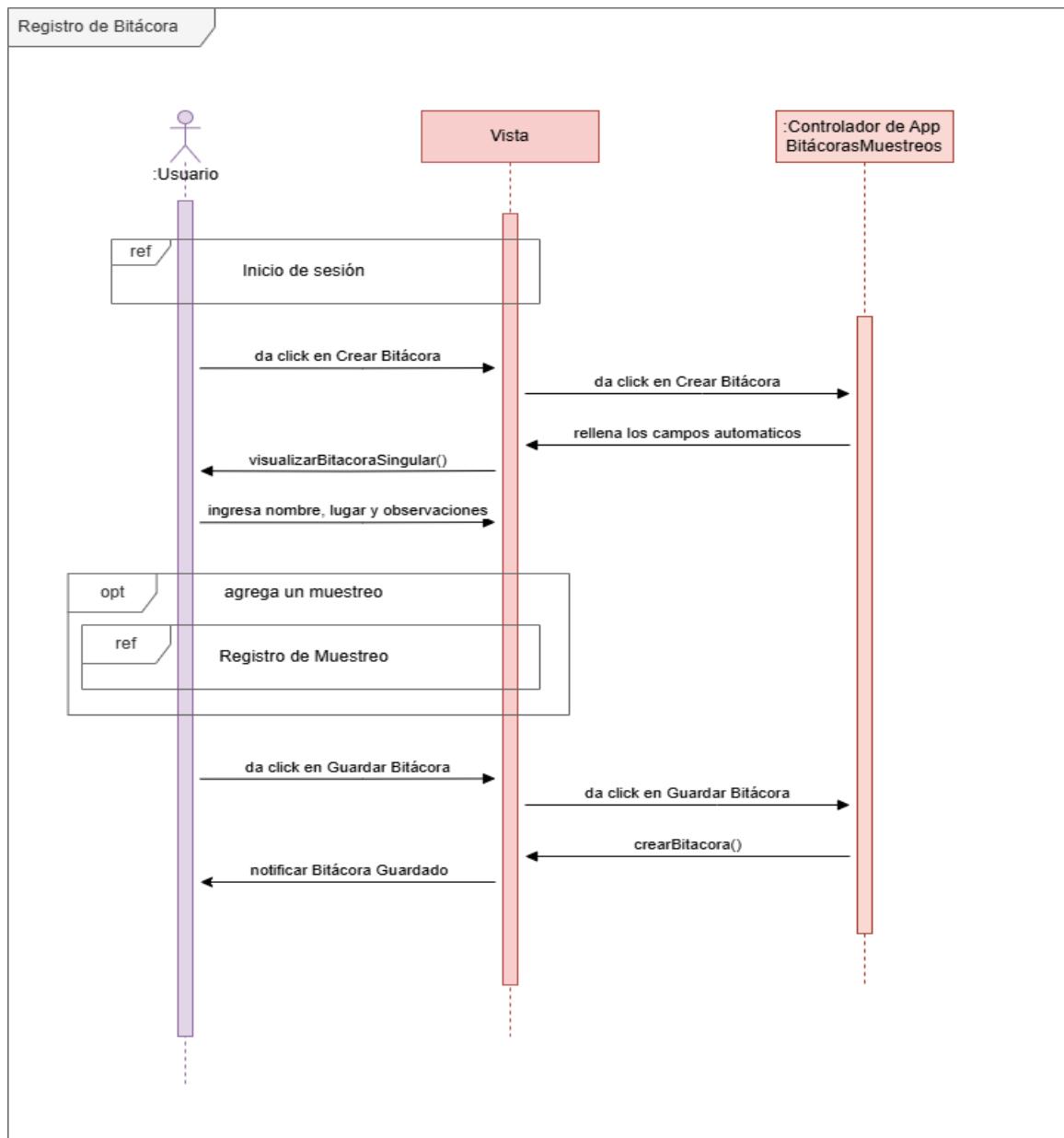
 - El usuario da clic en el menú de la aplicación.
 - La vista de la aplicación muestra el menú.
 - El usuario da clic en la opción de perfil dentro del menú.
 - La vista de la aplicación muestra la pantalla del perfil del usuario.
3. Acceso a la Pantalla de Editar Perfil:

 - El usuario selecciona la opción para editar el perfil.
 - La vista de la aplicación muestra la pantalla de editar perfil.
4. Ingreso de Datos para Editar:

 - El usuario ingresa los campos que desea editar en su perfil.
 - La vista de la aplicación envía estos datos al controlador de la aplicación (Usuario).
5. Actualización del Perfil del Usuario:

 - El controlador de la aplicación recibe los datos y llama al método editarUsuario() para actualizar la información del usuario en la base de datos.
 - Una vez actualizada la información, la vista de la aplicación muestra nuevamente la pantalla de editar perfil con los datos actualizados.

DS-04 Registro de Bitácora



El diagrama de secuencia presentado describe el proceso de registro de una bitácora (DS-04) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación y el controlador de la aplicación durante el proceso de creación de una nueva bitácora.

Descripción del Diagrama de Secuencia

1. Inicio de Sesión:

- El diagrama asume que el usuario ya ha iniciado sesión en la aplicación.

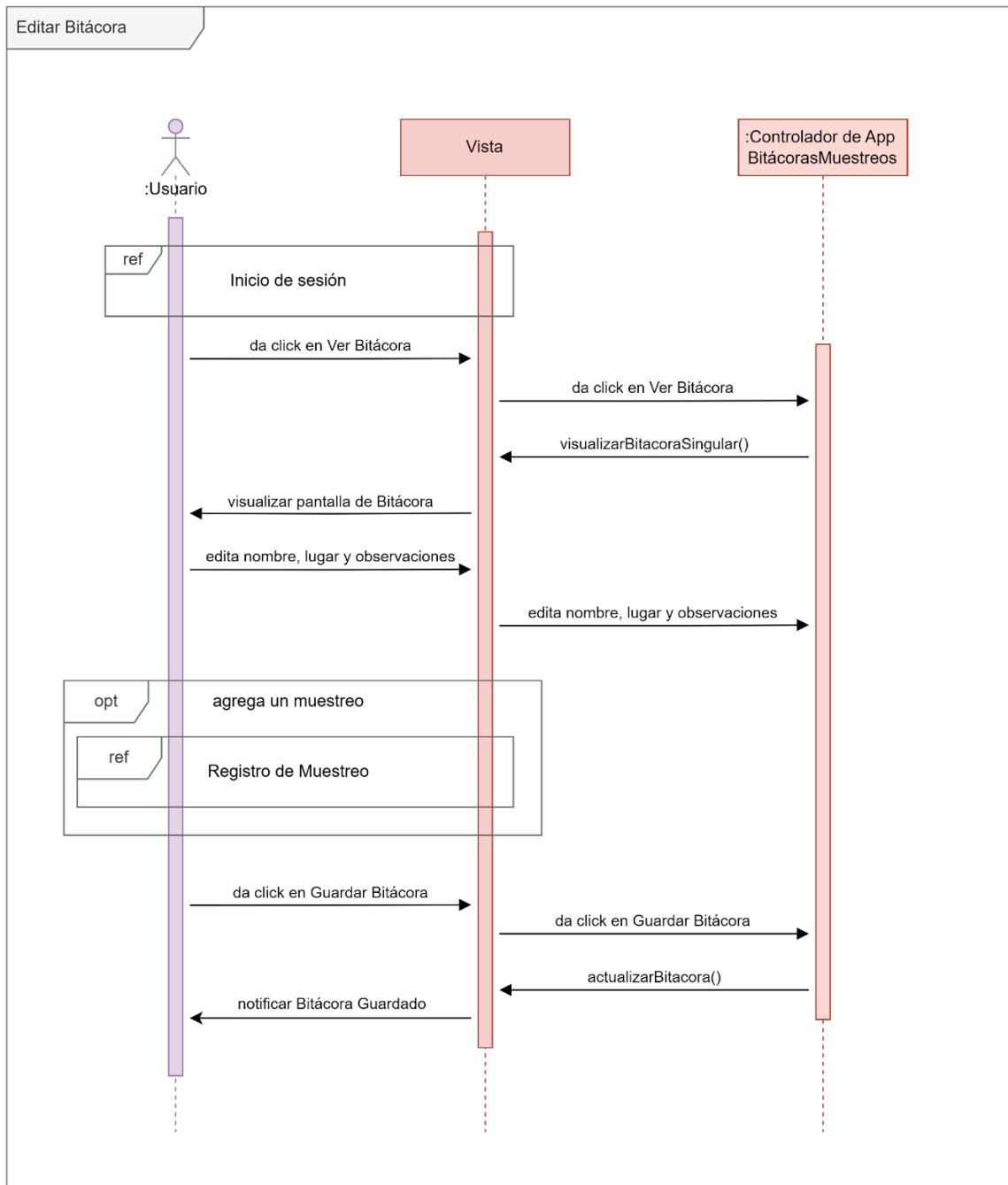
2. Navegación a Crear Bitácora:
 - El usuario da clic en la opción "Crear Bitácora".
 - La vista de la aplicación recibe la acción y solicita al controlador de la aplicación (BitácorasMuestreos) que inicialice el proceso de creación de una nueva bitácora.
 - La vista muestra la pantalla de creación de bitácora, con campos para ingresar ubicación, fecha y cantidad.

3. Ingreso de Datos de la Bitácora:
 - El usuario ingresa el nombre, lugar y observaciones para la bitácora.
 - La vista de la aplicación envía estos datos al controlador de la aplicación.

4. Creación de la Bitácora:
 - El controlador de la aplicación recibe los datos y llama al método crearBitacora() para guardar la nueva bitácora en la base de datos.
 - Una vez creada la bitácora, el controlador de la aplicación notifica a la vista que la bitácora ha sido guardada exitosamente.

5. Opción de Agregar Muestreo:
 - El usuario tiene la opción de agregar un muestreo a la bitácora recién creada.
 - Si el usuario elige agregar un muestreo, se sigue el proceso de registro de muestreo (referenciado como "Registro de Muestreo" en el diagrama).

DS-05 Editar Bitácora

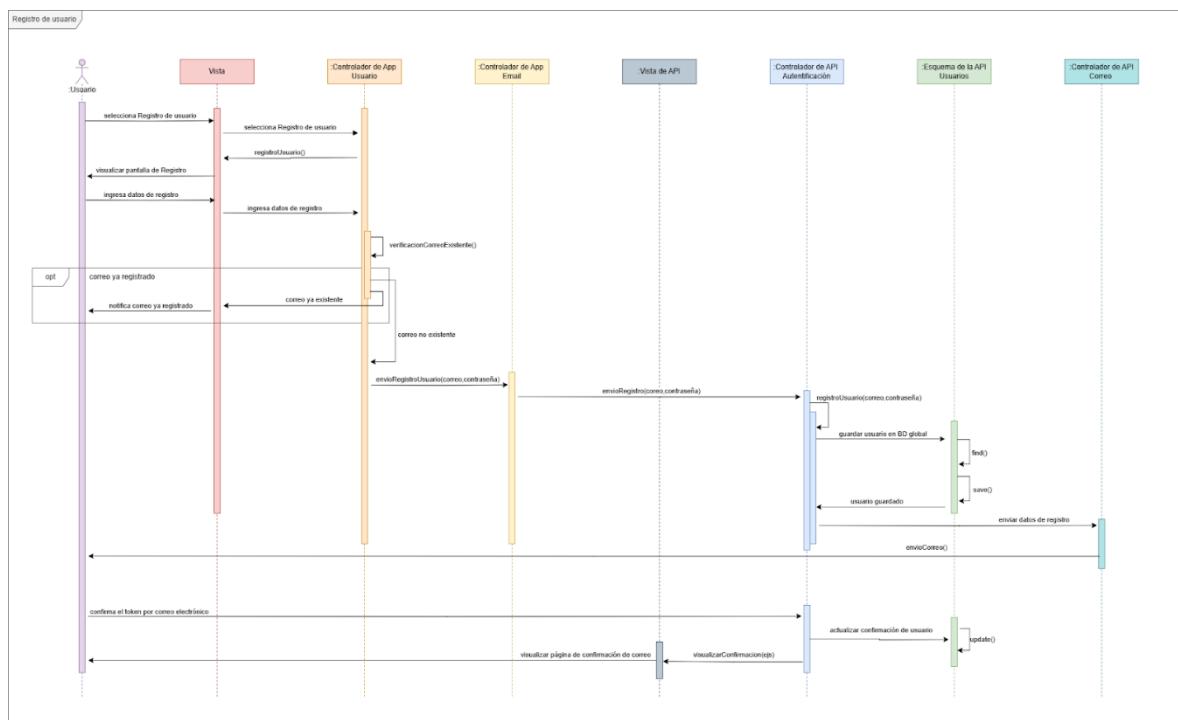


El diagrama de secuencia presentado describe el proceso de edición de una bitácora (DS-05) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación y el controlador de la aplicación durante el proceso de modificación de una bitácora existente.

Descripción del Diagrama de Secuencia

- 1. Inicio de Sesión:**
 - El diagrama asume que el usuario ya ha iniciado sesión en la aplicación.
- 2. Navegación a Ver Bitácora:**
 - El usuario da clic en la opción "Ver Bitácora".
 - La vista de la aplicación recibe la acción y solicita al controlador de la aplicación (BitácorasMuestreos) que muestre la bitácora seleccionada.
 - La vista muestra la pantalla de la bitácora seleccionada.
- 3. Edición de Datos de la Bitácora:**
 - El usuario edita el nombre, lugar y observaciones de la bitácora.
 - La vista de la aplicación envía estos datos editados al controlador de la aplicación.
- 4. Opción de Agregar Muestreo:**
 - El usuario tiene la opción de agregar un muestreo a la bitácora.
 - Si el usuario elige agregar un muestreo, se sigue el proceso de registro de muestreo (referenciado como "Registro de Muestreo" en el diagrama).
- 5. Guardado de la Bitácora:**
 - El usuario da clic en el botón "Guardar Bitácora" para confirmar los cambios.
 - La vista de la aplicación envía la solicitud de guardado al controlador de la aplicación.
 - El controlador de la aplicación recibe los datos editados y llama al método crearBitacora() para guardar los cambios en la base de datos.
 - Una vez guardada la bitácora, el controlador de la aplicación notifica a la vista que la bitácora ha sido guardada exitosamente.

DS-06 Eliminar Bitácora



El diagrama de secuencia presentado describe el proceso de eliminación de una bitácora (DS-06) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación y el controlador de la aplicación durante el proceso de eliminación de una bitácora existente.

Descripción del Diagrama de Secuencia

1. Inicio de Sesión:
 - El diagrama asume que el usuario ya ha iniciado sesión en la aplicación.
2. Navegación a Eliminar Bitácora:
 - El usuario da clic en la opción "Eliminar Bitácora".
 - La vista de la aplicación recibe la acción y notifica al usuario con una confirmación de eliminación.
3. Confirmación de Eliminación:
 - El usuario tiene dos opciones: cancelar la eliminación o confirmar la eliminación.

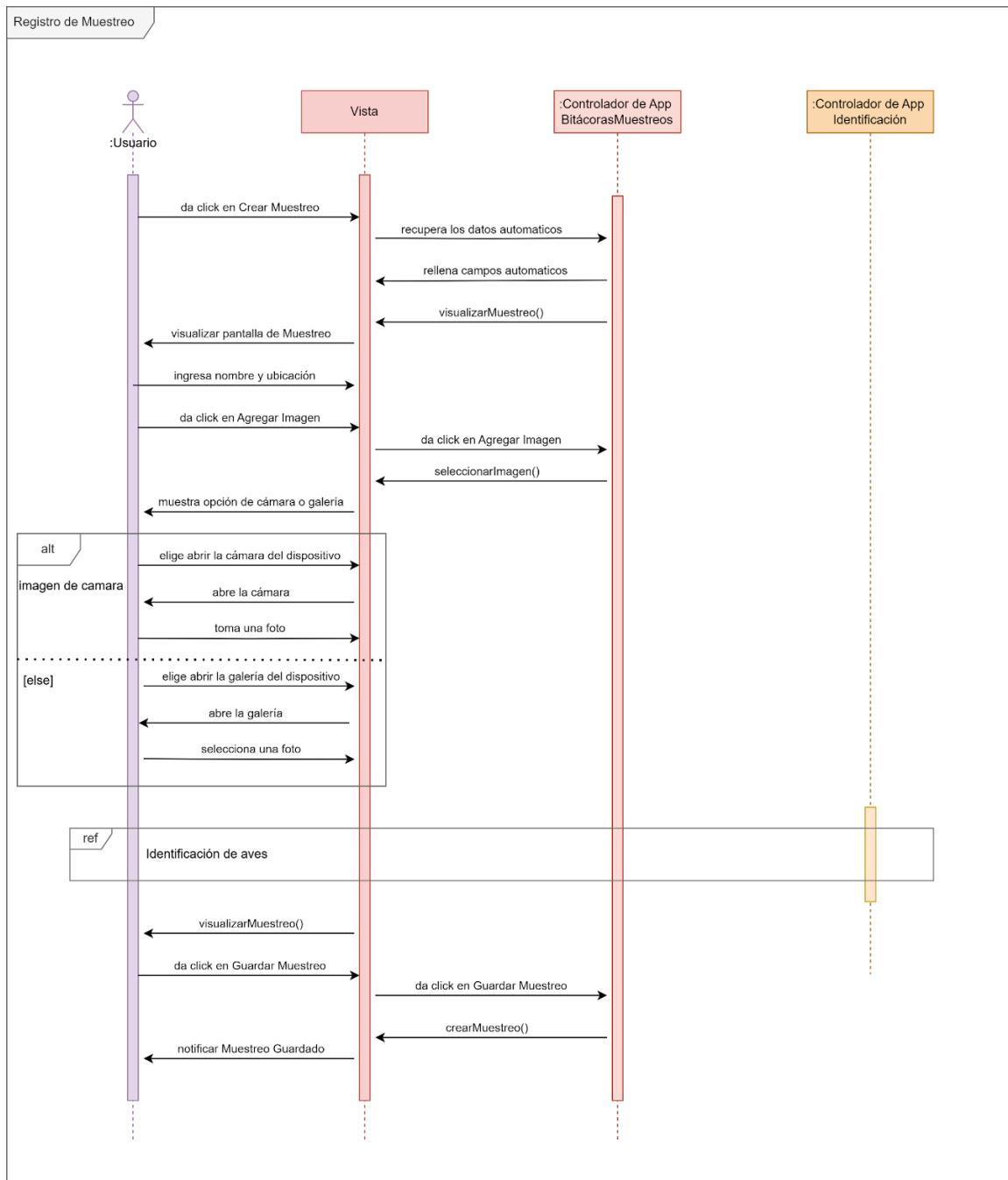
4. Cancelación de Eliminación:

- Si el usuario decide cancelar la eliminación, da clic en el botón "Cancelar".
- La vista de la aplicación cancela el proceso y muestra nuevamente las bitácoras disponibles utilizando el método visualizarBitacoras().

5. Confirmación de Eliminación:

- Si el usuario decide confirmar la eliminación, da clic en el botón "Confirmar".
- La vista de la aplicación envía esta acción al controlador de la aplicación (BitácorasMuestreos).
- El controlador de la aplicación llama al método eliminarBitacora() para eliminar la bitácora seleccionada de la base de datos.
- Una vez eliminada la bitácora, la vista de la aplicación muestra nuevamente las bitácoras disponibles.

DS-07 Registro de Muestreo

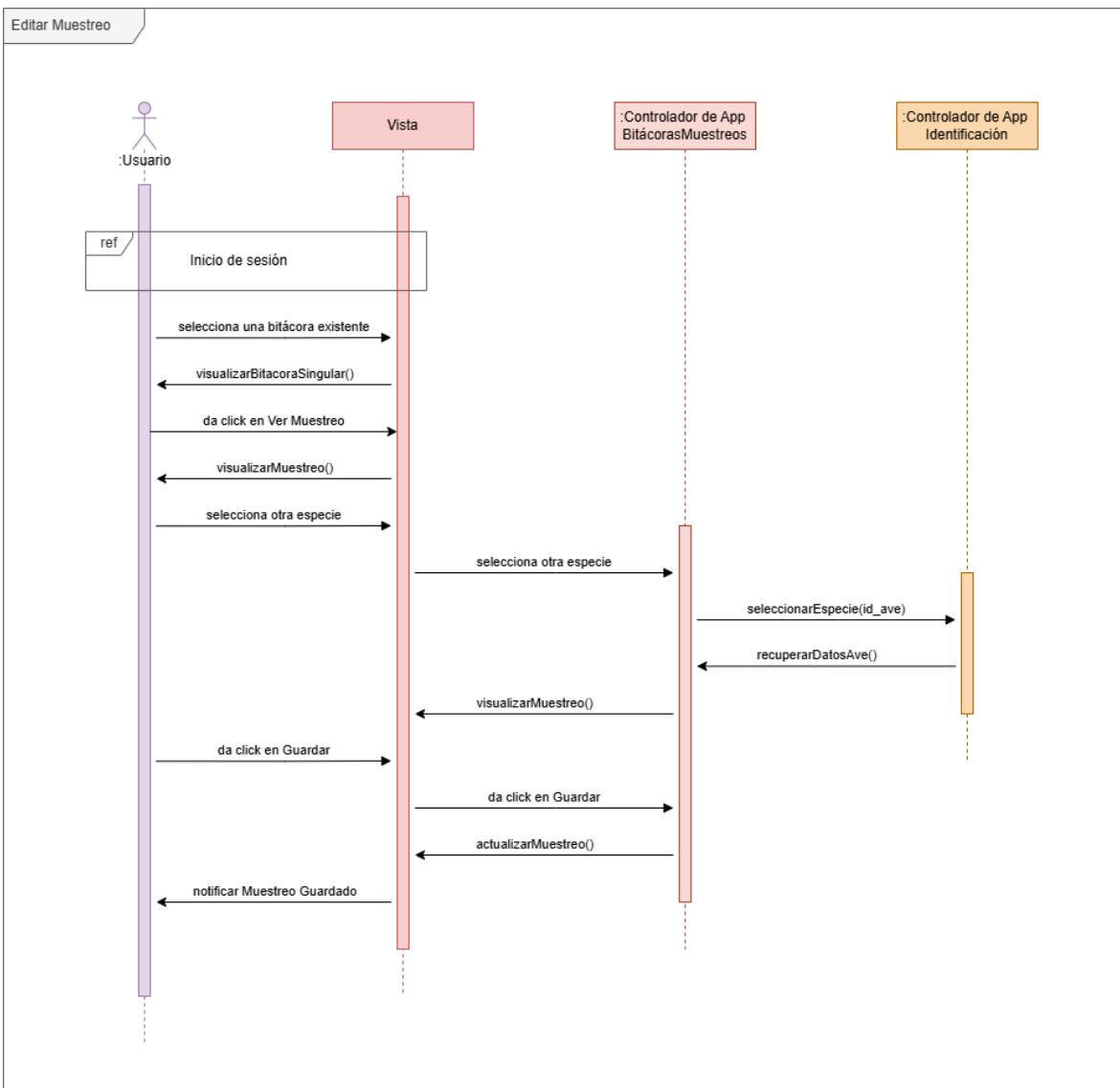


El diagrama de secuencia presentado describe el proceso de registro de un muestreo (DS-07) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación y los controladores de la aplicación durante el proceso de creación de un nuevo muestreo y la identificación de aves.

Descripción del Diagrama de Secuencia

1. Inicio del Proceso:
 - El diagrama asume que el usuario ya ha iniciado sesión en la aplicación y está en la vista de una bitácora específica.
2. Creación de Muestreo:
 - El usuario da clic en la opción "Crear Muestreo".
 - La vista de la aplicación recibe la acción y solicita al controlador de la aplicación (BitácorasMuestreos) que recupere los datos automáticos necesarios para el muestreo.
 - La vista de la aplicación rellena los campos automáticos y muestra la pantalla de muestreo.
3. Agregar Imagen:
 - El usuario da clic en "Agregar Imagen" para añadir una foto del ave.
 - La vista de la aplicación solicita al controlador de la aplicación (BitácorasMuestreos) que inicie el proceso de selección de imagen.
 - El controlador de la aplicación gestiona la selección de la imagen y la vista muestra la pantalla de agregar imagen.
4. Identificación de Aves (Referencia):
 - El proceso de identificación de aves se inicia como una referencia dentro del registro de muestreo. Esto incluye el flujo completo de identificación de aves, desde la selección de imagen hasta la obtención de datos de la especie identificada.
5. Finalización del Muestreo:
 - Una vez que el usuario ha agregado la imagen y se ha realizado la identificación de aves, la vista de la aplicación muestra nuevamente la pantalla de muestreo con los datos actualizados.
 - El usuario puede regresar a la vista de la bitácora singular, completando el proceso de registro de muestreo.

DS-08 Editar Muestreo



El diagrama de secuencia presentado describe el proceso de edición de un muestreo (DS-08) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación y los controladores de la aplicación durante el proceso de modificación de un muestreo existente.

Descripción del Diagrama de Secuencia

1. Inicio de Sesión:
 - El diagrama asume que el usuario ya ha iniciado sesión en la aplicación.
2. Selección de Bitácora y Muestreo:
 - El usuario selecciona una bitácora existente desde la vista principal.

- La vista de la aplicación muestra la bitácora seleccionada utilizando el método visualizarBitacoraSingular().
- El usuario da clic en "Ver Muestreo" para seleccionar el muestreo que desea editar.
- La vista de la aplicación muestra el muestreo utilizando el método visualizarMuestreo().

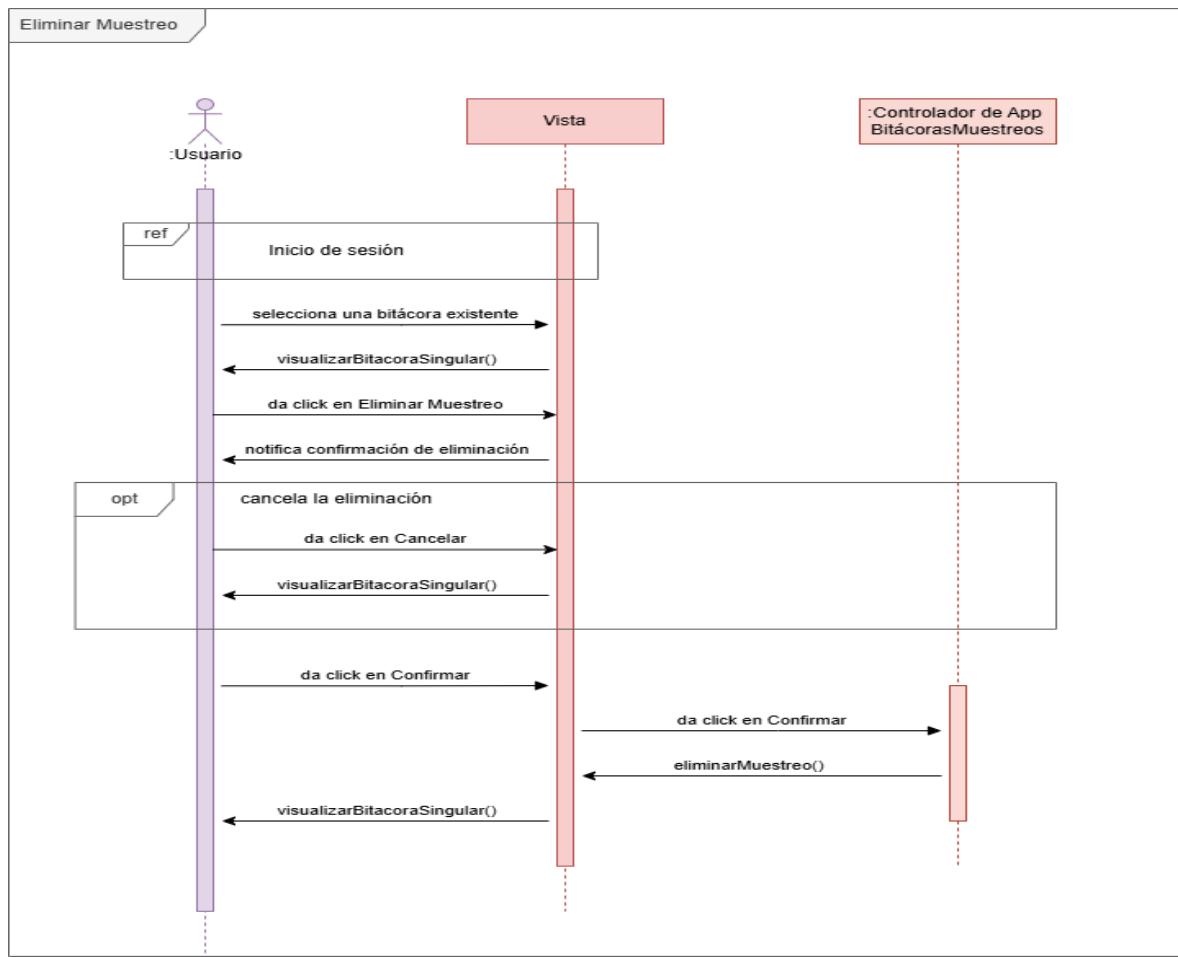
3. Edición de la Especie del Muestreo:

- El usuario selecciona otra especie para el muestreo.
- La vista de la aplicación envía esta acción al controlador de la aplicación (BitácorasMuestreos).
- El controlador de la aplicación llama al método seleccionarEspecie(id_ave) del controlador de la aplicación de identificación.
- El controlador de identificación recupera los datos de la nueva especie seleccionada utilizando el método recuperaDatosAve().

4. Actualización del Muestreo:

- La vista de la aplicación muestra nuevamente el muestreo con la nueva especie seleccionada utilizando el método visualizarMuestreo().
- El usuario da clic en "Guardar" para confirmar los cambios
- La vista de la aplicación envía esta acción al controlador de la aplicación.
- El controlador de la aplicación llama al método actualizarMuestreo() para guardar los cambios en la base de datos.
- Una vez guardado el muestreo, el controlador de la aplicación notifica a la vista que el muestreo ha sido guardado exitosamente.

DS-09 Eliminar Muestreo



El diagrama de secuencia presentado describe el proceso de eliminación de un muestreo (DS-09) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación y el controlador de la aplicación durante el proceso de eliminación de un muestreo existente.

Descripción del Diagrama de Secuencia

1. Inicio de Sesión:

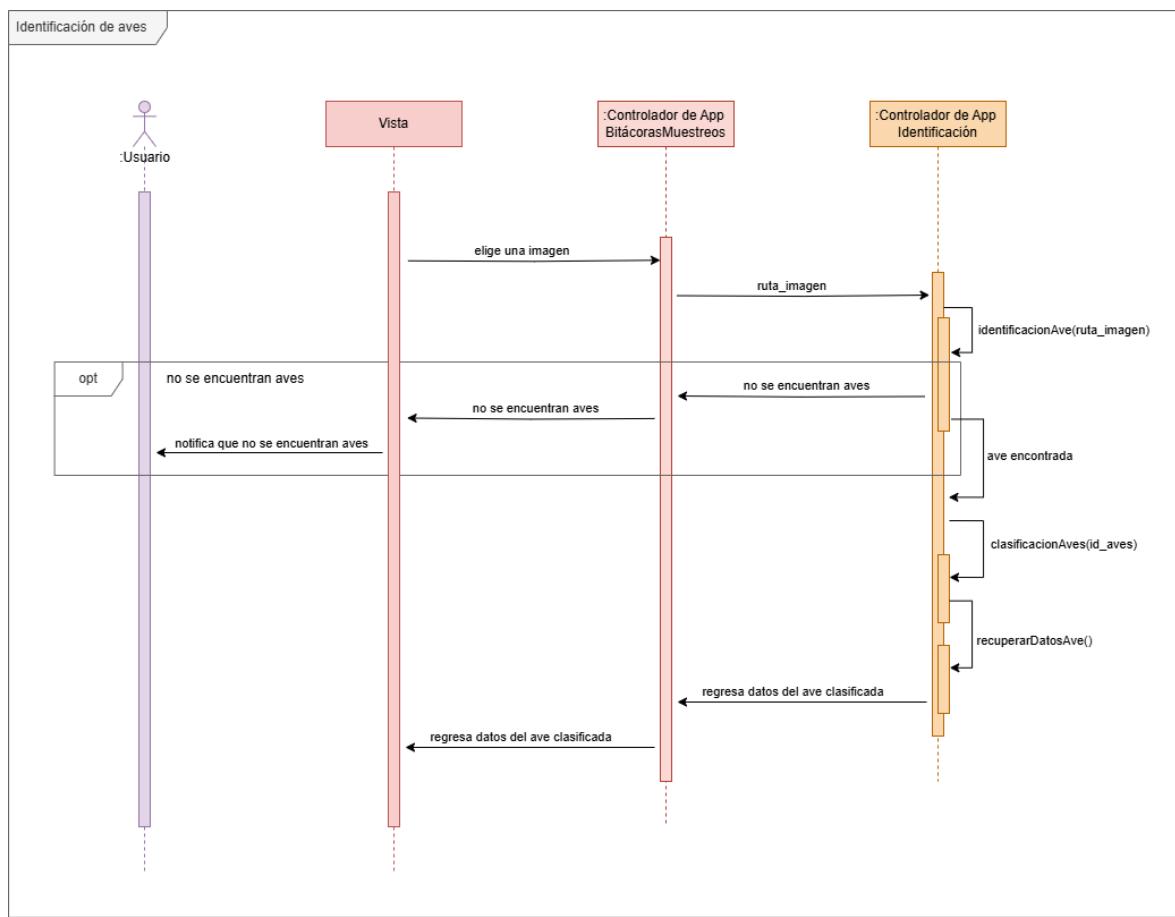
- El diagrama asume que el usuario ya ha iniciado sesión en la aplicación.

2. Selección de Bitácora y Muestreo:

- El usuario selecciona una bitácora existente desde la vista principal.
- La vista de la aplicación muestra la bitácora seleccionada utilizando el método `visualizarBitacoraSingular()`.

- El usuario da clic en "Eliminar Muestreo" para seleccionar el muestreo que desea eliminar.
 - La vista de la aplicación notifica al usuario con una confirmación de eliminación.
3. Confirmación o Cancelación de Eliminación:
- El usuario tiene dos opciones: cancelar la eliminación o confirmar la eliminación.
4. Cancelación de Eliminación:
- Si el usuario decide cancelar la eliminación, da clic en el botón "Cancelar".
 - La vista de la aplicación cancela el proceso y muestra nuevamente la bitácora utilizando el método visualizarBitacoraSingular().
5. Confirmación de Eliminación:
- Si el usuario decide confirmar la eliminación, da clic en el botón "Confirmar".
 - La vista de la aplicación envía esta acción al controlador de la aplicación (BitácorasMuestreos).
 - El controlador de la aplicación llama al método eliminarMuestreo() para eliminar el muestreo seleccionado de la base de datos.
 - Una vez eliminado el muestreo, la vista de la aplicación muestra nuevamente la bitácora utilizando el método visualizarBitacoraSingular().

DS-10 Identificación de Aves



El diagrama de secuencia presentado describe el proceso de identificación de aves (DS-10) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación y los controladores de la aplicación durante el proceso de identificación de un ave a partir de una imagen proporcionada por el usuario.

Descripción del Diagrama de Secuencia

1. Selección de Imagen:

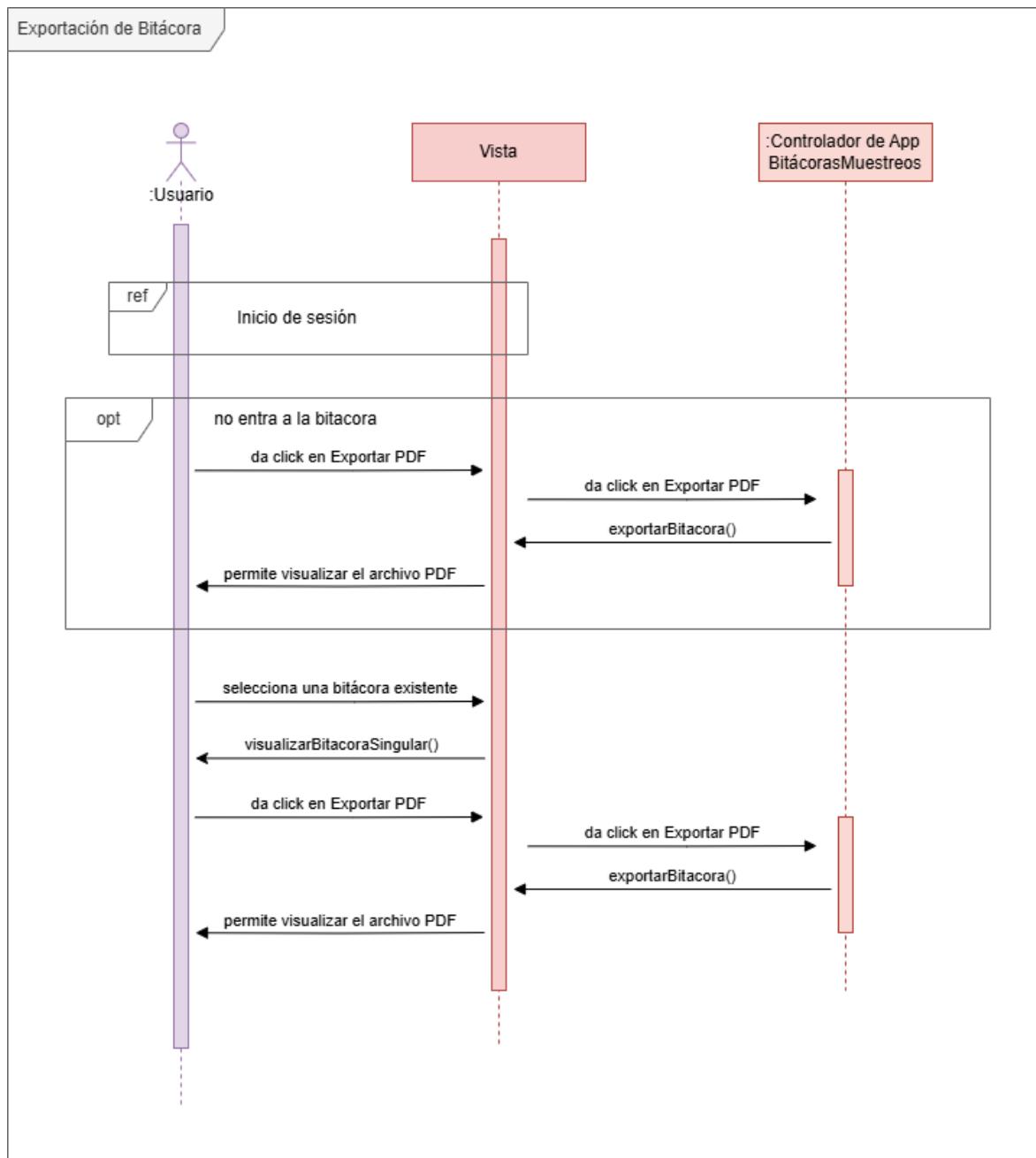
- El usuario elige una imagen desde la vista de la aplicación.
- La vista de la aplicación recibe la imagen seleccionada y envía la ruta de la imagen al controlador de la aplicación (BitácorasMuestreos).

2. Proceso de Identificación de Aves:

- El controlador de la aplicación (BitácorasMuestreos) envía la ruta de la imagen al controlador de la aplicación de identificación.

- El controlador de la aplicación de identificación utiliza el método identificacionAve(ruta_imagen) para procesar la imagen y buscar aves en ella.
3. Resultado de la Identificación:
- Si no se encuentran aves en la imagen, el controlador de la aplicación de identificación notifica al controlador de la aplicación (BitácorasMuestreos) que no se encontraron aves.
 - El controlador de la aplicación (BitácorasMuestreos) notifica a la vista que no se encontraron aves, y la vista informa al usuario.
4. Clasificación y Recuperación de Datos del Ave:
- Si se encuentran aves en la imagen, el controlador de la aplicación de identificación utiliza el método clasificacionAves(id_aves) para clasificar las aves encontradas.
 - Luego, el controlador de la aplicación de identificación utiliza el método recuperarDatosAve() para obtener los datos de las aves clasificadas.
5. Retorno de Datos:
- Los datos del ave clasificada se regresan al controlador de la aplicación (BitácorasMuestreos).
 - El controlador de la aplicación (BitácorasMuestreos) envía los datos a la vista de la aplicación.
 - La vista de la aplicación muestra los datos del ave clasificada al usuario.

DS-11 Exportación de Bitácora



El diagrama de secuencia presentado describe el proceso de exportación de una bitácora (DS-11) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación y el controlador de la aplicación durante el proceso de generación y descarga de un archivo PDF de una bitácora.

Descripción del Diagrama de Secuencia

1. Inicio de Sesión:
 - El diagrama asume que el usuario ya ha iniciado sesión en la aplicación.
2. Opción de Exportar PDF sin Entrar a la Bitácora:
 - El usuario puede optar por exportar una bitácora directamente sin entrar a verla.
 - El usuario da clic en "Exportar PDF" desde la vista principal.
 - La vista de la aplicación envía esta solicitud al controlador de la aplicación (BitácorasMuestreos).
 - El controlador de la aplicación llama al método exportarBitacora() para generar el archivo PDF de la bitácora.
 - Una vez generado el archivo PDF, la vista de la aplicación permite al usuario descargar el archivo.
3. Opción de Exportar PDF desde una Bitácora Existente:
 - Alternativamente, el usuario puede seleccionar una bitácora existente para verla antes de exportarla.
 - El usuario selecciona una bitácora existente desde la vista principal.
 - La vista de la aplicación muestra la bitácora seleccionada utilizando el método visualizarBitacoraSingular().
 - El usuario da clic en "Exportar PDF" desde la vista de la bitácora.
 - La vista de la aplicación envía esta solicitud al controlador de la aplicación (BitácorasMuestreos).
 - El controlador de la aplicación llama al método exportarBitacora() para generar el archivo PDF de la bitácora.
 - Una vez generado el archivo PDF, la vista de la aplicación permite al usuario descargar el archivo.

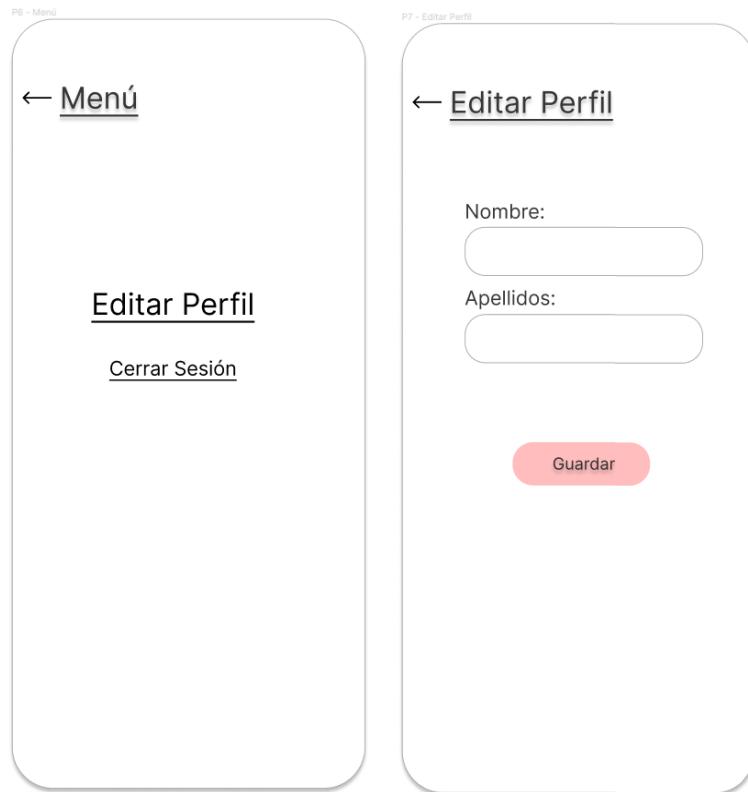
Diseño de prototipos

Logo de aplicación



Menú:

Contiene las configuraciones necesarias para acceder a las funciones como Editar Perfil o Cerrar Sesión



Pantalla de Inicio de Sesión y Registro de Usuarios:

En estas secciones contiene los espacios requeridos para acceder a la aplicación, así mismo como el registro para poder acceder a las funcionalidades de la aplicación,

P1 - Inicio de sesión

P2 - Pantalla De Registro

Inicio de Sesión

Correo:

Contraseña:

Iniciar Sesión

Crear Cuenta

Registro de Usuarios

Nombre:

Apellidos:

E-mail:

Confirmar e-mail

Contraseña:

Crear Cuenta

Cancelar

Pantalla de Bitácoras de Campo:

Esta pantalla contiene la previsualización de la Bitácora realizada por el usuario. Así mismo contiene botones que le permiten al usuario crear, ver y exportar la Bitácora.

Página Principal

Bitácoras de Campo



Nombre: Bitacora 3
Fecha: 16 de abril del 2024
Ubicación: Zacatecas

[Exportar PDF](#) [Ver Bitacora](#)



Nombre: Bitacora 2
Fecha: 12 de abril del 2024
Ubicación: Zacatecas

[Exportar PDF](#) [Ver Bitacora](#)



Nombre: Bitacora 1
Fecha: 09 de abril del 2024
Ubicación: Zacatecas

[Exportar PDF](#) [Ver Bitacora](#)

[+](#)

P3 - Página Principal Vacía

Bitácoras de Campo

[Crear Bitacora](#)

[+](#)

Pantalla de Bitácora:

En esta sección contiene los muestreos realizados por el usuario en las Bitácoras creadas, permitiéndole acceder a los muestreos realizados así como también crear un muestreo. Así mismo se le permite al usuario exportar en PDF su bitácora.

Bitácoras Completas

← Bitacora

Nombre: Bitacora 1
Fecha: 09 abril de 2024
Lugar: UPIIZ
Ubicación: 22.7840495,-102.617548,18.91z
Cantidad de Muestreos: 7
Observaciones: Pocas aves



Nombre: Cernícola
Fecha: 09 de abril del 2024
Ubicación: Zacatecas

[Ver Muestreo](#)



Nombre: Papamoscas Negro
Fecha: 09 de abril del 2024
Ubicación: Zacatecas

[Ver Muestreo](#)



P4 - Bitácoras Vacía

← Bitacora

Nombre: _____
Fecha: _____
Lugar: _____
Ubicación: _____
Cantidad de Muestreos: _____
Observaciones: _____

[Crear Muestreo](#)



Pantalla de Muestreo:

En esta pantalla el usuario podrá realizar o capturar una fotografía para realizar la identificación del ave y así mostrarle al usuario la información relacionada con la misma ave fotografiada.

Muestreo

← Muestreo



Nombre del muestreo: Falco Sparverius
Nombre coloquial del ave: Cernícalo Americano
Fecha: 09 de abril de 2024
Hora: 14:00 hrs
Coordenadas: 22.7840495,-102.617548,18.91z
Ubicación: UPIIZ
Color: Gris, Naranja, Blanco
Dimensiones: 12 cm x 8 cm

PS - Muestreo Vacío

← Muestreo



Nombre del muestreo: _____
Nombre coloquial del ave: _____
Fecha: _____
Hora: _____
Coordenadas: _____
Ubicación: _____
Color: _____
Dimensiones: _____

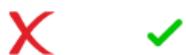
 

Alerts:

Estos elementos fueron realizados para mejorar la experiencia en el sistema del usuario y notificarle sobre algunos cambios o acciones realizadas en el sistema hechas por el mismo usuario.

PA1 - Alerta de Eliminar

¿Estas seguro de
borrar esta bitacora?



PA2 - Alerta de Duplicidad de Cuenta

Cuenta existente
Inicia sesión o regístrate con
información diferente.

PA3 - Notificación Usuario Confirmación de Correo

¡Registro exitoso!
Por favor, verifica tu correo
electrónico para confirmar tu cuenta.

PA4 - Alerta Inicio de Sesión Incorrecto

El correo electrónico no existe o aún
no se ha confirmado la dirección de
correo electrónico para la verificación

PAS5 - Alerta Registro Fallido

Registro fallido. El correo de
confirmación no se pudo enviar.

PA6 - Guardar Bitacora

Bitacora Guardada

PA7 - Guardar Muestreo

Muestreo Guardado

Formato para la exportación de la bitácora en PDF

PE1 - Formato Reporte Bitacora

Formato para el registro de aves mediante el método de conteo por puntos.

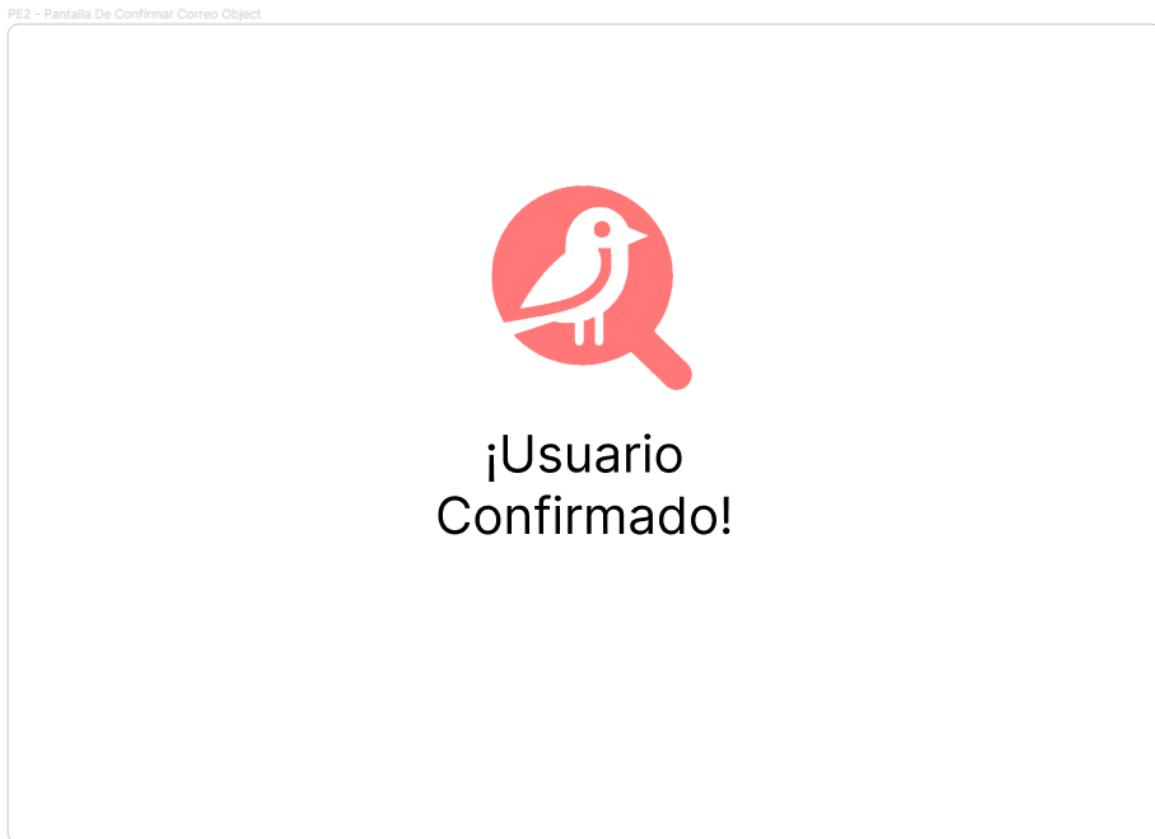
Nombre de la bitácora: _____ Lugar: _____ Fecha: _____

Ubicación: _____ Cantidad de muestreros: _____

Observaciones: _____

Pantalla web Confirmacion de Usuario:

Una vez que el usuario confirme su correo se le guiará hasta esta página donde se le dará a conocer que ya está registrado en el sistema y ya está listo para que él lo use.



Anexo F Documento de matriz de trazabilidad

CONTROL DE VERSIONES					
Autor(es)	Fecha de modificación	Versión	Descripción del cambio	Revisó	Estado
VSSL	10/04/2024	1.0	Creación del Documento	IIB	REVISADO

Objetivo	Requerimiento	Diagramas de diseño	Componente	Casos de uso	Pruebas	Intentos	Resultados
Registro y Autenticación de Usuarios	RF-01	P1 P2 PE2	Usuario Email Autenticación Postage API Usuarios	CU-01 CU-02	EP-01 EP-02 EP-03		
Gestión del Perfil de Usuario	RF-01	P6 P7	Usuario	CU-03	EP-04		
Creación y Gestión de Bitácoras de Campo	RF-02 RF-05	P3 P4	Bitácoras y Muestreos	CU-04 CU-05 CU-06 CU-11	EP-05 EP-06 EP-07 EP-12		
Manejo de Información de Muestreos	RF-03	P4 PE1	Bitácoras y Muestreos	CU-07 CU-08 CU-09	EP-08 EP-09 EP-10		
Operaciones con los Muestreos	RF-03	P4 P5	Bitácoras y Muestreos Identificación	CU-07 CU-08 CU-09	EP-08 EP-09 EP-10		
Identificación y Clasificación de las Aves Muestreadas	RF-04	P5 PE3	Identificación	CU-10	EP-11		
Cumplimiento de Normativas Ambientales	RF-05	PE1	Bitácoras y Muestreo	CU-10 CU-11	EP-11 EP-12		

Localización en Tiempo Real de los Registros	RF-02 RF-03	P4 P5	Bitácoras y Muestreos	CU-04 CU-07	EP-05 EP-08		
--	----------------	----------	-----------------------	----------------	----------------	--	--

Anexo G Plan de pruebas

ESPECIFICACIÓN DE PRUEBA						
Sistema:	feBird	Módulo:	Usuarios			
Componente:	Registro	Fecha:	23/may/24			
Id:	EP-01	Autor:	VSSL			
Nombre:	Registro de usuarios	Versión:	1.0.0			
Técnica de prueba:	Pruebas unitarias, Técnica:					
Objetivo:	Probar el correcto funcionamiento de los registros de usuarios nuevos y el envío de confirmación.					
Ambiente de pruebas:	Entorno de desarrollo, Bases de datos MySQL 8.0.29 o PostgreSQL 14.1. Computadora: Android Studio 15.1, Windows 11, Procesador: Intel i5, RAM: 24 GB Teléfono: Procesador: Snapdragon 700, RAM: 4GB, Almacenamiento: 128GB, Cámara: 13MP+2MP +2MP					
Requerimiento(s) que atiende:	RF-01					
DESCRIPCIÓN DEL ESCENARIO						
1 Iniciar la aplicación 2 Dar click en Crear cuenta 3 Ingresar el nombre de usuario 4 Ingresar los apellidos de usuario 5 Ingresar el correo de usuario 6 Ingresar el número de teléfono de usuario para su confirmación 7 Ingresar una contraseña para el usuario						

Caso de Prueba: Crear cuenta en el sistema						Fecha [dd/mm/aa]	Leer [d]
CASOS DE PRUEBA							
Caso	Nombre de usuario	Apellido de usuario	Correo del usuario	Contraseña del correo	Contraseña de usuario	Resultado esperado	Letra [V/S]
1	Vania	Lee	vaniafanybf@gmail.com	vaniafanybf@gmail.com	Vania10	Notificación usuario confirmación	[Correcto / Descripción del [X]]
2	Vania	Lee	vaniafanybf@gmail.com	vaniafanybf@gmail.com	Vania10	Alerta de duplicidad de cuenta	
3	[]	Lee	vaniafanybf@gmail.com	vaniafanybf@gmail.com	Vania10	Campos incompletos. Completar nombre de usuario	
4	Vania	[]	vaniafanybf@gmail.com	vaniafanybf@gmail.com	Vania10	Campos incompletos. Completar apellido de usuario	
5	Vania	Lee	vaniafanybf@gmail.com	[]	Vania10	Campos incompletos. Completar correo de usuario	
6	Vania	Lee	vaniafanybf@gmail.com	vaniafanybf@gmail.com	[]	Campos incompletos. Completar confirmación de contraseña	
7	Vania	Lee	correoinvalido@email.com	correovalid@email.com	Vania10	Alerta registro fallido	

ESPECIFICACIÓN DE PRUEBA			
Sistema:	IdBrid	Módulo:	Usuarios
Componente:	Registro	Fecha:	23/may/24
Id:	RF-01	Autor:	VISSEL
Nombre:	Confirmación de registro	Version:	1.0.0
Técnica de prueba:	Pruebas unitarias		
Objetivo:	Probar el correcto funcionamiento del envío de correos y la confirmación de registro de usuario.		
Ambiente de pruebas:	Conexión a internet, Base de datos: MongoDB 9.1. Computadora: Android Studio 15.1, Windows 11, Procesador: Intel i5, RAM: 24 GB teléfono: Procesador: Snapdragon 700, RAM: 4GB, Almacenamiento: 128GB, Cámara: 13MP+2MP +2MP		
Requerimiento(s) que	RF-01		

atende:																														
ESCRIPCION DEL SCENARIO																														
1 Abrir correo enviado																														
2 Dar click en el link de confirmación																														
RESULTADOS DE LAS																														
ASOS DE PRUEBA	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Asunto registrado</th> <th>Correo confirmado</th> <th>Resultado esperado</th> <th>Fecha [dd/mm/aa]</th> <th>Fecha [dd/mm/aa]</th> <th>Fecha [dd/mm/aa]</th> <th>Fecha [dd/mm/aa]</th> </tr> <tr> <th>Tester</th> <th>Tester</th> <th>Tester</th> <th>Tester</th> <th>Tester</th> <th>Tester</th> <th>Tester</th> </tr> </thead> <tbody> <tr> <td>Si</td> <td>Si</td> <td>Pantalla web de registro exitoso. Confirmado en BD</td> <td>US\$1</td> <td>US\$1</td> <td>US\$1</td> <td>US\$1</td> </tr> <tr> <td>Si</td> <td>No</td> <td>Usuario registrado, sin confirmación</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Asunto registrado	Correo confirmado	Resultado esperado	Fecha [dd/mm/aa]	Fecha [dd/mm/aa]	Fecha [dd/mm/aa]	Fecha [dd/mm/aa]	Tester	Si	Si	Pantalla web de registro exitoso. Confirmado en BD	US\$1	US\$1	US\$1	US\$1	Si	No	Usuario registrado, sin confirmación											
Asunto registrado	Correo confirmado	Resultado esperado	Fecha [dd/mm/aa]	Fecha [dd/mm/aa]	Fecha [dd/mm/aa]	Fecha [dd/mm/aa]																								
Tester	Tester	Tester	Tester	Tester	Tester	Tester																								
Si	Si	Pantalla web de registro exitoso. Confirmado en BD	US\$1	US\$1	US\$1	US\$1																								
Si	No	Usuario registrado, sin confirmación																												

ESPECIFICACIÓN DE PRUEBA			
Sistema:	IdBird	Módulo:	Usuarios
Componente:	Inicio de sesión	Fecha:	23/may/24
Id:	EP-03	Autor:	VSSL
Nombre:	Inicio de sesión de usuarios.	Versión:	1.0.0
Técnica de prueba:	Pruebas unitarias		
Objetivo:	Probar el correcto funcionamiento del inicio de sesión con cuentas ya registradas		
Ambiente de pruebas:	Conexión a internet, Base de datos: MongoDB 9.1. Computadora: Android Studio 15.1, Windows 11, Procesador: Intel i5, RAM: 24 GB Teléfono: Procesador: Snapdragon 700, RAM: 4GB, Almacenamiento: 128GB, Cámara: 13MP+2MP +2MP		
Requerimiento(s) que	RQ-01		

tiende	RF-01																														
DESCRIPCIÓN DEL ESCENARIO																															
1) Iniciar la aplicación 2) Ingresar el correo del usuario 3) Ingresar la contraseña del usuario 4) Dar click en Iniciar Sesión																															
RESULTADOS DE LAS PRUEBAS																															
<table border="1"> <thead> <tr> <th>Caso</th> <th>Usuario pre registrado</th> <th>Correo del usuario</th> <th>Contraseña del usuario</th> <th>Resultado esperado</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Si</td> <td>afelix12000@ultimo.ipn.mx</td> <td>Axel1011</td> <td>Inicio de sesión correcto</td> </tr> <tr> <td>2</td> <td>Si</td> <td>afelix12000@ultimo.ipn.mx</td> <td>axel1010</td> <td>Alerta de inicio de sesión</td> </tr> </tbody> </table>					Caso	Usuario pre registrado	Correo del usuario	Contraseña del usuario	Resultado esperado	1	Si	afelix12000@ultimo.ipn.mx	Axel1011	Inicio de sesión correcto	2	Si	afelix12000@ultimo.ipn.mx	axel1010	Alerta de inicio de sesión												
Caso	Usuario pre registrado	Correo del usuario	Contraseña del usuario	Resultado esperado																											
1	Si	afelix12000@ultimo.ipn.mx	Axel1011	Inicio de sesión correcto																											
2	Si	afelix12000@ultimo.ipn.mx	axel1010	Alerta de inicio de sesión																											
<table border="1"> <thead> <tr> <th>Caso</th> <th>Usuario pre registrado</th> <th>Correo del usuario</th> <th>Contraseña del usuario</th> <th>Resultado esperado</th> <th>Fecha [dd/mm/aa]</th> <th>Fecha [dd/mm/aa]</th> <th>Fecha [dd/mm/aa]</th> <th>Fecha [dd/mm/aa]</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Si</td> <td>afelix12000@ultimo.ipn.mx</td> <td>Axel1011</td> <td>Inicio de sesión correcto</td> <td>Tester VSSL</td> <td>Tester VSSL</td> <td>Tester AFFJ</td> <td>Tester AFFJ</td> </tr> <tr> <td>2</td> <td>Si</td> <td>afelix12000@ultimo.ipn.mx</td> <td>axel1010</td> <td>Alerta de inicio de sesión</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>					Caso	Usuario pre registrado	Correo del usuario	Contraseña del usuario	Resultado esperado	Fecha [dd/mm/aa]	Fecha [dd/mm/aa]	Fecha [dd/mm/aa]	Fecha [dd/mm/aa]	1	Si	afelix12000@ultimo.ipn.mx	Axel1011	Inicio de sesión correcto	Tester VSSL	Tester VSSL	Tester AFFJ	Tester AFFJ	2	Si	afelix12000@ultimo.ipn.mx	axel1010	Alerta de inicio de sesión				
Caso	Usuario pre registrado	Correo del usuario	Contraseña del usuario	Resultado esperado	Fecha [dd/mm/aa]	Fecha [dd/mm/aa]	Fecha [dd/mm/aa]	Fecha [dd/mm/aa]																							
1	Si	afelix12000@ultimo.ipn.mx	Axel1011	Inicio de sesión correcto	Tester VSSL	Tester VSSL	Tester AFFJ	Tester AFFJ																							
2	Si	afelix12000@ultimo.ipn.mx	axel1010	Alerta de inicio de sesión																											

ESPECIFICACIÓN DE PRUEBA					
Sistema:	Ajax	Modulo:	Usuarios		
Componente:	Actualización	Fecha:	27/may/24		
Id:	EP-04	Autor:	VSSL		
Nombre:	Actualizar usuario	Version:	1.0.0		
Técnica de prueba:	Pruebas unitarias				
Objetivo:	Probar el correcto funcionamiento de la edición y actualización de usuarios.				
Ambiente de pruebas:	Base de datos: Room				
	Computadora: Android Studio 15.1, Windows 11, Procesador: Intel i5, RAM: 24 GB				
	Teléfono: Procesador: Snapdragon 700, RAM: 4GB, Almacenamiento: 128GB, Cámara: 13MP+2MP +2MP				
Requerimiento(s) que atiende:	RF-01				

DESCRIPCIÓN DEL ESCENARIO					
1 Iniciar sesión					
2 Dar click en Menú					
3 Dar click en Editar perfil					
4 Ingresar un nombre nuevo en el espacio indicado					
5 Ingresar un apellido nuevo en el espacio indicado					
6 Dar click en Guardar					

CASOS DE PRUEBA					
Caso:	Nombre a actualizar	Apellido a actualizar	Resultado esperado		
1 Axel	Felix	Guardado de cambios exitoso en la BD local	[Correcto / Descripción del]	[X]	
2 Axel	[]	Actualización del campo apellido a en blanco	Tester	VSSL	
3 []	Felix	Actualización del campo nombre a en blanco			
4 []	[]	Actualización de los campos a en blanco			

RESULTADOS DE LAS PRUEBAS

ESPECIFICACIÓN DE PRUEBA					
Sistema:	Idbird	Modulo:	Bitácoras		
Componente:	Registro	Fecha:	27/may/24		
Id:	EP-05	Autor:	VSSL		
Nombre:	Registro de bitácoras	Version:	1.0.0		
Técnica de prueba:	Pruebas unitarias				
Objetivo:	Probar el correcto funcionamiento de los registros o creación de bitácoras de campo.				
Ambiente de pruebas:	Base de datos: Room				
	Computadora: Android Studio 15.1, Windows 11, Procesador: Intel i5, RAM: 24 GB				
	Teléfono: Procesador: Snapdragon 700, RAM: 4GB, Almacenamiento: 128GB, Cámara: 13MP+2MP +2MP				
Requerimiento(s) que atiende:	RF-02				

DESCRIPCIÓN DEL ESCENARIO					
1 Iniciar sesión					
2 Dar click en Crear Bitácora					
3 Ingresar el nombre de la bitácora					
4 Ingresar el lugar					
5 Ingresar las observaciones					
6 Dar click en Guardar Bitácora					

RESULTADOS DE LAS PRUEBAS

CASO					
Caso:	Nombre de bitácora	Lugar	Observaciones	Resultado esperado	
1 Bitácora 1	UPII2	Pocas aves	Alerta de Guardar Bitácora	[Correcto / Descripción del]	[X]
2 []	UPII2	Pocas aves	Alerta de Guardar Bitácora	Tester	VSSL
3 Bitácora 1	[]	Pocas aves	Alerta de Guardar Bitácora		
4 Bitácora 1	UPII2	[]	Alerta de Guardar Bitácora		

ESPECIFICACIÓN DE PRUEBA					
Sistema:	Idbird	Modulo:	Bitácoras		
Componente:	Actualización	Fecha:	27/may/24		
Id:	EP-06	Autor:	VSSL		
Nombre:	Actualización de bitácoras	Version:	1.0.0		
Técnica de prueba:	Pruebas unitarias				
Objetivo:	Probar el correcto funcionamiento de los edición y actualización de bitácoras de campo.				
Ambiente de pruebas:	Base de datos: Room				
	Computadora: Android Studio 15.1, Windows 11, Procesador: Intel i5, RAM: 24 GB				
	Teléfono: Procesador: Snapdragon 700, RAM: 4GB, Almacenamiento: 128GB, Cámara: 13MP+2MP +2MP				
Requerimiento(s) que atiende:	RF-02				

DESCRIPCIÓN DEL ESCENARIO					
1 Iniciar sesión					
2 Dar click en Ver Bitácora de la bitácora a editar					
3 Ingresar el nombre de la bitácora					
4 Ingresar el lugar					
5 Ingresar las observaciones					
6 Dar click en Guardar Bitácora					

RESULTADOS DE LAS PRUEBAS

CASOS DE PRUEBA					
Caso:	Nombre de bitácora	Lugar	Observaciones	Resultado esperado	
1 Bitácora 1	UPII2	Pocas aves	Alerta de Guardar Bitácora	[Correcto / Descripción del]	[X]
2 []	UPII2	Pocas aves	Alerta de Guardar Bitácora	Tester	VSSL
3 Bitácora 1	[]	Pocas aves	Alerta de Guardar Bitácora		
4 Bitácora 1	UPII2	[]	Alerta de Guardar Bitácora		

ESPECIFICACIÓN DE PRUEBA					
Sistema:	Idbird	Modulo:	Bitácoras		
Componente:	Eliminación	Fecha:	27/may/24		
Id:	EP-07	Autor:	VSSL		
Nombre:	Eliminación de bitácoras	Version:	1.0.0		
Técnica de prueba:	Pruebas unitarias				
Objetivo:	Probar el correcto funcionamiento de la eliminación de bitácoras de campo.				
Ambiente de pruebas:	Base de datos: Room				
	Computadora: Android Studio 15.1, Windows 11, Procesador: Intel i5, RAM: 24 GB				
	Teléfono: Procesador: Snapdragon 700, RAM: 4GB, Almacenamiento: 128GB, Cámara: 13MP+2MP +2MP				
Requerimiento(s) que atiende:	RF-02				

DESCRIPCIÓN DEL ESCENARIO					
1 Iniciar sesión					
2 Dar click en la cruz de Eliminar Bitácora de la bitácora a eliminar					
3 Dar click en confirmar borrado de bitácora					

RESULTADOS DE LAS PRUEBAS

CASOS DE PRUEBA					
Caso:	Confirmación de eliminación	Resultado esperado			
1 Confirmado	Eliminación de bitácora		[Correcto / Descripción del]	[X]	
2 Cancelado	Sin cambios				

ESPECIFICACIÓN DE PRUEBA						
Sistema:	Idbird	Módulo:	Muestreos			
Componente:	Registro	Fecha:	27/may/24			
Id:	EP-08	Autor:	VSSL			
Nombre:	Registro de muestreos	Versión:	1.0.0			
Técnica de prueba:	Pruebas unitarias					
Objetivo:	Probar el correcto funcionamiento de los registros o creación de muestreos.					
Base de datos:	Room					
Ambiente de pruebas:	Computadora: Android Studio 15.1, Windows 11, Procesador: Intel i5, RAM: 24 GB					
Requerimiento(s) que atiende:	RF-02					
DESCRIPCIÓN DEL ESCENARIO						
1	Dar click en Ver Bútorca de la bútorca seleccionada					
2	Dar click en Crear Muestreo					
3	Ingresar el nombre del muestreo					
4	Ingresar la ubicación					
5	Seleccionar el botón de cámara					
6	Seleccionar cámara o galería					
7	Seleccionar o capturar la foto del ave					
8	Dar click en Guardar Muestreo					
CASOS DE PRUEBA						
Caso:	Nombre del muestreo	Ubicación	Fuente de la imagen	Resultado esperado		
1	Registro 1	UPTIZ	Cámara	Alerta de Guardar Muestreo	Correcto / Descripción del	[X]
2	Registro 2	UPTIZ	Galería	Alerta de Guardar Muestreo		
3	Muestreo 3	La Bufa	Cámara	Alerta de Guardar Muestreo		
4	Muestreo 4	La Bufa	Galería	Alerta de Guardar Muestreo		
RESULTADOS DE LAS PRUEBAS						
Fecha	Fecha	Fecha	Fecha	Fecha	Fecha	Fecha
[dd/mm/aa]	[dd/mm/aa]	[dd/mm/aa]	[dd/mm/aa]	[dd/mm/aa]	[dd/mm/aa]	[dd/mm/aa]
Tester	Tester	Tester	Tester	Tester	Tester	Tester
VSSL	VSSL	VSSL	VSSL	VSSL	VSSL	VSSL
AFFJ	AFFJ	AFFJ	AFFJ	AFFJ	AFFJ	AFFJ

ESPECIFICACIÓN DE PRUEBA						
Sistema:	Idbird	Módulo:	Muestreos			
Componente:	Actualización	Fecha:	27/may/24			
Id:	EP-10	Autor:	VSSL			
Nombre:	Actualización de muestreos	Versión:	1.0.0			
Técnica de prueba:	Pruebas unitarias					
Objetivo:	Probar el correcto funcionamiento de la edición y actualización de muestreos.					
Base de datos:	Room					
Ambiente de pruebas:	Computadora: Android Studio 15.1, Windows 11, Procesador: Intel i5, RAM: 24 GB					
Requerimiento(s) que atiende:	RF-02					
DESCRIPCIÓN DEL ESCENARIO						
1	Dar click en Ver Bútorca de la bútorca seleccionada					
2	Ingresar el nombre del muestreo					
3	Ingresar la ubicación					
5	Seleccionar una opción de las proporcionadas en Nombre coloquial del ave					
6	Dar click en Guardar Muestreo					
CASOS DE PRUEBA						
Caso:	Nombre del muestreo	Ubicación	Nombre coloquial del ave	Resultado esperado		
1	Muestreo de carpintero	UPTIZ	Carpintero de Arizona	Alerta de Guardar Muestreo	Correcto / Descripción del	[X]
1		La Bufa	Papamoscas píñero	Alerta de Guardar Muestreo		
3	Muestreo de carbonero	[]	Carbonero mexicano	Alerta de Guardar Muestreo		
4	Muestreo de zacatonero	La Bufa	[]	Alerta de Guardar Muestreo		
RESULTADOS DE LAS PRUEBAS						
Fecha	Fecha	Fecha	Fecha	Fecha	Fecha	Fecha
[dd/mm/aa]	[dd/mm/aa]	[dd/mm/aa]	[dd/mm/aa]	[dd/mm/aa]	[dd/mm/aa]	[dd/mm/aa]
Tester	Tester	Tester	Tester	Tester	Tester	Tester
VSSL	VSSL	VSSL	VSSL	VSSL	VSSL	VSSL
AFFJ	AFFJ	AFFJ	AFFJ	AFFJ	AFFJ	AFFJ

ESPECIFICACIÓN DE PRUEBA						
Sistema:	Idbird	Módulo:	Identificación			
Componente:	Identificación	Fecha:	27/may/24			
Id:	EP-11	Autor:	VSSL			
Nombre:	Identificación de aves	Versión:	1.0.0			
Técnica de prueba:	Pruebas unitarias					
Objetivo:	Probar el correcto funcionamiento de la identificación y clasificación de las imágenes.					
Base de datos:	Room					
Ambiente de pruebas:	Computadora: Android Studio 15.1, Windows 11, Procesador: Intel i5, RAM: 24 GB					
Requerimiento(s) que atiende:	RF-02					
DESCRIPCIÓN DEL ESCENARIO						
1	Elegir la imagen a identificar					
CASOS DE PRUEBA						
Caso:	Imagen	Imagen	Resultado esperado			
1	Imagen sin aves	Imagen no identificada	Imagen no identificada	Correcto / Descripción del	[X]	
2	Imagen no nítida	Imagen no identificada	Imagen no identificada			
3	Imagen de otra ave	Imagen identificada, Ave no clasificada	Imagen identificada, Ave no clasificada			
4	Imagen de Carpintero de Arizona	Imagen identificada, Ave clasificada y datos de Carpintero de Arizona	Imagen identificada, Ave clasificada y datos de Carpintero de Arizona			
5	Imagen de Papamoscas cardenalito	Imagen identificada, Ave clasificada y datos de Papamoscas cardenalito	Imagen identificada, Ave clasificada y datos de Papamoscas cardenalito			
6	Imagen de Carbonero mexicano	Imagen identificada, Ave clasificada y datos de Carbonero mexicano	Imagen identificada, Ave clasificada y datos de Carbonero mexicano			
7	Imagen de Zacatonero serrano	Imagen identificada, Ave clasificada y datos de Zacatonero serrano	Imagen identificada, Ave clasificada y datos de Zacatonero serrano			
RESULTADOS DE LAS PRUEBAS						
Fecha	Fecha	Fecha	Fecha	Fecha	Fecha	Fecha
[dd/mm/aa]	[dd/mm/aa]	[dd/mm/aa]	[dd/mm/aa]	[dd/mm/aa]	[dd/mm/aa]	[dd/mm/aa]
Tester	Tester	Tester	Tester	Tester	Tester	Tester
VSSL	VSSL	VSSL	VSSL	VSSL	VSSL	VSSL
AFFJ	AFFJ	AFFJ	AFFJ	AFFJ	AFFJ	AFFJ

ESPECIFICACION DE PRUEBA										
Sistema:	Idbird	Modulo:	Bitácoras							
Componente:	Registro	Fecha:	27/may/24							
Plataforma:	RF-05	Nombre:	VSSL							
Nombre:	Registro de bitácoras	Versión:	1.0.0							
Técnica de prueba:	Pruebas unitarias									
Objetivo:	Probar el correcto funcionamiento de los registros o creación de bitácoras de campo.									
Ambiente de pruebas:	Base de datos: Room Código: Java, Android Studio 15.1, Windows 11, Procesador: Intel i5, RAM: 24 GB Teléfono: Processor: Snapdragon 700, RAM: 4GB, Almacenamiento: 128GB, Cámara: 13MP+2MP +2MP									
Requerimiento(s) que atiende:	RF-02									
DESCRIPCIÓN DEL ESCENARIO										
1)Iniciar sesión 2)Dar click en Ver Bitácora de la bitácora elegida 3)Dar click en Exportar Bitácora										
RESULTADOS DE LAS PRUEBAS										
CASO	Detalles de bitácora	Resultado esperado	Fecha [dd/mm/aa]	Fecha [dd/mm/aa]	Fecha [dd/mm/aa]					
1	Bitácora vacía	Se crea un archivo pdf con el formato. El usuario decide que hacer con el.	[Correcto / Descripción del]	[X]	[dd/mm/aa]					
2	Bitácora con muestras	Se crea un archivo pdf con el formato. El usuario decide que hacer con el.	VSSL	VSSL	AFEJ					