

**PROTOTIPO DE UN SISTEMA DE VISIÓN ARTIFICIAL BASADO EN
TÉCNICAS DE APRENDIZAJE DE MÁQUINA PARA EL RECONOCIMIENTO DE
AVES DEL ÁREA METROPOLITANA DE BUCARAMANGA**

MARÍA ANGÉLICA MOSQUERA MORENO

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA-UNAB
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA
2021

**PROTOTIPO DE UN SISTEMA DE VISIÓN ARTIFICIAL BASADO EN
TÉCNICAS DE APRENDIZAJE DE MÁQUINA PARA EL RECONOCIMIENTO DE
AVES DEL ÁREA METROPOLITANA DE BUCARAMANGA**

MARÍA ANGÉLICA MOSQUERA MORENO

Trabajo de grado para optar por el título de ingeniero de sistemas.

Director:

Msc. Feisar Moreno Corzo

Codirector:

Mtr. Ariel Orlando Ortiz Beltrán

Asesor:

Blgo. Fernando Cediel Martínez

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA-UNAB
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA
2021**

Nota de Aceptación

Presidente del Jurado

Jurado

Jurado

Bucaramanga, Santander, Colombia. 09 de mayo de 2021.

CONTENIDO

	Pág.
1. INTRODUCCIÓN	13
2. PLANTEAMIENTO DEL PROBLEMA	15
2.1 DEFINICIÓN DEL PROBLEMA	15
2.2 JUSTIFICACIÓN	17
2.3 PREGUNTA DE INVESTIGACIÓN.....	17
3. OBJETIVOS	18
3.1 OBJETIVO GENERAL	18
3.2 OBJETIVOS ESPECÍFICOS	18
4. RESULTADOS OBTENIDOS.....	19
4.1 RESULTADOS OBTENIDOS POR OBJETIVO ESPECIFICO.....	19
5. ANTECEDENTES.....	22
6. ESTADO DEL ARTE.....	23
7. MARCO TEÓRICO	36
8. DISEÑO METODOLÓGICO.....	58
8.1 INTRODUCCIÓN	58
8.2. MODELO DE CICLO DE VIDA DEL SOFTWARE Y MODELO DE IA.	59
8.2.1. MODELO DEL CICLO DE VIDA DEL SOFTWARE	59
8.2.2. MODELO DEL CICLO DE VIDA DEL MODELO MACHINE LEARNING	63
8.3 ANÁLISIS Y DISEÑO	65
8.3.1 INFORMACIÓN DEL DOMINIO DEL PROBLEMA.....	65
8.3.2 CATÁLOGO DE REQUISITOS DEL SISTEMA A DESARROLLAR	66
8.3.3 MODELO DE CLASES DEL SISTEMA	94
8.3.4 ARQUITECTURA DEL SISTEMA	96
8.4. DISEÑO DE INTERFACES Y MANUAL DE USUARIO DEL SISTEMA.	97
8.4.1. MAPA DEL SISTEMA	97
9. ANÁLISIS DE RESULTADOS.....	101
9.1. ANÁLISIS EXPLORATORIO	101
9.1.1 DATAFRAMES.....	102
9.1.2. ANÁLISIS POR ESPECIE.....	105

9.2. ANÁLISIS DE DESARROLLO DEL MODELO DE IA.....	114
9.3. ANÁLISIS DE TEST, PRUEBAS DE FUNCIONALIDAD Y ACEPTACIÓN.	115
10. CONCLUSIONES Y RECOMENDACIONES	127
BIBLIOGRAFÍA.....	129
DESARROLLO.....	142

LISTA DE TABLAS

Tabla 1 Resultados Esperados por Actividades.....	19
Tabla 2 Estado del arte.....	24
Tabla 3 Prototipos por Caso de Uso	62
Tabla 4 Prototipos Complementarios.....	62
Tabla 5 Descripción de Actores	65
Tabla 6 Requisitos Generales del Sistema	66
Tabla 7 Descripción de Casos de Usos del Sistema.....	68
Tabla 8 Especificación de Casos de Uso del Sistema	70
Tabla 9 Requisitos no Funcionales del Sistema.....	90
Tabla 10 Descripción de Datos por Especie	101
Tabla 11 Plan de Pruebas de Modelo de IA.....	114
Tabla 12 Información de Usuarios de Prueba	118
Tabla 13 Respuestas de Test de Pruebas	118
Tabla 14 Respuestas de Encuesta de Funcionalidad	121
Tabla 15 Cumplimiento de Casos de Uso.....	122
Tabla 16 Respuestas de percepción Cualitativa por Funcionalidades.....	124
Tabla 17 Cuadro de Clasificación	136
Tabla 18 Clasificación de Proyectos por Arquitecturas CN	140
Tabla 19 Información de los Usuarios Entrevistados	143
Tabla 20 Preguntas test Antes y Después de Usar la App.....	155
Tabla 21 Plan de Pruebas de Funcionalidad.....	157
Tabla 22 Plan de Pruebas de Aceptación	158

LISTA DE FIGURAS

Ilustración 1 Árbol de problemas.....	16
Ilustración 2 Distribución Estado del Arte.....	23
Ilustración 3 Tendencia de uso de artículos.....	24
Ilustración 4 Ejemplo Data Augmentation	39
Ilustración 5 Ejemplo de Dropuot	39
Ilustración 6 Diagrama de Bloques de las Etapas de un Sistema de Visión Artificial.....	41
Ilustración 7 Procesamiento de Imágenes	42
Ilustración 8 Clasificación/Reconocimiento de Formas.	43
Ilustración 9 Modelo Lineal como un Gráfico.	43
Ilustración 10 Grafo de un Modelo de dos Capas.	44
Ilustración 11 Grafo de un Modelo de Tres Capas.....	45
Ilustración 12 Gráfico del Modelo de Tres Capas con Función de Activación.	46
Ilustración 13 Función de activación sigmoide.....	47
Ilustración 14 Función de activación ReLU.	47
Ilustración 15 Esquema General de una CNN.	48
Ilustración 16 Cambios del peso con el Objetivo de Alcanzar el Mínimo en SGD	49
Ilustración 17 Gradiente por Descenso Estocástico.....	50
Ilustración 18 Transfer Learning.	51
Ilustración 19 Modelo de Servicios	52
Ilustración 20 Arquitectura de microservicio.....	53
Ilustración 21 Logo de Flask.	54
Ilustración 22 Ejemplo de Script con Flask.....	55
Ilustración 23 Estructura de archivos con Flask.....	55
Ilustración 24 Logo de Flutter.....	56
Ilustración 25 Logo Dart.....	57
Ilustración 26 Jerarquía de Widgets en Flutter.....	57
Ilustración 27 Modelo Ciclo de Vida Prototipados.	60
Ilustración 28 Modelo detallado prototipados.....	61
Ilustración 29 ciclo de vida modelo Machine Learning.	64
Ilustración 30 Diagrama de Casos de Uso.....	69
Ilustración 31 Diagrama de Secuencia CU001-1	83

Ilustración 32 Diagrama de Secuencia CU001-2	83
Ilustración 33 Diagrama de Secuencia CU001-3	84
Ilustración 34 Diagrama de Secuencia CU001-4	84
Ilustración 35 Diagrama de Secuencia CU002-1	85
Ilustración 36 Diagrama de Secuencia CU002-2	85
Ilustración 37 Diagrama de Secuencia CU002-3	86
Ilustración 38 Diagrama de Secuencia CU002-4	86
Ilustración 39 Diagrama de Secuencia CU003-1	87
Ilustración 40 Diagrama de Secuencia CU003-2	87
Ilustración 41 Diagrama de Secuencia CU004.....	88
Ilustración 42 Diagrama de Secuencia CU005.....	88
Ilustración 43 Diagrama de Secuencia CU006.....	89
Ilustración 44 Diagrama de Secuencia CU007.....	89
Ilustración 45 Diagrama de Secuencia CU008.....	90
Ilustración 46 Diagrama de Clases	94
Ilustración 47 Modelo Entidad - Relación.....	94
Ilustración 48 Modelo Relacional	95
Ilustración 49 Diagrama de Despliegue	96
Ilustración 50 Interfaces Sistema Web.....	97
Ilustración 51 Interfaces Aplicación Móvil	98
Ilustración 52 Flujo de Interfaces Aplicación Web	99
Ilustración 53 Flujo de Intefaces Aplicación Móvil	100
Ilustración 54 Descripción de datos por clases Dataframe Resumen.....	102
Ilustración 55 Estadísticas Básicas Dataframe Resumen	103
Ilustración 56 Dataframe Train.....	103
Ilustración 57 Dataframe Validate	103
Ilustración 58 Dataframe Test.....	103
Ilustración 59 Imagen y Nombre de Cada Especie	104
Ilustración 60 Análisis Especie Amazilia tzacatl	105
Ilustración 61 Análisis de Especie Brotogeris jugularis	106
Ilustración 62 Análisis de Especie Buteo magnirostris	107
Ilustración 63 Análisis de Especie Columbina talpacoti.....	108
Ilustración 64 Especie Cuatro Aplicando Algoritmo de Semejanza, Imagen Tres	108

Ilustración 65 Análisis de Especie Coragyps Atratus	109
Ilustración 66 Análisis de Especie Melanerpes Rubricapillus	110
Ilustración 67 Análisis de Especie Pitangus Sulphuratus	111
Ilustración 68 Análisis de Especie Tiaris Bicolor	112
Ilustración 69 Análisis de Especie Tyrannus Melancholicus.....	113
Ilustración 70 Edad de Usuarios de Prueba.....	118
Ilustración 71 Puntaje Obtenido por Usuarios Antes de Utilizar la Aplicación.....	121
Ilustración 72 Gráfica de Cumplimiento de Funcionalidades.....	123
Ilustración 73 Observaciones de Usuarios Encuesta de Funcionalidad.	123
Ilustración 74 Gráfica de percepción Cualitativa de la Aplicación.....	125
Ilustración 75 Gráfica de percepción de aspectos de Experiencia de Usuario.	125
Ilustración 76 Observaciones de Usuario Encuesta de Percepción	126
Ilustración 77 Interfaz "explorar" PlantSnap	145
Ilustración 78 Interfaz "Usuario" PlantSnap.....	145
Ilustración 79 Interfaz "foto" PlantSnap	145
Ilustración 80 Interfaz "Información de la Planta" PlantSnap.....	145
Ilustración 81 Interfaz de Tomar o Seleccionar imagen del ave Merlin.....	147
Ilustración 82 Menú Lateral Merlin	147
Ilustración 83 Interfaz de Centrar el Ave en la Imagen Merlin	148
Ilustración 84 Información del Ave Merlin	148
Ilustración 85 Sonidos del ave Merlin.....	148
Ilustración 86 Paquete de Aves a Instalar de la Aplicación	148
Ilustración 87 Interfaz "Atrapar al perro" DogScanner	149
Ilustración 88 Calculando la Raza del Perro DogScanner.....	149
Ilustración 89 Información de la Raza del Perro DogScanner	150
Ilustración 90 Predicción de la Raza del Perro DogScanner	150

LISTA DE ANEXOS

ANEXO 1.....	135
ANEXO 2.....	141
ANEXO 3.....	152
ANEXO 4.....	154
ANEXO 5.....	159

GLOSARIO

Avifauna: Conjunto de las aves de un país o región (RAE, 2020).

Machine Learning: es una disciplina científica del ámbito de la Inteligencia Artificial que crea sistemas que aprenden automáticamente. Aprender en este contexto quiere decir identificar patrones complejos en millones de datos (González, 2019).

Visión Artificial o Visión por Computadora: se define como un campo de estudio que busca desarrollar técnicas para ayudar a las computadoras a “ver” y comprender el contenido de imágenes digitales como fotografías y videos (Brownlee, 2019).

PROTOTIPO DE UN SISTEMA DE VISIÓN ARTIFICIAL BASADO EN TÉCNICAS DE APRENDIZAJE DE MÁQUINA PARA EL RECONOCIMIENTO DE AVES DEL ÁREA METROPOLITANA DE BUCARAMANGA

María Angélica Mosquera Moreno, Autor
Feisar Enrique Moreno Corzo, Director
Ariel Orlando Ortiz Beltrán, Codirector

RESUMEN

La avifauna colombiana es una de las más diversas a nivel mundial. Desafortunadamente, es uno de los campos menos enseñados por las instituciones educativas e incluso con menor información a nivel nacional. Seguido, la falta de interés por parte de la población hace más difícil fomentar el apropiamiento del entorno que les rodea e impulsa el deterioro ambiental y la extinción de muchas especies de aves endémicas de la zona.

Actualmente existen diferentes aplicaciones que buscan suplir esta problemática en diferentes lugares del mundo, pero, dado que existen muchas variedades de especies, en estas soluciones se dificulta la identificación de especies de la zona metropolitana de Bucaramanga. AveDex busca crear apropiación local, ya que es una aplicación con solo aves del lugar, permitiendo que sea apoyo para los habitantes y ornitólogos.

El desarrollo de este proyecto permitió la obtención de un prototipo capaz de clasificar 9 aves originarias del área metropolitana de Bucaramanga, con una precisión del 87% por medio de una imagen tomada desde la cámara del celular o una imagen escogida de la galería. Finalmente, demostró que el uso constante de ella o utilizada como herramienta de información eleva el aprendizaje sobre las aves que actualmente se pueden identificar. Actualmente, el proyecto hace parte del grupo GTI y la línea de sistemas de información e ingeniería del software más específicamente en el tema de inteligencia artificial de la Universidad Autónoma de Bucaramanga.

Palabras clave – Machine Learning, visión artificial, avifauna, clasificación.

1. INTRODUCCIÓN

El presente trabajo de proyecto de grado tiene como objetivo principal construir una aplicación móvil capaz de identificar aves en el área metropolitana de Bucaramanga mediante la clasificación de imágenes desde un sistema de visión artificial para proveer información sobre la avifauna a los habitantes de la zona. Colombia es uno de los países con mayor registro de aves en el mundo, pero se tiene muy poca información a la mano para los habitantes e incluso para los turistas que realizan aviturismo en el país. Las aplicaciones móviles forman parte de la vida cotidiana, en un estudio desarrollado por Deloitte mostró que la mayoría de las personas tiene acceso a un smartphone (71% año 2015) (Deloitte, 2017). Por otro lado, para el 2019 se realizó un estudio sobre cuánto es el consumo de los móviles en Colombia, lo cual arrojó que un 76% de los usuarios manifestaron usar demasiado su teléfono (Deloitte, 2019) lo que da pie a que se creen soluciones didácticas y prácticas.

La visión artificial es un campo de la inteligencia artificial que, mediante la utilización de técnicas adecuadas, permite la obtención, procesamiento y análisis de cualquier tipo de información obtenida a través de imágenes digitales (Generalitat de Catalunya, 2012). Actualmente, la tecnología de visión por computadora se está utilizando en diferentes sectores como, la atención médica para el diagnóstico de enfermedades; la fabricación para asegurar que los productos y procesos de fabricación no tengan imperfecciones; la seguridad como los análisis biométricos como la retina, reconocimiento facial y escaneo de huellas dactilares; el transporte para guiar vehículos autónomos mediante la detección de obstáculos, personas y señales de tráfico en el camino (Vlvek, 2019).

El documento está estructurado de la siguiente forma: inicialmente, se encuentra el planteamiento del problema en donde se hace la definición del problema y la justificación del desarrollo de este proyecto sustentado en dar solución al problema planteado. Luego, se encuentran los objetivos generales y específicos, los cuales expresan las metas a alcanzar en el desarrollo del proyecto. Seguidamente, se definen los resultados esperados especificados por objetivos, posteriormente por actividades derivadas de cada objetivo específico. Después de esto, se enunciarán los antecedentes que describen la trayectoria que ha tenido esta idea desde su concepción hasta la actualidad. Posteriormente, encontrará la revisión literaria o estado del arte enumerados por fichas, en donde se describe de forma ordenada la información de cada artículo y lo que aportan al proyecto.

Asimismo, la definición del marco teórico, en donde se especifican las teorías que dan sustento al proyecto y son necesarias para el entendimiento de este. A continuación, se hace la descripción del diseño metodológico con el que se cumplirán los objetivos planteados, dando paso al cronograma de actividades definido a un año en donde se especifican la distribución de las semanas en cada actividad. Finalmente, se tiene el presupuesto total del costo del proyecto y la bibliografía.

2. PLANTEAMIENTO DEL PROBLEMA

2.1 DEFINICIÓN DEL PROBLEMA

El estudio de la avifauna colombiana ha catapultado al país al reconocimiento mundial por ser uno de los más ricos en vida silvestre y tropical. Ahora bien, no existe una información amplia y clasificada de las especies que se encuentran en cada región de Colombia, por motivos de que no se han hecho estudios, y los pocos que hay son muy específicos y no están publicados (Acevedo-Charry, 2020). En la zona metropolitana de Bucaramanga, según el ornitólogo Fernando Cediél egresado de la universidad industrial de Santander (Cediél Martínez, 2020) hay un estimado de 160 especies de aves y se aumenta cerca de 200; si se agregan los sitios periurbanos se obtiene un total de 1909 en todo el territorio continental e insular del país (Avendaño, y otros, 2017). Por otro lado, desde el punto de vista de Cristián Samper, Colombia tiene una riqueza estimada de 1941 especies siendo este el país con mayor diversidad de aves, en el prólogo del libro “Avifauna Colombiana” dice que, a pesar de que era un niño muy curioso nunca tuvo acceso a una guía de campo que le acompañara y que luego como estudiante de biología en varias expediciones realizadas a diferentes zonas rurales le generaba una gran frustración no contar con más libros de distribución o ilustraciones de las aves (Ayerbe Quiñones, 2019).

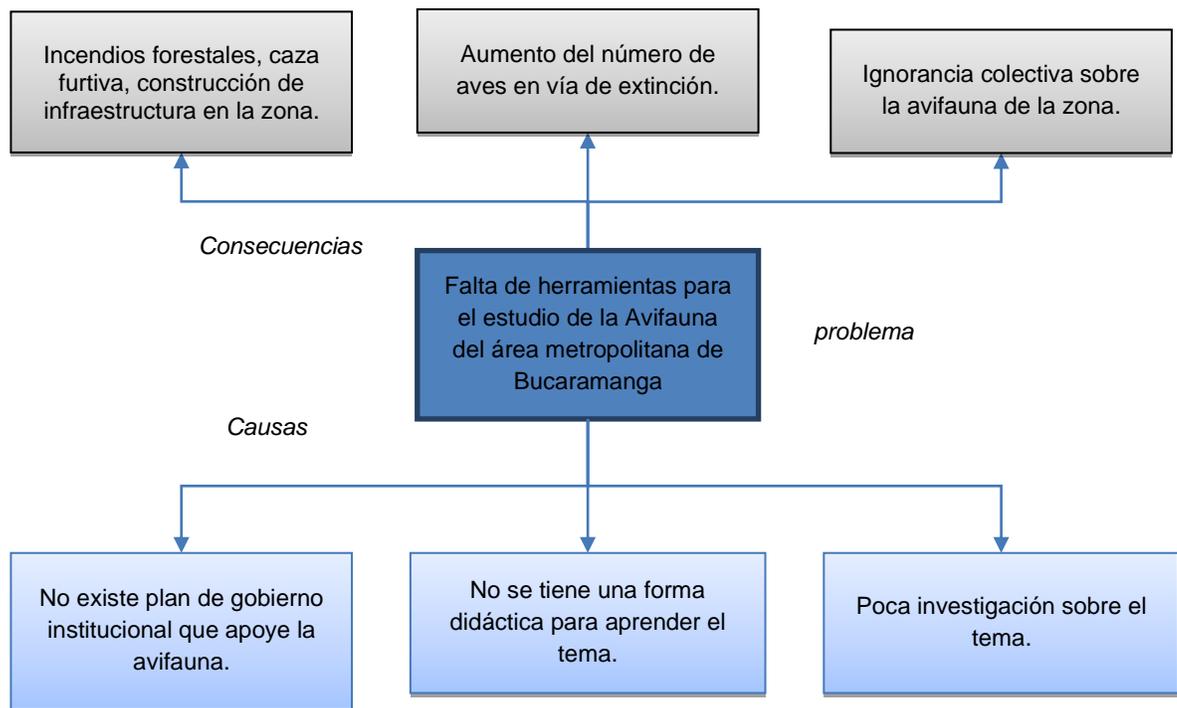
Las personas que tienen como tarea realizar estas investigaciones por lo general son ornitólogos, guardabosques o consultores ecológicos; En los últimos años ha estado creciendo el aviturismo, un estudio realizado en el 2016 que presentó Strategy Fund encontró que al menos 14.987 turistas estarían interesados en venir a Colombia para observar aves (Revista semana, 2018). A pesar de esto, se puede presentar una reducción de visitas porque es muy difícil dependiendo de la experiencia de la persona identificar a cada especie y poder clasificarla sin una guía o herramienta específica que ayuden con el trabajo.

Los conocimientos generales sobre la naturaleza que posee el país no es uno de los mejores apoyados desde el sistema educativo porque no existe un plan de gobierno institucional que se enfoque en la fauna colombiana (Ministerio de educación, 1962) en donde se puede aprender asignaturas como biología, a nivel general, pero no específicamente sobre las especies primitivas de Colombia, más específicamente la avifauna. Los estudiantes de básica primaria, básica

secundaria o incluso universitaria no son capaces de diferenciar la fauna entre los países, por lo que no se tiene un crecimiento amigable con la biodiversidad y medio ambiente llevando a que ese niño, adolescente o adulto no le interese la avifauna que se encuentra en el país. Consecuentemente, se destruye el ecosistema por medio de incendios forestales, construcción de infraestructura, caza furtiva, o contaminación de bosques dando paso a que cada vez haya más aves en vía de extinción (Martinez Castillo, 2012).

Una encuesta aplicada entre agosto de 2018 y agosto de 2020 (Mosquera Moreno, 2020) a 59 habitantes del área metropolitana determinó diferentes factores que han impedido conocer el medio, estos son: que se debe a la poca enseñanza que proveen los colegios de la zona equivalente 16,9% en adición con 67,8%; la falta de motivación por parte de los habitantes para salir a observar las especies equivalente al 10,2%; es mucha información para ser recordada equivalente al 5,1%; no hay una manera didáctica en el aula para enseñar el tema y finalmente que hay poca investigación sobre el tema en la población.

Ilustración 1 Árbol de problemas



Fuente: Autor.

2.2 JUSTIFICACIÓN

Esta idea nació de la falta de una herramienta para identificar aves sin tener el conocimiento empírico o profesional para hacerlo, enfocada a todas las personas de la zona para generar apropiación del entorno que poseen de una forma interactiva. También busca crear conciencia sobre los daños que causa el calentamiento global, es bien sabido que diferentes ambientes se han venido afectando a causa de este fenómeno, ocasionando el desplazamiento de muchas especies a lugares que no son su hábitat natural, como consecuencia de este proceso, muchas aves se convierten en especies invasoras y alteran el equilibrio natural de las zonas a las que llegan (AQUAE Fundación, 2020), en la zona metropolitana de Bucaramanga puede ocurrir la llegada de alguna de estas especies, con ayuda de esta aplicación y el reconocimiento de estas especies no endémicas de la región, las personas tendrán la posibilidad de entregar estos datos a autoridades competentes que puedan implementar una solución e identificar las especies en vía de extinción.

Por otro lado, se impulsará a los municipios implicados en el turismo de avistamiento de una manera creativa y organizada. Además, con la implantación de esta aplicación se puede obtener más información sobre las especies de aves que posee la zona metropolitana de Bucaramanga que puede ser de ayuda para investigaciones de profesionales en el tema o de gusto por la naturaleza que tienen las personas. Finalmente, el desarrollo de este sistema mostrará las habilidades obtenidas durante toda la formación de la carrera de ingeniería de sistemas, desde la abstracción de una problemática hasta la creación de una herramienta tecnológica que lo resuelva.

2.3 PREGUNTA DE INVESTIGACIÓN.

¿Puede el uso de una herramienta software promover en las personas el aprendizaje sobre la avifauna del área metropolitana de Bucaramanga de una manera didáctica y fácil, que genere gusto y eleve el número de actividades sobre turismo ecológico?

3. OBJETIVOS

3.1 OBJETIVO GENERAL

Desarrollar un prototipo de un sistema para identificar aves en el área metropolitana de Bucaramanga mediante la clasificación de imágenes desde un sistema de visión artificial para proveer información sobre la avifauna a los habitantes de la zona.

3.2 OBJETIVOS ESPECÍFICOS

- Contrastar tecnologías pertinentes en el reconocimiento de aves que permiten el tratamiento de imágenes en visión por computador.
- Diseñar el sistema de reconocimiento de aves orientado a una aplicación móvil.
- Implementar el sistema de reconocimiento de aves en la aplicación móvil diseñada.
- Evaluar la aplicación por medio de pruebas de funcionalidad y aceptación que permitan determinar el incremento de información sobre la avifauna del área metropolitana.

4. RESULTADOS OBTENIDOS

En la tabla 1 se encuentran los resultados obtenidos detallados por cada una de las actividades asociadas a cada objetivo específico que dan cumplimiento al objetivo general. La columna de “anexo” contiene links interactivos a la ubicación del documento referenciado dentro del presente.

4.1 RESULTADOS OBTENIDOS POR OBJETIVO ESPECIFICO.

Tabla 1 Resultados Esperados por Actividades

Objetivo	Actividad	Resultado	Anexo
<i>Contrastar tecnologías pertinentes en el reconocimiento de aves que permiten el tratamiento de imágenes en visión por computador.</i>	Búsqueda y revisión bibliográfica.	Estado del arte.	6. Estado Del Arte
	Construcción de un cuadro de clasificación sobre tecnologías.	Documento con clasificación de tecnologías utilizadas en la visión artificial.	Anexo 1
	Selección de la tecnología a utilizar.	Definición de requisito no funcional.	Anexo 1
<i>Diseñar el sistema de reconocimiento de aves orientado a una aplicación móvil.</i>	Elaboración y aplicación de entrevista a usuarios especializados y usuarios normales.	Lista de requerimientos funcionales y no funcionales.	Anexo 2
	Análisis de diferentes soluciones tecnológicas en la clasificación de imágenes.	Lista de requerimientos funcionales y no funcionales.	Anexo 2
	Diseño del flujo de trabajo Deep Learning.	Diagrama de flujo de trabajo.	8.2.2. Modelo Del Ciclo De Vida Del Modelo Machine Learning
	Definición de la arquitectura de la solución que permita la interacción entre la aplicación móvil y el modelo de clasificación.	Diagrama de despliegue.	8.3.4 Arquitectura Del Sistema
	Definición de funcionalidades del	Diagrama de casos de uso y diagrama de	8.3.2 Catálogo De

	aplicativo móvil.	interacción.	Requisitos Del Sistema A Desarrollar
	Modelado de base de datos.	Modelo entidad-relación y relacional de la base de datos.	8.3.3.1 Diagramas de Clases del Sistema
	Definición del vocabulario de la base de datos.	Diagrama de clases del sistema.	8.3.3 Modelo De Clases Del Sistema
	Definición de interfaces.	Diagrama de flujo de interfaces de la aplicación móvil.	8.4.1. Mapa Del Sistema Flujo de interfaces AveDex App Flujo de interfaces AveDex Web
	Búsqueda de datasets de especies de aves.	Lista de dataset disponibles.	Anexo 3
<i>Implementar el sistema de reconocimiento de aves en la aplicación móvil diseñada.</i>	Creación banco de imágenes sobre aves.	Dataset de especies de aves definidas.	Banco De Imágenes
	Análisis del dataset creado.	Análisis exploratorio.	9.1. Análisis Exploratorio Análisis Exploratorio Completo

	Desarrollo de sistema de visión artificial.	Modelo entrenado del clasificador.	Modelo
	Creación de la aplicación.	APK	APK pública Anexo 5
<i>Evaluar la aplicación por medio de pruebas de funcionalidad y aceptación que permitan medir el incremento del interés por la avifauna del área metropolitana de Bucaramanga.</i>	Diseño de pruebas de usuario.	Plan de pruebas de funcionalidad.	Anexo 4
	Diseño de pruebas de aceptación.	Plan de pruebas de aceptación	Anexo 4
	Ejecución de plan de pruebas de funcionalidad.	Informe técnico de pruebas de funcionalidad.	9.3. Análisis de test, pruebas de funcionalidad y aceptación.
	Ejecución de plan de pruebas de aceptación.	Informe de aceptación de pruebas. Documento con posibles trabajos futuros.	9.3. Análisis de test, pruebas de funcionalidad y aceptación. 10. Conclusiones y recomendaciones

Fuente: Autor.

5. ANTECEDENTES

La idea de este proyecto surgió en el semillero de investigación MediaLab a cargo del Mtr. Ariel Ortiz Beltrán en el periodo académico 2018-II junto a la estudiante de ingeniería de sistemas Bibiana Patricia Franco Florez, enmarcada en el grupo GTI y la línea de sistemas de información e ingeniería del software más específicamente en el tema de inteligencia artificial de la Universidad Autónoma de Bucaramanga.

Este proyecto se presentó como una propuesta de investigación la cual hizo participé en el XII encuentro de semilleros de investigación “El intercambio del conocimiento: base para el desarrollo” realizado en el campus el jardín de la universidad autónoma de Bucaramanga en el semestre 2018-II.

Luego, se hizo participé como proyecto de semillero en Ingeniotic: Salón del Ingenio y la Innovación, evento organizado por el programa de ingeniería de Sistemas de la UNAB en el año 2018.

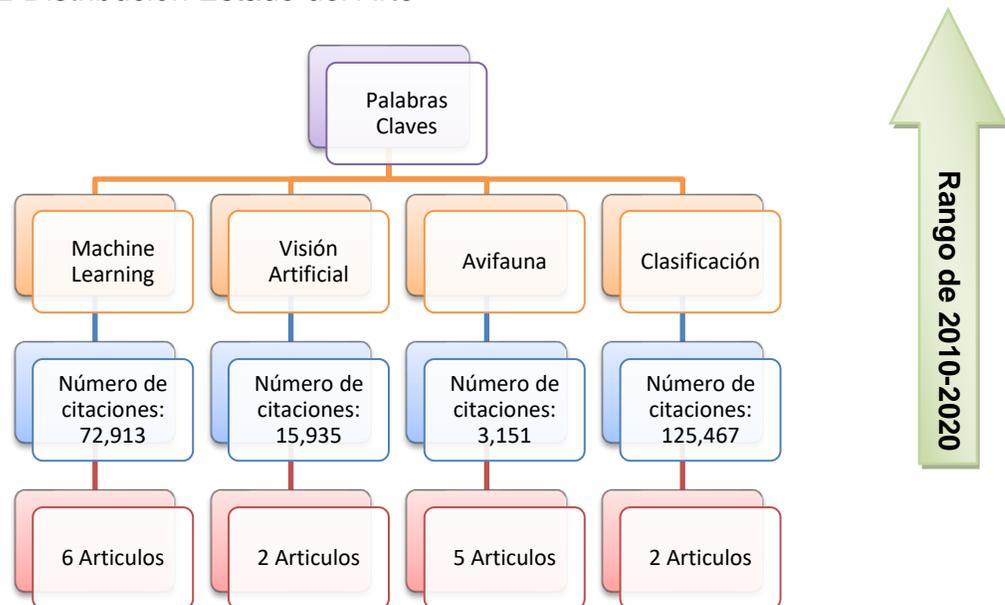
Como prototipo beta se desarrolló una aplicación web en Python con Flask, la cual tenía la funcionalidad de recibir la imagen del ave, comunicarse con el motor de clasificación y redireccionar a una página web definida con la información del ave identificada. El clasificador fue desarrollado con Tensorflow y redes convolucionales, para el entrenamiento, se tenía un dataset de autoría propia con un total de 7 especies: pyrocephalus rubinus - Titiribí; Ramphocelus Icteropus – Tangara culiguala; Yellow-naped amazon – Loro; Ara macao – Guacamaya roja; Trochilidae – Colibri; Thraupis episcopus – Azulejo; Serinus canaria – Canario. Cada una tenía más de 100 fotos en diferentes dimensiones y ángulos encontradas en diferentes sitios web, en el preprocesamiento estas eran redimensionadas a un tamaño uniforme para realizar un entrenamiento limpio, obteniendo un accuracy de 70% de clasificación.

Actualmente, este proyecto se encuentra inscrito en el semillero de investigación SISWEB a cargo del Msc. Paulo Cesar Ramírez Prada, con un alcance definido a un año, en el cual se hará el desarrollo de la aplicación móvil para la clasificación de las aves basado en una metodología de gamificación, también se espera mejorar el accuracy y entrenar con imágenes de aves nativas de la zona metropolitana de Bucaramanga.

6. ESTADO DEL ARTE

Se realizó una búsqueda bibliográfica en diferentes tesauros como IEEEXplore y link Springer, tomando las siguientes palabras claves: machine learning, visión artificial, avifauna y clasificación, con un rango de fechas de 2010-2020 con el fin de obtener información actualizada de lo que se está realizando con proyectos similares. En la ilustración 1 se tiene la distribución de las palabras claves por artículos citados en el documento.

Ilustración 2 Distribución Estado del Arte

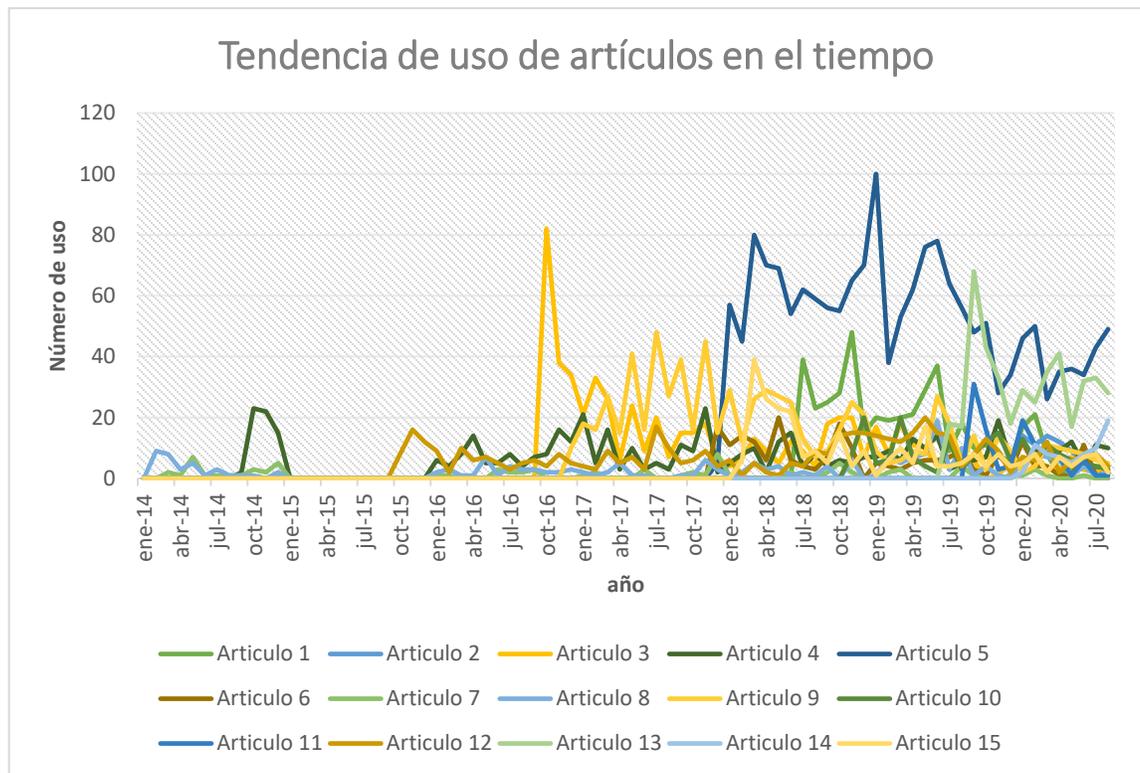


Fuente: Autor.

Los 6 artículos correspondientes a la palabra "Machine Learning" son del 1 al 6 de la tabla 3; los 2 correspondientes a "visión artificial" son los 12 y 13; los 5 de "avifauna" son los de 7 al 11 y finalmente, los que pertenecen a "clasificación" son el 14 y 15.

Por otra parte, en la ilustración 2 se hizo la recopilación de los usos que han tenido los artículos numerados en relación con la tabla 2 por año. A partir de esto, se concluye que las palabras claves han sido mencionadas con más auge los últimos 4 años (2016-mitad de 2020). Inicialmente, la gráfica fue construida desde el 2010 pero, dado que entre el 2010 y 2014 no tenían muchas visitas, se omitió de esta.

Ilustración 3 Tendencia de uso de artículos



Fuente: Autor.

En total se revisaron 15 artículos enumerados en la tabla 2, especificados por referencia bibliográfica que hace referencia al nombre del artículo y la referencia. Luego, de manera resumida se expone el problema del artículo. Seguidamente, la solución planteada y conclusión obtenida. Finalmente, se tiene el aporte que hace al proyecto de forma objetiva.

Tabla 2 Estado del arte

1	Referencia bibliográfica	Performance Comparison of Deep Learning Techniques for Recognizing Birds in Aerial Images (Liu, y otros, 2018)
	Problema del proyecto	La mayoría de los métodos de aprendizaje profundo para la detección de objetos se han desarrollado para objetos grandes y su rendimiento en la detección de objetos pequeños no es muy bueno.

	Solución	Este artículo contribuye a la investigación en detección de objetos pequeños de baja resolución mediante la evaluación del desempeño de los principales métodos de aprendizaje profundo para la detección de objetos utilizando un conjunto de datos en común, que es un nuevo conjunto de datos para la detección de aves.
	Conclusión	Entre los métodos de detección de objetos evaluados, SSH (Single Stage Headless) obtuvo el mejor rendimiento en los casos fáciles, mientras que Tiny Face logró la mejor precisión en los casos difíciles. Entre los métodos de segmentación de instancias, U-Net logró un mejor rendimiento que Mask R-CNN. Estos resultados son útiles para identificar las fortalezas y debilidades de los métodos existentes y el desarrollo de métodos futuros con un rendimiento mejorado.
	Aporte al proyecto	Aporta un ponderado de diferentes métodos de detección de objetos y el análisis de cada uno de ellos para identificar aves en imágenes aéreas.
2	Referencia bibliográfica	MobileNet Model for Classifying Local Birds of Bangladesh from Image Content Using Convolutional Neural Network (Islam, Tasnim, & Bhatta Shuvo, 2019)
	Problema del proyecto	Clasificar especies de aves es una tarea bastante difícil debido a la compleja interdependencia en varios factores. Ha habido numerosos intentos de perfeccionar la clasificación.
	Solución	Clasificar las especies de aves a partir de datos de imágenes con un sistema de clasificación de visión por computadora.
	Conclusión	En este artículo, se presentó un modelo llamado MobileNet que ofrece una precisión de hasta el 100% para la clasificación local de especies de aves.
	Aporte al proyecto	Ofrece un análisis del modelo MobileNet y la construcción de este en el marco de

		TensorFlow para la clasificación de aves.
3	Referencia bibliográfica	Machine Learning-Based Image Classification for Wireless Camera Sensor Networks (Jungmo, Jeongyeup, & JeongGil, 2016)
	Problema del proyecto	¿Cómo recopilar y analizar datos del mundo real a microescala?
	Solución	Utilizar redes de sensores inalámbricos para capturar fenómenos físicos a microescala comparado a la desventaja de usar cámaras: la muestra (imágenes) tienen tamaños grandes, lo que obliga al sistema intercambiar más datos y, a su vez, disminuye la eficiencia energética.
	Conclusión	El objetivo de este trabajo es utilizar algoritmos de aprendizaje automático para caracterizar el contexto de las imágenes capturadas de implementaciones del mundo real, utilizan y optimizan algoritmos de aprendizaje automático para operar en plataformas integradas de baja potencia y recursos limitados.
	Aporte al proyecto	Provee una propuesta de arquitectura de sistema como una red de sensores de cámaras para el monitoreo de las aves que permita obtener imágenes de tamaños más compactos y fáciles de administrar y calcular en el motor de clasificación.
4	Referencia bibliográfica	Birdsnap: Large-Scale Fine-Grained Visual Categorization of Birds (Berg, y otros, 2014)
	Problema del proyecto	Categorización visual detallada a gran escala, describiendo nuevos métodos que utilizaron para producir una guía de campo en línea para 500 especies de aves de América del Norte. Se enfocan en el desafío que surge cuando se le pide al sistema que distinga entre especies de aves muy similares.
	Solución	Presentaron diferentes formas de hacer más preciso el clasificador: "Clasificadores uno contra la mayoría" en donde eliminan

		especies muy similares durante el entrenamiento; estimación de los antecedentes de clase espaciotemporales a partir de observaciones que se muestrean en ubicaciones irregulares y sesgadas; finalmente, el rendimiento de reconocimiento de vanguardia en un nuevo conjunto de datos grande.
	Conclusión	Al evaluar todos los métodos propuestos se concluyó que los tres daban resultados bastante satisfactorios para la clasificación de aves.
	Aporte al proyecto	Aporta un dataset de imágenes de aves que se pueden tener en común para entrenar el modelo de clasificación.
5	Referencia bibliográfica	Learning Multi-attention Convolutional Neural Network for Fine-Grained Image Recognition (Zheng, Fu, Mei, & Luo, 2017)
	Problema del proyecto	El reconocimiento de categorías de especies de aves depende en gran medida de la localización discriminativa del ave y el aprendizaje de características detallada basado en el ave. Los enfoques existentes resuelven estos desafíos de forma independiente, sin tener en cuenta la correlación de la localización parcial (ej, La cabeza de un pájaro) y el aprendizaje de rasgos finos (ej. La forma de la cabeza)
	Solución	Proponen un enfoque novedoso de aprendizaje de partes mediante una red neuronal convolucional de atención múltiple (MA-CNN), donde la generación de partes y el aprendizaje de características pueden reforzarse mutuamente.
	Conclusión	Se propuso una red neuronal convolucional de atención múltiple para el reconocimiento de especies de aves, que aprenden conjuntamente la localización discriminativa de partes y la representación de rasgos de las especies.
	Aporte al proyecto	Provee el nombre de un conjunto de datos de especies de aves (Caltch-UCSD Birds) y

		el nombre de diferentes algoritmos para el entrenamiento: 5 de ellos utilizan la supervisión humana, y los últimos ocho se basan en métodos de aprendizaje parcial no supervisado.
6	Referencia bibliográfica	Automatic bird identification for offshore wind farms: A case study for deep learning (Niemi & Tanttu, 2019)
	Problema del proyecto	Se requiere un sistema automático de identificación de aves para los parques eólicos marinos en Finlandia a partir de imágenes digitales.
	Solución	Aplicaron un método de aprendizaje profundo para la clasificación de imágenes y desarrollaron una técnica de expansión de datos para los datos de entrenamiento.
	Conclusión	La red neuronal convolucional no profunda es un modelo adecuado para la clasificación de imágenes en la aplicación del mundo real, específicamente cuando la cantidad de datos de entrenamiento es limitada. Además, presentaron y demostraron que la técnica de datos mejora significativamente el rendimiento de la clasificación.
	Aporte al proyecto	Aporta un valioso diseño del sistema desde la parte global hasta cuando se aplica el modelo de red convolucional que puede ser una opción para desarrollar el sistema de clasificación de aves en la zona metropolitana de Bucaramanga.
7	Referencia bibliográfica	Topic model based bird breed classification and annotation (Huang, Luo, Tang, Liu, & Ma, 2013)
	Problema del proyecto	La clasificación de imágenes es un problema central en la visión artificial que juega un papel importante en muchas áreas, como la compresión de la esencia y la detección de objetos. La clasificación de imágenes de tipos de aves ha presentado un desafío para los métodos tradicionales por la variación intraclase y la similitud entre clases dificultan la obtención de una alta precisión de

		clasificación.
	Solución	Utilizar un modelo para la clasificación de razas de aves, se propuso un modelo gráfico que considera la prominencia (GMS). En el modelo, se integraron el contexto local, el contexto global y la información destacada y se utilizó SVM (Support Vector Machine) para hacer la clasificación.
	Conclusión	En este artículo, se clasificó el ave según un modelo gráfico que considera la saliencia, es decir lo notable o sobresaliente a simple vista. Al utilizar la información proporcionada por el mapa de prominencia, se puede lograr un mejor resultado de anotación y clasificación.
	Aporte al proyecto	Aporta un conjunto de datos llamado Caltech-UCSD Birds. Además, la explicación de la segmentación de la imagen para obtener las características importantes para la clasificación.
8	Referencia bibliográfica	Bird Species Classification Based on Color Features (Marini, Facon, & Koerich, 2013)
	Problema del proyecto	La identificación de especies de aves a partir de imágenes es un problema importante y desafiante con muchas aplicaciones en el mundo real, como la protección del medio ambiente y el rescate de animales en peligro.
	Solución	Desarrollar un modelo para el tratamiento de las imágenes obteniendo características de color para construir una línea para la identificación de especies de aves.
	Conclusión	Presentaron una serie de experimentos realizados en un conjunto de datos compuestos por más de 6.000 imágenes de 200 especies de aves diferentes. Han considerado dos espacios de color diferentes, RGB y HSV, y un número diferente de especies que se clasificarán para evaluar la escalabilidad del enfoque propuesto
	Aporte al proyecto	Provee el tratamiento de imágenes para

		obtener una buena clasificación a partir de sus colores.
9	Referencia bibliográfica	Picking Deep Filter Responses for Fine-Grained Image Recognition (Zhang, Xiong, Zhou, Lin, & Tian, 2016)
	Problema del proyecto	Reconocer las subcategorías detalladas como las aves y los perros es extremadamente difícil debido a las diferencias altamente localizadas y sutiles en algunas partes específicas.
	Solución	El documento propone un enfoque de reconocimiento automático detallado que está libre de cualquier anotación de objeto, parte en las etapas de entrenamiento y prueba.
	Conclusión	Proponen elegir buenos filtro que respondan a partes específicas de manera significativa y consistente. En base a esos filtros seleccionados, se elige elaboradamente muestras positivas y se entrena un conjunto de detectores discriminativos de forma iterativa. Además, proponen un método de codificación llamado SWFV-CNN que incluye descriptores locales de CNN a través de una combinación espacialmente ponderada de Vectores Fisher.
	Aporte al proyecto	Utilizaron el conjunto de datos Caltech-UCSD Bird para realizar el entrenamiento y las pruebas: contiene 11.788 imágenes que abarcan 200 subespecies. También, la forma del preentrenamiento que les dieron a los datos.
10	Referencia bibliográfica	Automatic Bird-Species Recognition using the Deep Learning and Web Data Mining (Kang & Hong, 2018)
	Problema del proyecto	Los problemas de sobreajuste y el lento tiempo de aprendizaje, que fueron los principales problemas de la técnica de reconocimiento de aprendizaje profundo, se han resuelto gradualmente mediante desarrollos de software y hardware, y la investigación relacionada se ha ampliado a

		varios campos de aplicación. Además, la cantidad exponencialmente aumentada de datos en línea se ha vuelto más fácil de recopilar como datos de aprendizaje para las redes neuronales (NN), y los datos refinados se han compartido fácilmente para el aprendizaje de NN.
	Solución	Proponen un método para generar un modelo de reconocimiento de especies de aves mediante el aprendizaje de un modelo basado en datos de entrenamientos refinados. En primer lugar, se ingresa el nombre de la raza de aves objetivo, la imagen se recopilará de la Web utilizando el rastreo de imágenes. Para refinar las imágenes recopiladas en el conjunto de datos de entrenamiento, la imagen dañada se corrige, se elimina el valor atípico y, finalmente, la imagen se expande para obtener los datos de entrenamiento refinados
	Conclusión	En este documento, cuando solo se ingresa la lista de nombres con respecto a la creación del reconocedor, la imagen necesaria para el reconocedor de imágenes se recopila automáticamente utilizando el recopilador de imágenes web. En este estudio, se presenta la comparación de las precisiones y las matrices de confusión de dos modelos DCNN VGGNet.
	Aporte al proyecto	Aporta el modelado de la clasificación de imágenes de aves mediante un diagrama de flujo y el estudio de dos modelos que pueden ser de ayuda para construir el planteado.
11	Referencia bibliográfica	Bird Detection in Agriculture Environment using Image Processing and Neural Network (Lee, Lee, Jeon, & Smith, 2019)
	Problema del proyecto	La detección de aves de plaga en un campo bajo sigue siendo un desafío, ya que es difícil distinguir objetos en el fondo dinámico. Aunque, se ha utilizado el aprendizaje

		profundo para solucionarlo, detectar objetos en un video no es lo suficientemente rápido ya que el proceso necesita un alto costo computacional y si el objeto es pequeño, la precisión se reduce.
	Solución	Proponen un método de detección de aves de forma dinámica para dispositivos de bajo rendimiento mediante el uso de procesamiento de imágenes y aprendizaje profundo. Utilizan dos técnicas, la sustracción de fondo y la red neuronal del aprendizaje profundo.
	Conclusión	Aplicaron tres técnicas de procesamiento de imágenes para la detección de objetos en movimiento, los ruidos se redujeron de manera efectiva. Utilizaron modelos de redes neuronales para la clasificación de objetos. El modelo pre-entrenado y el modelo reentrenado se realizaron para comparar desempeños.
	Aporte al proyecto	Provee la técnica de sustracción de fondo para realizar la clasificación en tres pasos: <ol style="list-style-type: none"> 1. La resta de fondo basada en mezcla gaussiana captura todos los objetos en movimiento. 2. Las técnicas de extracción de color se utilizan para eliminar los colores de los movimientos dinámicos que se encuentran dentro de los árboles y césped. 3. Se aplican filtros medios para minimizar el ruido.
12	Referencia bibliográfica	Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection (Van Horn, y otros, 2015)
	Problema del proyecto	La visión por computadora ha entrado en una era de grandes datos, donde se busca recopilar conjuntos de datos más grandes, en términos de la cantidad de clases, la cantidad de imágenes por clase y el nivel de anotación por imagen, parece ser primordial

		para la mejora continua del rendimiento y expandiendo el conjunto de aplicaciones solucionables.
	Solución	En este documento, se presentaron herramientas y metodologías para construir grandes conjuntos de datos de visión por computadoras de alta calidad, basados en el uso de un grupo alternativo de anotadores de multitudes: científicos ciudadanos. Los cuales son encargados de clasificar estas imágenes.
	Conclusión	Descubrieron que los algoritmos de aprendizaje basados en las características de CNN y la localización de piezas eran robustos para ejemplos de entrenamiento mal etiquetados.
	Aporte al proyecto	Contribuye con: <ol style="list-style-type: none"> 1. Metodologías para recopilar conjuntos de datos de visión por computadora de alta calidad por medio de científicos ciudadanos. 2. NABirds: conjunto de datos de alta calidad de 555 categorías seleccionadas por expertos con un total de 48, 562 imágenes. 3. Análisis empírico del efecto que tiene la calidad de anotación al entrenar algoritmos de visión por computadora de última generación para la categorización.
13	Referencia bibliográfica	Piecewise Classifier Mappings: Learning Fine-Grained Learners for Novel Categories With Few Examples (Wei, Wang, Liu, & Wu, 2019)
	Problema del proyecto	El reconocimiento de razas detalladas como la identificación de aves, perros o automóviles ha presentado un problema desafiante porque las diferentes categorías, ya que solo se pueden distinguir por diferencias leves y sutiles.
	Solución	Proponen una red entrenable de extremo a extremo que se inspira en el modelo de

		reconocimiento de vanguardia y se adapta a la tarea (FSFG) porque hay categorías que son identificadas por pocos ejemplos y los clasificadores no pueden generalizar más allá de las supervisiones limitadas utilizando mapeos por partes.
	Conclusión	La aplicación de la propuesta redujo significativamente los parámetros de los modelos dando resultados prometedores.
	Aporte al proyecto	Provee la estrategia de mapeo por partes aplicadas al clasificador CNN (red neuronal convolucional) desde un reconocimiento de imágenes de pocos disparos aplicado al reconocimiento de patrones, que aprenden sobre la marcha, a partir de pocos o incluso un solo ejemplo.
14	Referencia bibliográfica	Are These Birds Similar: Learning Branched Networks for Fine-grained Representations (Nawaz, Calefati, Caraffini, Landro, & Gallo, 2019)
	Problema del proyecto	La clasificación de imágenes detalladas se enfoca en discriminar entre categorías o clases difíciles de distinguir lo que hace difícil la clasificación de aves.
	Solución	Proponen una estrategia para aprender la representación conjunta de la descripción de las imágenes desde el lenguaje natural utilizando una red de dos ramas con múltiples capas para mejorar la tarea de clasificación detallada.
	Conclusión	Los resultados obtenidos muestran que el enfoque es efectivo y aumenta significativamente la precisión obtenida con los métodos existentes con una mejora de alrededor del 10%.
	Aporte al proyecto	Aporta la referencia de un conjunto de datos de imágenes de aves llamado CUN-200-2011 que contiene 11788 imágenes de 200 especies de aves dividida en 5994 y 5794 imágenes para entrenar y probar.
15	Referencia bibliográfica	Bird species recognition based on SVM classifier and decision tree (Qiao, Zhou,

	Yang, & Cao, 2017)
Problema del proyecto	El reconocimiento de aves basado en señales de imágenes enfrenta un desafío debido a la iluminación, diferentes puntos de la cámara, el fondo, etc.
Solución	Proponen un método de reconocimiento de aves basado en los clasificadores decisión Tree y Support vector machine (SVM) a partir de la característica del pico para clasificarlas. Aunque, también usan el color del cuerpo y la cabeza para identificarlas.
Conclusión	La tasa de clasificación correcto que alcanzaron los métodos fue del 84%, la precisión de clasificación tiene alguna variación al cambiar la función del pico. El método de usar el árbol de decisión tuvo un impacto positivo que aumentó la tasa de 3 a 5%.
Aporte al proyecto	Aporta una característica que puede ser fundamental para identificar aves como es la del pico y el tratamiento de los colores como el RGB de las imágenes para mejorar la clasificación.

Fuente: Autor.

7. MARCO TEÓRICO

INTELIGENCIA ARTIFICIAL.

¿Pueden pensar las máquinas? (Alan Turing Computing machinery and Intelligence, Mind, octubre, 1950). “*Existirá Inteligencia Artificial cuando no seamos capaces de distinguir entre un ser humano y un programa de ordenador en una conversación a ciegas*” Alan Turing (Pineda, 2009)

El nombre de Inteligencia Artificial surgió en la conferencia de John McCarthy en 1956 en el Dartmouth College, New Hampshire. En dicho foro, se discutía sobre como simular la inteligencia humana a través de máquinas. Por lo tanto, se ha llegado a que la inteligencia artificial es una de las áreas más interesantes y llena de retos de las ciencias de la Computación ya que ha tomado a la inteligencia como la característica universalmente aceptada para diferenciar a los humanos de otras criaturas ya sean vivas o inanimadas, para construir programas o computadoras inteligentes. Además, dichas máquinas, pretenden imitar funciones cognitivas humanas como el razonamiento, la memoria, la decisión y finalmente la inteligencia a los ordenadores.

Por consiguiente, se puede decir que la inteligencia artificial son sistemas que piensan como humanos, sistemas que actúan como humanos, sistemas que piensan racionalmente y sistemas que actúan racionalmente. Entonces, surgen conceptos como son el aprendizaje de máquinas (**machine learning**), el aprendizaje profundo (**deep learning**) y las redes neuronales. Una parte de la inteligencia artificial se basa en el aprendizaje de máquinas, es decir, el desarrollo de programas que, basándose en el suministro de información en forma de ejemplos, puedan generalizar comportamientos, de forma parecida a como aprende el cerebro humano. Otra analogía con la neurología ha sido la creación de redes neuronales que permiten generar sistemas que imitan el funcionamiento del cerebro en la forma en que este clasifica la información recibida, se trata de máquinas que realizan sus diagnósticos o predicciones basándose en la probabilidad. Finalmente, el concepto de aprendizaje profundo hace referencia a que las redes neuronales tienen una estructura basada en numerosas capas. Las redes neuronales que permiten el aprendizaje profundo son complejos sistemas matemáticos que aprenden por sí solos analizando vastas cantidades de datos.

Por otra parte, el **reconocimiento de patrones** trata de diferentes técnicas de clasificación para identificar los subgrupos con características comunes en cada grupo, y con el grado de asociación se obtiene una conclusión diferente. Los algoritmos que se desarrollan en dicha área son herramientas útiles en otros campos como el reconocimiento de lenguaje natural, la visión por computadora, reconocimiento de imágenes, el diagnóstico de fallos de equipos, entre otros.

Machine Learning.

Es una rama de la inteligencia artificial encargada de crear programas de software capaces de generalizar comportamientos a partir de los datos obtenidos. Dentro del Machine Learning existen tres problemas generales que se pueden abordar de forma diferentes a partir de ejemplos y experiencia.

1. Método de Regresión.

Este método se utiliza para predecir el valor de un atributo continuo. Consiste en encontrar la mejor ecuación que atraviese de forma óptima un conjunto de puntos. Se utiliza cuando la precisión no es crítica y el número de variables es pequeño (Contreras, 2016)

2. Método de Clasificación.

Este método es utilizado para predecir un resultado de un atributo con valor discreto dadas unas características. El método simple de clasificación es el binario, donde se clasifica un registro de variables de entrada 1 o 0. La clasificación múltiple es una extensión de la clasificación binaria (Contreras, 2016)

3. Método de Agrupación.

Se utiliza cuando se necesita clasificar las instancias de datos, pero no se conocen previamente las categorías. Esta agrupación permite construir grupos (clúster) coherentes de instancias teniendo en cuenta las variables de la data. En palabras sencillas, permite encontrar qué se tiene en la data (Contreras, 2016)

Los algoritmos utilizados se pueden agrupar en dos: aprendizaje supervisado, y aprendizaje No supervisado. En primer lugar, se tiene que un **aprendizaje supervisado** es aquel que para un conjunto de datos de entrada conocemos de antemano los datos correctos de salida. En la fase de entrenamiento se cuenta con un conjunto de datos que son con los que se enseñan o se entrena al algoritmo para encontrar los patrones y relaciones en la data. Posteriormente, en la fase de pruebas se cuenta con una data de pruebas, la cual sirve para validar el rendimiento del algoritmo. Los métodos que se encuentran en este tipo de algoritmos son: **Regresión**, el cual trata de predecir resultados con una salida continua. Es decir, mapear variables de entrada a alguna función continua. **Clasificación**, trata de predecir resultados con una salida discreta, mapear variables de entrada en categorías discretas.

Seguidamente, se tiene en el **aprendizaje no supervisado** para un conjunto de datos de entrada, NO conocemos de antemano los datos de salida. Se utiliza para

obtener una agrupación coherente de los datos en función de las relaciones entre las variables definidas en la data.

- **Sobreajuste - Alta Varianza (Overfitting)** (Contreras, 2016): la hipótesis o función seleccionada mapea casi perfectamente la tendencia de los datos de entrenamiento, pero puede fallar en la generalización de nuevos registros.

Para combatir el sobreajuste se puede:

- a. reducir el número de características o eliminar las no relevantes.
- b. seleccionar otro tipo de algoritmo que maneje mejor este problema.
- c. aplicar la regularización.

- **Subajuste - Alta Oscilación (Underfitting)** (Contreras, 2016): la hipótesis o función seleccionada mapea pobremente la tendencia de los datos de entrenamiento, con lo cual también tiene problemas con la nueva generalización de nuevos registros.

Para combatir subajuste se puede:

- aumentar el número de características relevantes.
- incluir un número mayor de registros.

MÉTODOS DE OPTIMIZACIÓN:

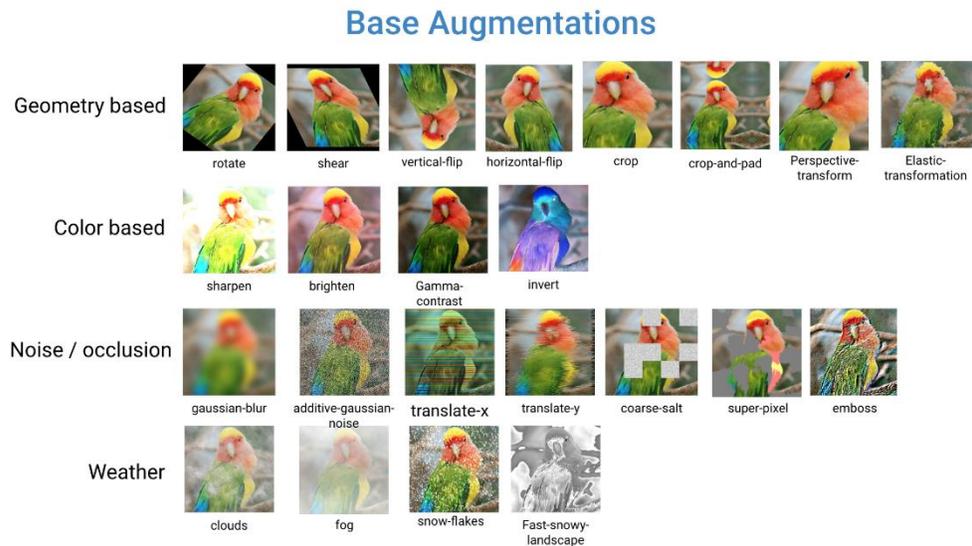
Se utilizan para disminuir el sobreajuste o subajuste:

- **Capas de regularización L1 y L2:** consisten en añadir penalizaciones a la función de pérdida, es decir, se añade un término a la función de pérdida. Las capas de regularización permiten disminuir la función de pérdida evitando la complejidad del modelo.
L1, consiste en añadir $-\lambda\|W\|$, este método ayuda a dispersar los vectores de pesos. La L2, añade $-\frac{1}{2}\lambda\|W\|^2$ lo que penaliza la magnitud cuadrados de todos los parámetros directamente en la función de pérdida que ayuda a mitigar los posibles picos en los vectores de pesos (Sancho Peña, 2017).
- **Data Augmentation:** es una manera de disminuir el Overfitting o underfitting, este proceso consiste en tomar los datos existentes y modificarlo con el fin de aumentar la variedad (evitando el sesgo) durante el entrenamiento. En la clasificación de imágenes existen diferentes técnicas que pueden ser la rotación, la trasposición de las imágenes, añadir ruido

aleatorio o variaciones en el color con el fin de mejorar la capacitación de la red neuronal (González Diez, 2019).

El aumento de datos consigue que todas las clases tengan las mismas muestras, es decir, que se encuentren balanceados.

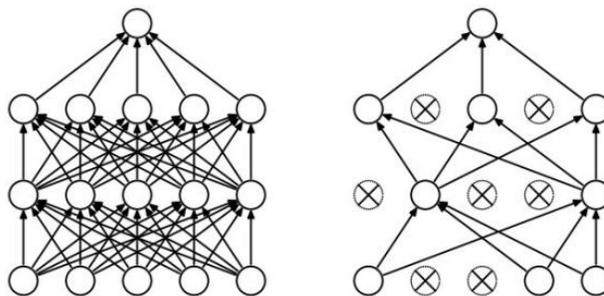
Ilustración 4 Ejemplo Data Augmentation



Fuente: (González Diez, 2019).

- **Dropout:** Se aplica entre las capas fully connected, su funcionamiento consiste en desconectar ciertas neuronas de dichas capas durante el entrenamiento con una probabilidad p . Su propósito es el de evitar que ciertas neuranas se adapten demasiados entre sí (Sancho Peña, 2017).

Ilustración 5 Ejemplo de Dropout



Fuente: (Sancho Peña, 2017).

- **Batch Normalization:** Es una técnica de aprendizaje supervisado que convierte las salidas entre capas en una red neuronal en un formato

estándar, llamado normalización, con el fin de establecer la distribución de la salida de la capa anterior para que la posterior la procese de manera más eficiente. (DeepAI, 2020).

VISIÓN ARTIFICIAL.

Se puede definir la “Visión Artificial” como un campo de la “Inteligencia Artificial” que, mediante la utilización de las técnicas adecuadas, permite la obtención procesamiento y análisis de cualquier tipo de información espacial obtenida a través de imágenes digitales (Centro Integrado Politécnico "ETI").

La visión artificial es una disciplina que engloba todos los procesos y elementos que proporcionan ojos a una maquina y se podría decir que: la visión artificial o comprensión de imágenes describe la deducción automática de la estructura y propiedades de un mundo tridimensional, posiblemente dinámico, bien a partir de una o varias imágenes bidimensionales de ese mundo (Ordieres-Meré, y otros, 2006).

Etapas en un proceso de visión artificial:

El primer paso en el proceso es **adquirir la imagen digital**. Para ello se necesitan sensores y la capacidad para digitalizar la señal producida por el sensor.

Una vez que la imagen digitalizada ha sido obtenida, el siguiente paso consiste en el **preprocesamiento de dicha imagen**. El objetivo del preprocesamiento es mejorar la imagen de forma que el objetivo final tenga mayores posibilidades de éxito.

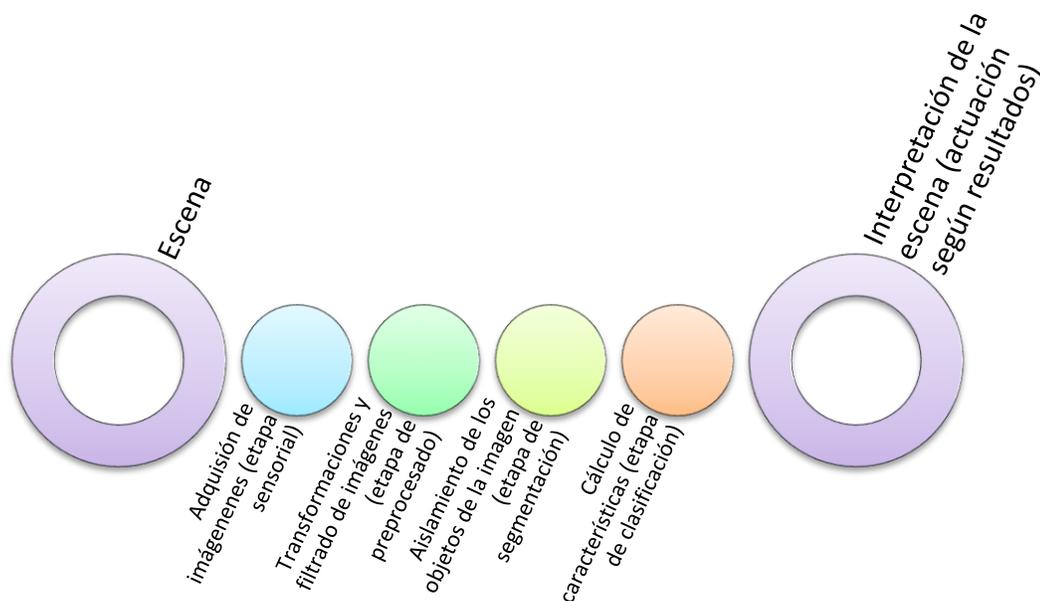
El paso siguiente es la **segmentación**. Definida en sentido amplio, su objetivo es dividir la imagen en las partes que la constituyen o los objetos que la forman. En general la segmentación autónoma es uno de los problemas más difíciles en el procesamiento de la imagen. Por una parte, una buena segmentación facilitará mucho la solución del problema; por otra, la segmentación errónea conducirá al fallo.

La salida del proceso de segmentación es una imagen de datos que, o bien contienen la frontera de la región o los puntos de ella misma. Es necesario convertir estos datos a una forma que sea apropiada para el ordenador. La primera decisión es saber si se va a usar la representación por frontera o región completa. La representación por la frontera es apropiada cuando el objetivo se centra en las características de la forma externa como esquinas o concavidades y convexidades. La representación por regiones es apropiada cuando la atención se centra en propiedades internas como la textura o el esqueleto. Sin embargo, en muchas aplicaciones ambas representaciones coexisten.

La elección de una representación es sólo una parte de la transformación de los datos de entrada. Es necesario especificar un método que extraiga los datos de interés. La **parametrización**, que recibe también el nombre de selección de rasgos se dedica a extraer rasgos que producen alguna información cuantitativa de interés o rasgos que son básicos para diferenciar una clase de objetos de otra.

En último lugar se encuentran el **reconocimiento y la interpretación**. El reconocimiento es el proceso que asigna una etiqueta a un objeto basada en la información que proporcionan los descriptores (clasificación). La interpretación lleva a asignar significado al conjunto de objetos reconocidos.

Ilustración 6 Diagrama de Bloques de las Etapas de un Sistema de Visión Artificial



Fuente: Autor en base (Centro Integrado Politécnico "ETI").

Para utilizar la visión artificial es necesario contemplar:

- **Procesamiento de imagen:** procesador, algoritmos de pretratamiento y filtrado de la imagen, de segmentación, reconocimiento de formas, de extracción de descriptores y de clasificación.

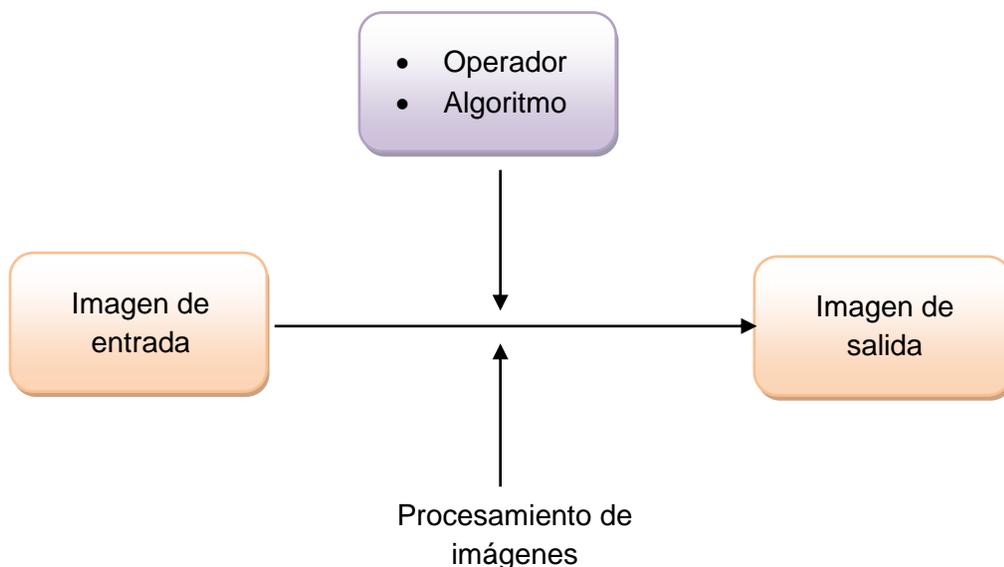
Niveles de procesamiento en visión artificial:

- *A Bajo Nivel (Procesamiento de Imágenes)* (Escuela Superior de Ingeniería y Tecnología, 2005) (*Visión Artificial*):
 - ✓ Poco uso del conocimiento acerca del contenido de las imágenes

- ✓ Procesamiento guiado por operador humano o algún algoritmo de alto nivel.
- ✓ A la entrada y salida se tienen datos del mismo tipo (imágenes)
- ✓ Codificación/Transmisión/Compresión de imágenes, captura, digitalización, almacenamiento, recuperación, supresión de ruido, realce, segmentación.
- *A Alto Nivel (Interpretación de imágenes)* (Escuela Superior de Ingeniería y Tecnología, 2005)
 - ✓ Se caracteriza por el uso explícito de conocimiento acerca de la escena
 - ✓ Procesamiento guiado por objetivos o fines a cumplir
 - ✓ Empleo de técnicas propias de I.A.
 - ✓ Obtención de conocimiento o datos simbólicos acerca de la escena partiendo de datos del tipo “imagen”.

Procesamiento de imágenes: Mejora de calidad visual para un análisis por un operador humano o bien para acometer procesamientos posteriores en mejores condiciones.

Ilustración 7 Procesamiento de Imágenes

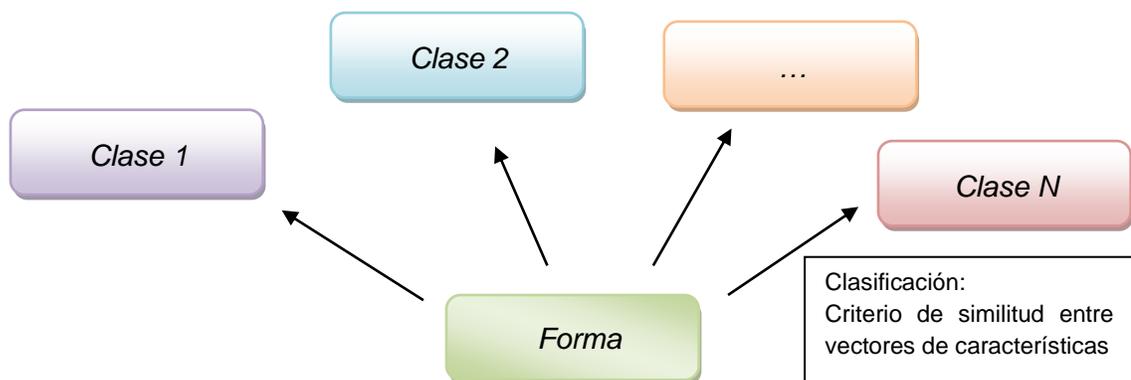


Fuente: (Escuela Superior de Ingeniería y Tecnología, 2005)

Clasificación/Reconocimiento de Formas: Clasificar una forma dada dentro de alguna de las categorías previamente conocidas.

- ✓ Forma - Vector de características
- ✓ Categoría o clase - Vector de características
- ✓ Comparación de “forma incógnita” con cada clase
- ✓ Toma de decisión: Asignar a la clase con la que muestre mayor similitud

Ilustración 8 Clasificación/Reconocimiento de Formas.

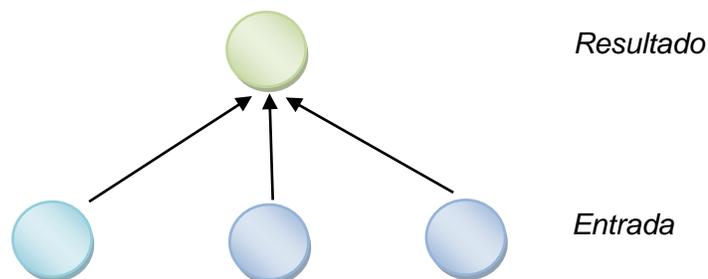


Fuente: (Escuela Superior de Ingeniería y Tecnología, 2005).

REDES NEURONALES (Google Developers, 2020).

Está enfocado en resolver problemas no lineales.

Ilustración 9 Modelo Lineal como un Gráfico.



Fuente: (Google Developers, 2020).

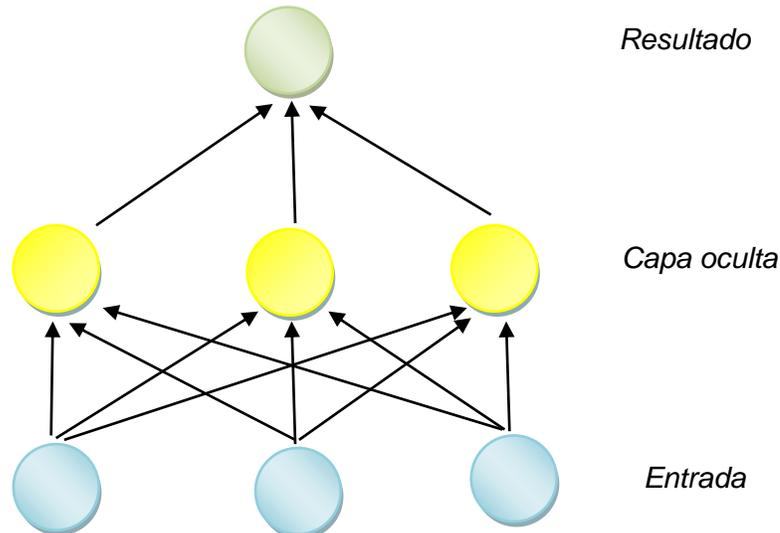
Cada círculo azul representa un atributo de entrada. El círculo verde representa la suma ponderada de las entradas.

¿Cómo se puede alterar este modelo para mejorar su capacidad para resolver problemas no lineales?

Capas ocultas

En el modelo que se muestra en el siguiente grafo, agregamos una "capa oculta" de valores intermedios. Cada nodo amarillo en la capa oculta es una suma ponderada de los valores de los nodos de entrada azul. El resultado es una suma ponderada de los nodos amarillos.

Ilustración 10 Grafo de un Modelo de dos Capas.



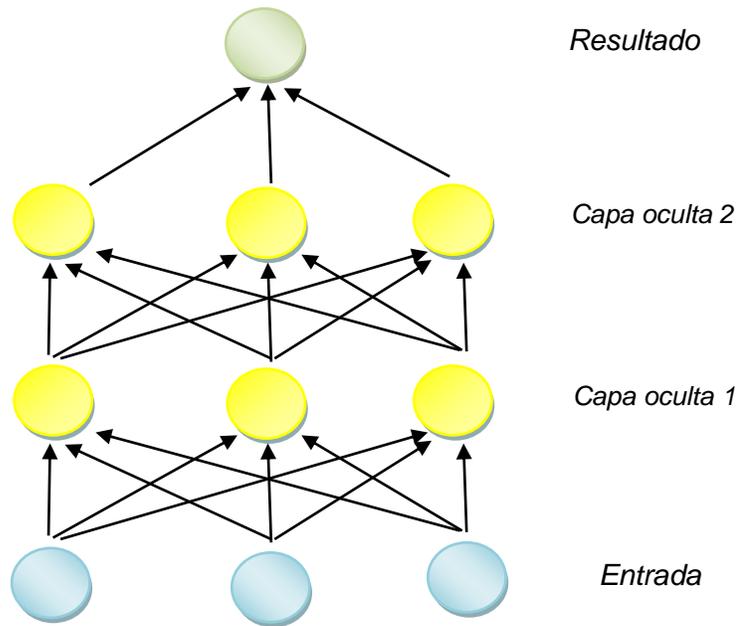
Fuente: (Google Developers, 2020).

Tres círculos azules en fila con la etiqueta "Entrada" conectados por flechas a una fila de círculos amarillos con la etiqueta "Capa oculta" encima de ellos, los cuales, a su vez, están conectados a un círculo verde con la etiqueta "Resultado" en la parte superior.

¿Este modelo es lineal? Sí, su resultado es una combinación lineal de las entradas.

En el modelo que se muestra en el siguiente grafo, agregamos una segunda capa oculta de sumas ponderadas.

Ilustración 11 Grafo de un Modelo de Tres Capas.



Fuente: (Google Developers, 2020).

Tres círculos azules en fila con la etiqueta "Entrada" conectados por flechas a una fila de círculos amarillos con la etiqueta "Capa oculta" encima de ellos, los cuales, a su vez, están conectados a otra fila de círculos amarillos con la etiqueta "Capa oculta 2", que están conectados a un círculo verde con la etiqueta "Resultado" en la parte superior.

Funciones de activación.

Para modelar un problema no lineal, se debe introducir directamente una no linealidad. A cada nodo de la capa oculta se puede aplicar una función no lineal.

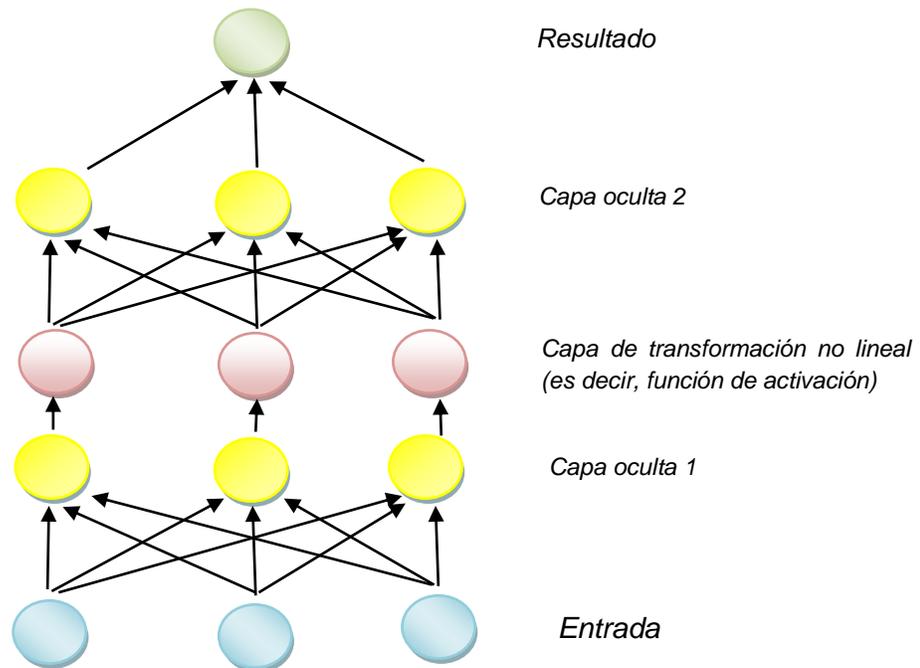
En el modelo que se muestra en el siguiente gráfico, el valor de cada nodo en la Capa oculta 1 se transforma mediante una función no lineal antes de llegar a las sumas ponderadas de la siguiente capa. Esta función no lineal se denomina "función de activación".

Lo mismo que la ilustración 12, excepto que se agregó una fila de círculos rosa con la etiqueta "Capa de transformación no lineal" en medio de las dos capas ocultas.

Ahora que agregamos una función de activación, las capas tienen más impacto. Al apilar no linealidades a no linealidades, se pueden modelar relaciones muy complicadas entre las entradas y los resultados predichos. En resumen, cada capa

aprende una función más compleja y de nivel más alto de las entradas sin procesar.

Ilustración 12 Gráfico del Modelo de Tres Capas con Función de Activación.



Fuente: (Google Developers, 2020).

Funciones de activación comunes.

La siguiente función de activación sigmoide convierte la suma ponderada a un valor entre 0 y 1.

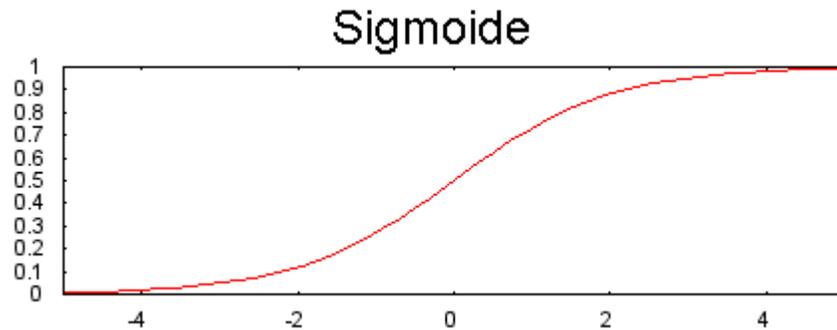
Ecuación 1 Función sigmoide

$$f(x) = \frac{1}{1 + e^{-x}}$$

Fuente: (Google Developers, 2020).

A continuación, se muestra una representación gráfica:

Ilustración 13 Función de activación sigmoide



Fuente: (Google Developers, 2020).

La siguiente función de activación de unidad lineal rectificadora (abreviada como ReLU por sus siglas en inglés) a menudo funciona mejor que una función suave, como la sigmoide, y es mucho más fácil de calcular.

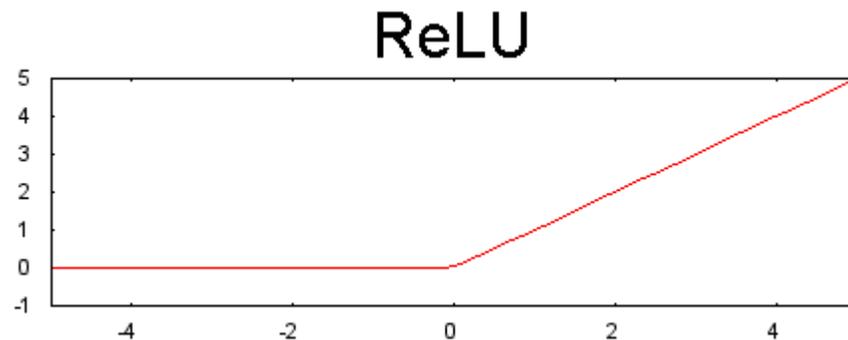
Ecuación 2 Función ReLU

$$f(x) = \max(0, x)$$

Fuente: (Google Developers, 2020).

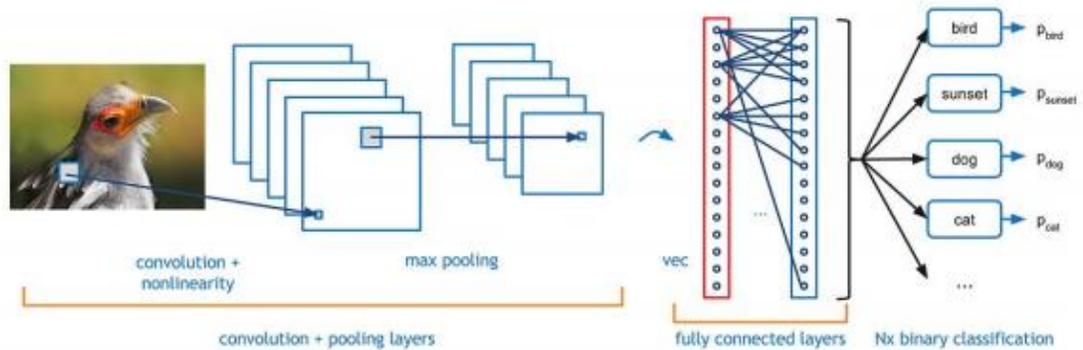
La superioridad de ReLU se basa en las investigaciones empíricas, probablemente debido a que ReLU tiene un rango de capacidad de respuesta más útil. La capacidad de respuesta de una función sigmoide decae relativamente rápido en ambos extremos.

Ilustración 14 Función de activación ReLU.



Fuente: (Google Developers, 2020).

Ilustración 15 Esquema General de una CNN.



Fuente: (Vega Arias, 2019).

FUNCIÓN DE PÉRDIDA

Existen diferentes funciones y su objetivo es realizar una estimación de cómo fue el resultado a partir del valor que se ha obtenido y la etiqueta del ejemplo de entrenamiento que denota el valor esperado. En otras palabras, es una forma de penalizar la desviación entre el resultado esperado y el resultado predicho por la red.

Según (Sancho Peña, 2017) la forma más común es utilizar una función softmax, que tiene la capacidad de convertir sus entradas en una distribución probabilística. Luego, a esta función, se le aplica una función de cross-entropy para calcular la pérdida (los) que determina la penalización.

La función softmax, definida por

$$S(Y_i) = \frac{e^{Y_i}}{\sum_{j=0}^k e^{Y_j}}$$

Es una función exponencial normalizada que convierte el vector unidimensional de la salida en una distribución probabilística para las clases predichas. Luego, la pérdida se calcula usando la función de entropía cruzada, la cual está definida por

$$C(S(Y), L) = - \sum_{i=0}^K L_i \log(S_i)$$

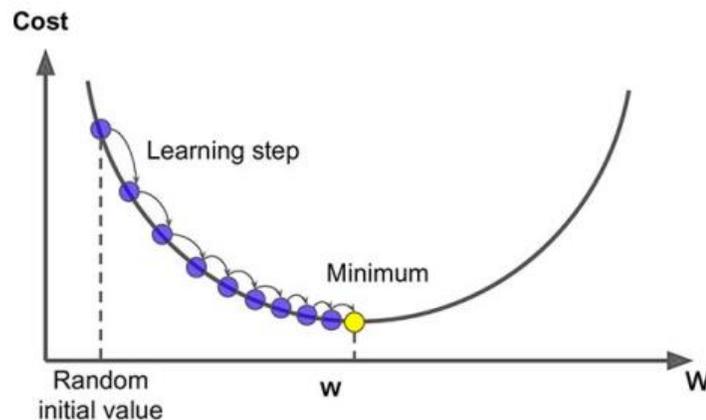
Mide la diferencia entre dos vectores de probabilidades. Donde uno de ellos es un vector de etiquetas L, en el cual todos los valores serán 0 excepto en la posición

del resultado esperado, que será un 1. El otro, es el vector de salida de la función softmax.

MÉTODO DEL GRADIENTE DESCENDENTE:

Es una técnica iterativa de optimización, tiene como objetivo predecir en qué dirección se encuentra la pendiente y modificar el peso para reducir el error, en otras palabras, se dirige cuesta abajo (Pérez Roldán, 2019). Esto se logra calculando la derivada de la función de pérdida para producir el gradiente. La derivada aproxima el grado de cambio de la función (Sancho Peña, 2017).

Ilustración 16 Cambios del peso con el Objetivo de Alcanzar el Mínimo en SGD



Fuente: (Pérez Roldán, 2019).

En una ConvNet, se puede simplificar la red como una función $L = f(X, W)$ donde X es la entrada, W los pesos de las capas, y que tiene como salida una pérdida L . Esta función corresponde a lo que se conoce como el *forward pass*. A partir de ella, se pueden computar los gradientes para cada una de las capas usando *backpropagation* y actualizar los pesos usando dichos gradientes. Se puede definir esto como

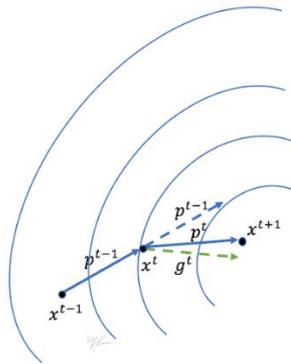
$$W_{t+1} = W_t - \alpha \nabla f(W_t, X)$$

Donde W_t son el conjunto de pesos calculados para la iteración t usando los gradientes calculados y la entrada de la iteración $t-1$. La tasa de aprendizaje α determinará el tamaño del paso en la dirección del gradiente negativo (Sancho Peña, 2017).

DESCENSO POR GRADIENTE ESTOCASTICO (SGD):

Es una modificación del descenso original, su función es calcular el gradiente y actualizar el vector parámetro por cada lote de n muestras de entrenamiento con el fin de acelerar la función de aprendizaje y, por el contrario, no calcular el gradiente y actualizar el vector de parámetros para cada ejemplo de entrenamiento (Pérez Roldán, 2019)

Ilustración 17 Gradiente por Descenso Estocástico.



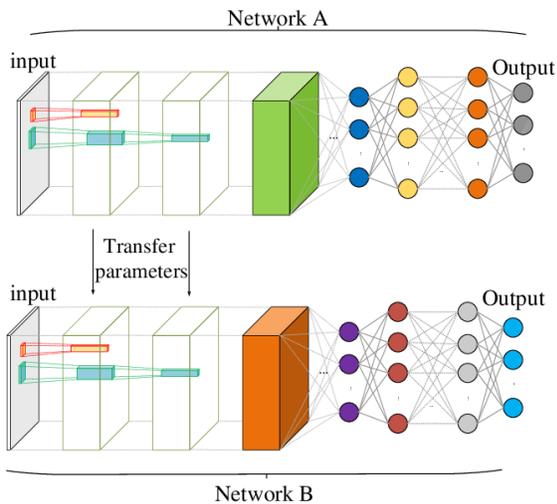
Fuente: (Pérez Roldán, 2019).

TRANSFER LEARNING:

Luego de escoger la arquitectura más adecuada para crear un modelo de red neural profunda debemos definir como inicializaremos sus parámetros; es decir, elegir el valor inicial de todas las constantes que multiplicaran (pesos) o se sumaran (bias) a las variables de entrada de cada capa en nuestra red. Anteriormente, estos parámetros se inicializaban de forma aleatoria dentro de un pequeño rango, con el fin de ayudar de alguna manera el algoritmo encargado de ajustar estos valores durante el entrenamiento. Transfer Learning nace como una solución a este problema, inicializando los parámetros obtenidos después de entrenar la misma arquitectura base con un conjunto grande de imágenes etiquetas donde ya existían diferentes cosas, de esta forma se estaría transfiriendo los conocimientos adquiridos en una tarea más genérica a un caso en particular, y no se partiría desde 0 (Durán, 2020).

Esto es lo que aportan los modelos previamente entrenados sobre el inmenso dataset del proyecto ImageNet, que actualmente cuenta con más de 14 millones de imágenes y casi 22,000 etiquetas distintas. El caso quizá mas conocido es el de la arquitectura ResNet.

Ilustración 18 Transfer Learning.



Fuente: (Lemley, Bazrafkan, & Corcoran, 2017).

MODELO CLIENTE/SERVIDOR (Marini E. , 2012).

Es un modelo de aplicación distribuida en el que las tareas se reparte entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Las aplicaciones Clientes realizan peticiones a una o varias aplicaciones Servidores, que deben encontrarse en ejecución para atender dichas demandas.

Modelo de Servicios: es una vista abstracta que establece las definiciones, reglas y relaciones entre las estructuras relacionadas con la aplicación. Sirve de base para el intercambio de ideas durante el desarrollo lógico de la aplicación y determina cómo será la aplicación resultante:

- **Lógica de Presentación:** es la responsable del control de todos los aspectos relacionados con la interacción entre el usuario y la aplicación. Para llevar a cabo esta tarea de control, es necesario conocer que tipos de usuarios utilizaran la aplicación, que actividades tienen que realizar. Con el fin de independizar la interfaz de usuario de las características propias de los procesos, debemos tener presente que la codificación de las tareas asociadas a esta lógica consiste, principalmente, en la llamada a procesos independientes situados en las otras lógicas, cuya ejecución es totalmente transparente.
- **Lógica de Negocio:** es la lógica de la aplicación que controla la secuencia de acciones y fuerza el cumplimiento de las reglas del negocio propias de cada empresa; además, asegura la integridad de las

transacciones de las operaciones necesarias que haya que realizar para que se cumplan dichas reglas.

- **Lógica de Datos:** aquí entran los procesos encargados del mantenimiento de los datos, de garantizar las reglas de integridad referencias establecidas, así como de la gestión de las transacciones. Estas tareas son realizadas, generalmente, por un sistema de Gestión de Bases de Datos Relacionales.

Ilustración 19 Modelo de Servicios



Fuente: autor en base (Marini E. , 2012).

Modelo Cliente/Servidor multicapa:

La arquitectura cliente/servidor genérico tiene dos tipos de nodos en la red: clientes y servidores. Algunas redes dispones de tres tipos de nodos:

- Clientes que interactúan con los usuarios finales.
- Servidores de aplicación que procesan los datos para los clientes.
- Servidores de la base de datos que almacenan los datos para los servidores de aplicación.

Esta configuración se llama una arquitectura de tres-capas el cual aporta una flexibilidad adicional en la construcción de aplicación cuando estas aumentan su complejidad. Influye tanto en el modelo de aplicación (lógicas de presentación, del

negocio y de datos) como en la distribución de los servicios. El modelo conceptual de una aplicación establece sus definiciones, reglas y relaciones, así como su estructura. Hay partes de la lógica que residen en el cliente, normalmente las que se refieren a la interfaz de usuario, mientras que las de negocio y de datos suelen residir en los servidores, que proporcionan los mecanismos necesarios para el trabajo en entornos multiusuarios.

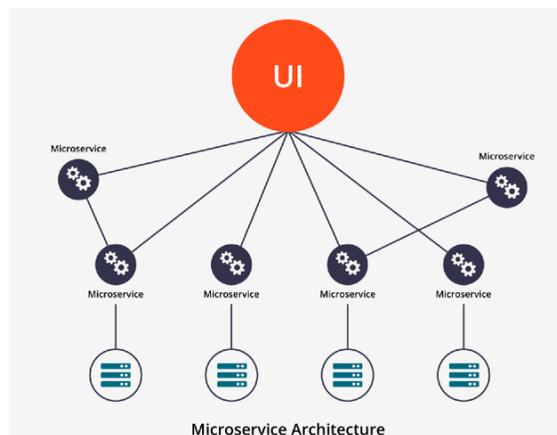
En la actualidad las técnicas y lenguajes de programación de servidores han avanzado de tal forma de permitir desarrollos modulares (mediante la arquitectura MVC: model-control-view), orientación a objetos y existen innumerables frameworks y librerías para simplificar la tarea de los desarrollares de software.

Una de las características principales de este modelo reside en la desconexión total entre la lógica de presentación y la lógica de los datos. Las conexiones que se producen se dan entre las lógicas de presentación y del negocio, y las lógicas del negocio y de datos.

ARQUITECTURA ORIENTADA A MICROSERVICIOS

Los microservicios buscan dividir los sistemas en partes individuales, permitiendo que se puedan tratar y abordar los problemas de manera independiente sin afectar el resto. Mientras que una arquitectura monolítica el software se desarrolla como una única unidad, una arquitectura de microservicios funciona como un conjunto de pequeños servicios que se ejecutan de manera automática e independiente (decide, 2019).

Ilustración 20 Arquitectura de microservicio.



Fuente: (decide, 2019).

La arquitectura de microservicios es un método de desarrollo de aplicaciones software que funciona como un conjunto de pequeños servicios que se ejecutan

de manera independiente y automática, proporcionando una funcionalidad de negocio completa. Dentro de esta, cada microservicio es un código que puede estar en un lenguaje de programación diferente, y que desempeña una función específica. Los microservicios se comunican entre sí a través de APIs, y cuentan con sistemas de almacenamiento propios, lo que evita la sobre carga y caída de la aplicación.

FLASK.

Flask es un framework web del lenguaje Python. Flask proporciona una biblioteca y una colección de códigos que se pueden usar para crear sitios web sin la necesidad de hacer todo desde cero. Debido a sus características simples, flask es más ligero y no depende de muchas librerías externas. En general, el matraz proporciona “Werkzeug” que es útil para recibir solicitudes (url) y responder.

Ilustración 21 Logo de Flask.



Fuente: (Flask, 2010).

Flask es un micro framework web escrito en el lenguaje de programación Python, fue creado por Armin Ronacher en 2004. Flask está licenciado bajo una licencia BSD de tres cláusulas, está diseñado para hacer aplicaciones web de manera rápida y fácil, con la capacidad de mejorar aplicaciones complejas. Comenzó como una simple envoltura alrededor de Werkzeug y Jinja pero, se ha convertido en uno de los marcos de aplicaciones web de Python más populares (Mufid, Al Rasyid, Basofi, Rokhim, & Rochimansyah, 2019).

Ilustración 22 Ejemplo de Script con Flask

```
from flask import Flask, escape, request

app = Flask(__name__)

@app.route('/')
def hello():
    name = request.args.get("name", "World")
    return f'Hello, {escape(name)}!'
```

Fuente: (Mufid, Al Rasyid, Basofi, Rokhim, & Rochimansyah, 2019).

Estructura del proyecto con flask (Grinberg, 2018):

Flask no tiene una estructura estricta definida. Cuando se tienen aplicaciones web pequeñas es muy conveniente almacenarla en un solo archivo de script, aunque a medida que crece la complejidad, trabajar con un solo archivo se vuelve problemático. Flask deja la forma de estructurar la aplicación al desarrollar. Pero, por defecto flask constituye la siguiente estructura.

Ilustración 23 Estructura de archivos con Flask

```
| -flasky
| -app/
|   |-templates/
|   |-static/
|   |-main/
|     |-__init__.py
|     |-errors.py
|     |-forms.py
|     |-views.py
|     |-__init__.py
|     |-email.py
|     |-models.py
| -migrations/
| -tests/
|   |-__init__.py
|   |-test*.py
| -venv/
| -requirements.txt
| -config.py
| -flasky.py
```

Fuente: (Grinberg, 2018).

Esta estructura tiene cuatro carpetas de nivel superior:

- La aplicación Flask está dentro de un paquete genéricamente llamado aplicación.
- La carpeta de migraciones contiene los scripts de migración de la base de datos.
- Las pruebas unitarias están escritas en un paquete de pruebas.
- La carpeta venv contiene el entorno virtual de Python.

También hay algunos archivos nuevos:

- requerimientos.txt enumera las dependencias del paquete para que sea fácil regenerar un Entorno virtual idéntico en una computadora diferente.
- cong.py almacena los ajustes de configuración.
- flasky.py define la instancia de la aplicación Flask y también incluye algunas tareas que ayuda a administrar la aplicación.

FLUTTER

Es un framework de desarrollo de aplicaciones móviles multiplataforma desarrollado por Google. Su primera versión, conocida como “Sky”, fue presentada en 2015 durante el Dart Developer Summit. La primera versión estable fue presentada al público el 4 de diciembre de 2018 durante el evento Flutter Live en Londres.

Ilustración 24 Logo de Flutter



Fuente: (Flutter-Dev, 2011)

Como lenguaje de programación utiliza Dart, también desarrollado por Google (Zazo Millán, 2019), Dart es un lenguaje de programación de código abierto creado por Google en 2011 con la intención de proporcionar una alternativa más moderna a JavaScript. Se trata de un lenguaje especialmente optimizado para la creación de interfaces de usuario.

Ilustración 25 Logo Dart.

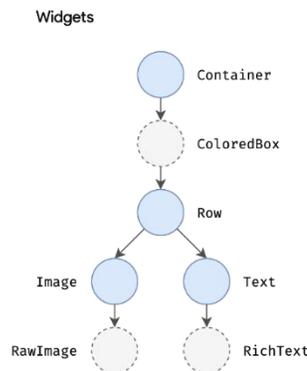


Fuente: (Dart Dev, 2011).

El objetivo de Flutter es permitir a los desarrolladores ofrecer aplicaciones de alto rendimiento que se sientan naturales en diferentes plataformas, aceptando las diferencias en los comportamientos de desplazamiento, tipografía, iconos y más. Una de sus ventajas, es la creación de prototipos e iteración fácil, se puede experimentar cambiando el código y recargándolo a medida que se ejecuta en la aplicación (con recargar en caliente). Además, permite solucionar bloqueos y continuar con la depuración desde donde la aplicación se quedó (Flutter-dex, 2020).

En flutter se construye toda la aplicación usando Dart, incluida la interfaz de usuario. Para ello, se apoya fuertemente en el paradigma de la programación orientada a objetos, más concretamente en la composición y herencia de los widgets, los cuales componen la interfaz. Flutter construye los widgets a nivel de aplicación, lo que hace que el desarrollador pueda personalizarlos y extenderlos de manera sencilla, consiguiendo una mayor libertad en el diseño de las aplicaciones (Vázquez Rodríguez, 2019). Los widgets forman una jerarquía basada en la composición, es decir la herencia de las propiedades del padre, puede ser: un elemento estructural (como un botón o menú), un elemento de estilo (como una fuente o esquema de color), un aspecto del diseño (como el relleno).

Ilustración 26 Jerarquía de Widgets en Flutter



Fuente: (Flutter-dex, 2020).

8. DISEÑO METODOLÓGICO

8.1 INTRODUCCIÓN

El proyecto tuvo una duración de 1 año (32) semanas divididas en cuatro fases esenciales que agruparon el conjunto de actividades definidas en el anteproyecto (ver tabla 1).

Se plantearon 20 actividades, las tres primeras corresponden al objetivo específico número uno, las 9 siguientes al objetivo específico número dos. Luego, las 4 que continúan al número 3 y finalmente, las 4tro últimas al objetivo número cuatro.

Fase I: con relación al primero objetivo específico, se buscó definir la tecnología y arquitectura para la realización de un modelo de clasificación adecuado para la identificación de aves, esto se hizo a través de búsqueda, revisión bibliográfica y la construcción de un cuadro que contrastó las opciones obtenidas de los documentos analizados.

Inicialmente, se realizó la definición de palabras claves sobre clasificación de aves con visión artificial, esto permitió una búsqueda objetiva y reciente (con un rango de 10 años) en base de datos científicas como IEE Xplorer, Springer, entre otras. Seguidamente, se extrajo información relevante como: librerías utilizadas, arquitectura de la red neuronal, dataset utilizado y tipo de desarrollo, esto dio forma al cuadro comparativo.

Además, se realizó una búsqueda en repositorios de GitHub, con el fin de obtener más información y aumentar las opciones para la realización del contraste de tecnologías. A partir de lo anterior, se hicieron pruebas funcionales para la definición de la arquitectura utilizada.

Fase II: en relación al segundo objetivo específico, se abarcó el diseño de la aplicación móvil. En primera instancia, se levantaron los requerimientos funcionales y no funcionales que tiene el sistema, por medio de entrevistas a ornitólogos y habitantes de la zona metropolitana de Bucaramanga. En segundo lugar, se hizo una revisión de aplicaciones que identificaban cualquier objetivo, persona, animal o cosa que complementó los requerimientos ya definidos. Seguidamente, se realizó el flujo de trabajo de Deep Learning, especificando las entradas y salidas del algoritmo, que dio paso a la definición de la arquitectura de

la solución que permitió la interacción entre la aplicación y el modelo de clasificación dando como resultado el diagrama de despliegue.

A partir de lo anterior se elaboró el diagrama de casos de uso, que permitió el modelado y definición de vocabulario de la base de datos (modelo entidad-relación, relacional y diagrama de clases). Luego, se definieron las interfaces de la aplicación móvil obteniendo el diagrama de flujo de interfaces. Finalmente, se realizó una búsqueda de los datasets obtenidos en la primera fase.

Fase III: En referencia al tercer objetivo específico, se desarrolló e implementó el modelo de clasificación y la aplicación móvil diseñada. Inicialmente, se creó un dataset en base a un conjunto de especies representativas de la zona definido por un usuario experto. Seguidamente, se hizo un análisis exploratorio sobre las imágenes recopiladas para identificar patrones entre las especies.

Después, se desarrolló el código fuente del modelo clasificador en base a la arquitectura definida en la primera fase, se entrenó y se obtuvieron las métricas. Luego, se desarrolló el código fuente de la aplicación móvil basado en los diseños realizados en la fase II y se implementó el modelo entrenado. Por otro lado, se desarrolló el sistema web para la administración de la base de datos.

Fase IV: En referencia al cuarto objetivo específico, fue la fase final del proyecto y su propósito fue evaluar el funcionamiento y aceptación de la aplicación desarrollada. En primer lugar, se realizó el plan de pruebas de funcionalidad y el de aceptación. En segundo lugar, se ejecutaron los planes y se documentaron los resultados obtenidos.

Finalmente, se documentaron los posibles trabajos futuros que mejoren y expandan el proyecto desarrollado.

8.2. MODELO DE CICLO DE VIDA DEL SOFTWARE Y MODELO DE IA.

8.2.1. MODELO DEL CICLO DE VIDA DEL SOFTWARE

El presente proyecto se desarrolló bajo el modelo del ciclo de vida de prototipado, en donde las actividades se ejecutan como en el flujo evolutivo siguiendo los pasos:

1. Definir si el prototipo va a evolucionar o no.
2. Diseño rápido.
3. Hacer prototipo.
4. Evaluar.

5. Refinar prototipo.
6. Decidir si el prototipo muere o es producto.

Ilustración 27 Modelo Ciclo de Vida Prototipados.

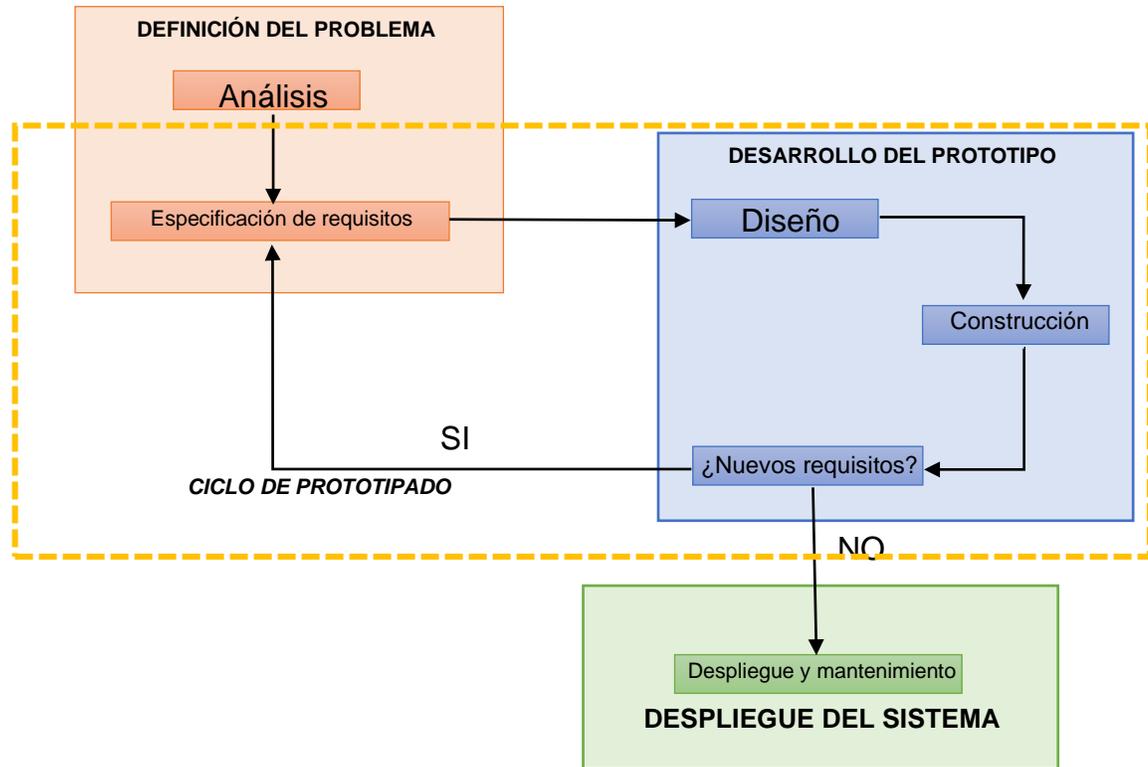


Fuente: Autor en base (Entre líneas, 2008).

Este modelo fue escogido por la facilidad que provee para validar y verificar requerimientos. Además, de ofrecer la oportunidad de formar programadores sin experiencia en la tecnología, ya que la tecnología utilizada no fue conocida al inicio del proyecto. Una de las principales ventajas, es que permite realizar pruebas eficientes en cada prototipo desarrollado para evaluar con el cliente y hacer ajustes de manera inmediata.

Este modelo es complejo desde el punto de que puede caer en un ciclo continuo, es decir, no saber cuándo se debe parar, lo que tendría un costo excesivo a los recursos de tiempo, dinero y personal del proyecto. Al final, termina como el modelo del ciclo de vida en cascada, porque se tendría el producto final con todas las funcionalidades.

Ilustración 28 Modelo detallado prototipados



Fuente: Autor en base (Junta de Andalucía, 2017).

El proyecto fue dividido en diferentes prototipos agrupando las funcionalidades dependiendo de su similitud y relación con los diferentes casos de uso, los cuales se encuentran detallados en la tabla 3. Listado de casos de uso. Por otro lado, dado la plataforma, el software web fue desarrollado en flask, la aplicación móvil con Flutter y el sistema de clasificación en Python.

El flujo de conexión es la siguiente, la aplicación web dada para el administrador, también algunas funcionalidades para el usuario y el ornitólogo, está conectada al sistema clasificador que, a su vez, está en conexión con la aplicación móvil de donde se enviará la imagen tomada se consulta en donde el sistema clasificador regresará el nombre del ave identificada para dar paso a la consulta de la información a la base de datos para mostrar la información en las interfaces.

Tabla 3 Prototipos por Caso de Uso

LISTADO DE CASOS DE USO			
Numero	Nombre del Caso de Uso	Actor (Es) que intervienen	Prototipo
ECU001	Crud Usuario	Administrador, Usuario, Ornitólogo	1
ECU002	Crud Ave	Administrador	1
ECU003	Crud Ubicación	-	1
ECU004	Clasificar Ave	Usuario, Ornitólogo.	2
ECU005	Consultar listado de aves identificadas	Usuario, Ornitólogo.	2
ECU006	Recuperar Contraseña	Administrador, Usuario, ornitólogo.	3
ECU007	Consultar clasificador	-	2

Fuente: Autor.

Aunque en tabla 3 se encuentran los prototipos propuestos, en el desarrollo se generaron nuevos prototipos, con el fin de mejorar la experiencia de usuario y mejorar problemas de conexión.

Tabla 4 Prototipos Complementarios.

LISTADO DE CASOS DE USO			
Numero	Nombre de la funcionalidad mejorada.	Actor (Es) que intervienen	Prototipo
-	Información del ave: se añadieron cards para mejorar la visibilidad de la información de las aves que se pueden clasificar en avedex.	Usuario, Ornitólogo.	4
-	Clasificar Ave: se añadió un popup de información, para indicarle al usuario que debía realizar en el mapa donde debía escoger la ubicación donde había visto el ave.	Usuario, Ornitólogo.	5
-	Aplicación en general: se añadió el nombre y el logo a la aplicación.	-	6

Fuente: Autor.

8.2.2. MODELO DEL CICLO DE VIDA DEL MODELO MACHINE LEARNING.

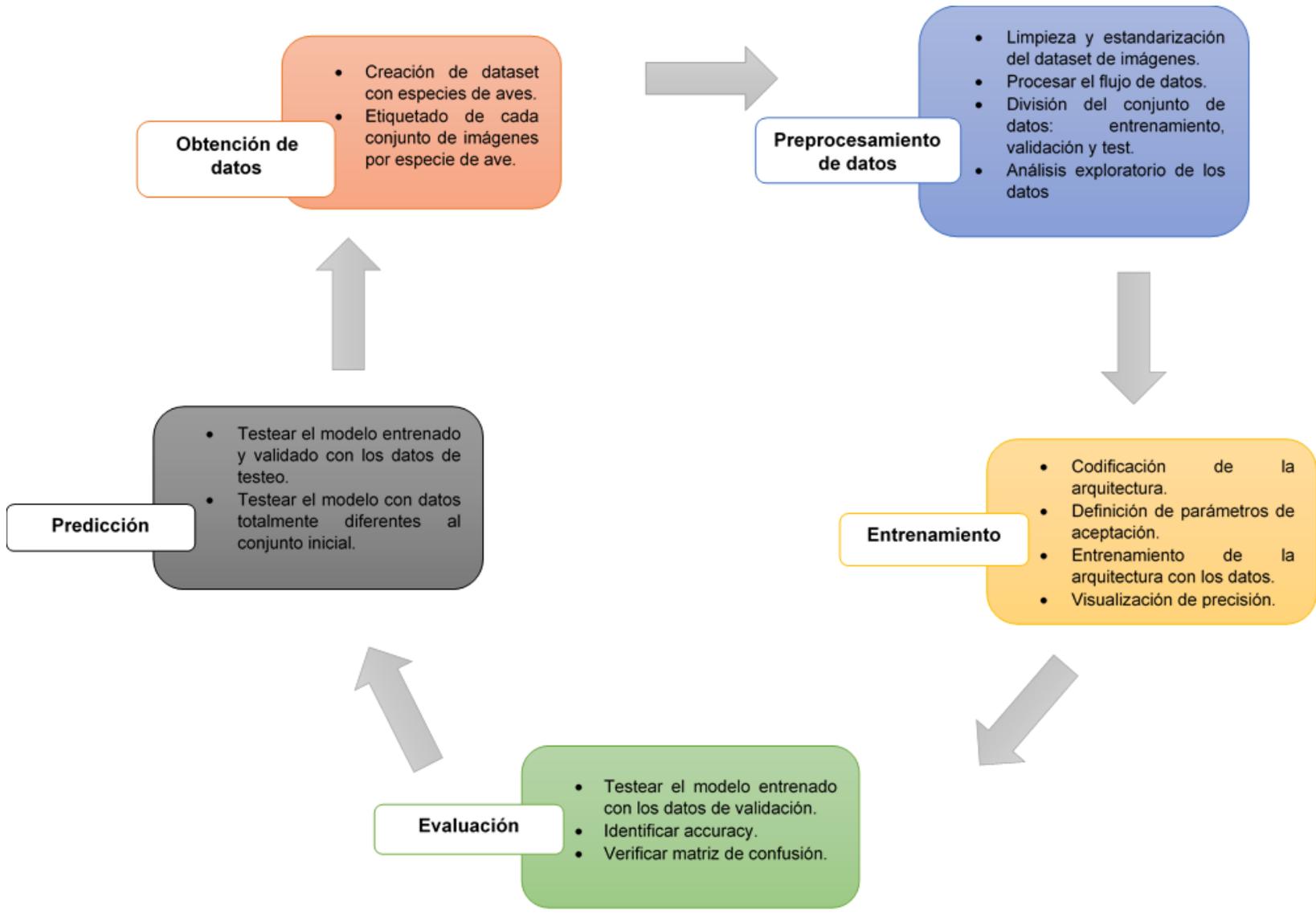
El sistema clasificador posee su propio modelado para identificar en qué momento en el tiempo se tendrá un buen modelo. Por lo tanto, el modelado propuesto para este proyecto tiene diferentes etapas que tienen como fin determinar un alcance y no quedar iterando en el proceso de la obtención de un nuevo modelo. Es importante mencionar, que cada proyecto de inteligencia artificial tiene sus propios requerimientos y que, por lo tanto, se debe asumir un punto propio de llegada y los requerimientos necesarios.

En la ilustración 28, se encuentra especificado el flujo de trabajo que se aplicará para el desarrollo del modelo, esta dividido por las siguientes fases.

1. Obtención de datos.
2. Preprocesamiento de datos.
3. Entrenamiento.
4. Evaluación.
5. Predicción.

Para determinar si es un modelo aceptable, se hará énfasis en la fase número 3, en donde con ayuda de los parámetros de evaluación del entrenamiento nos aseguraremos de tener un modelo con una precisión aceptable. Además, se hará énfasis en la fase número 5, en donde por medio de la obtención de nuevos datos se harán predicciones para evaluar el rendimiento del modelo clasificador.

Ilustración 29 ciclo de vida modelo Machine Learning.



Fuente: Autor.

8.3 ANÁLISIS Y DISEÑO

8.3.1 INFORMACIÓN DEL DOMINIO DEL PROBLEMA

Dado el problema planteado se buscó desarrollar una aplicación móvil y un sistema web que contuviera un modelo de clasificación que permitiera realizar predicciones que den paso a la consulta de las aves en la base de datos. Además, en el sistema web se tiene la administración de la base de datos en un motor de SQL.

8.3.1.1 Glosario de Términos

CRUD: crear, leer, actualizar y borrar de la base de datos. (Create, Read, Update and Delete)

8.3.1.2 Descripción de los Actores

Se identificaron 3 actores relevantes del sistema, que se describen a continuación.

Tabla 5 Descripción de Actores

<id>001	<i><Usuario No Especializado></i>
[Versión]	<i><0100> (<01/09/20>)</i>
[Dependencias]	<i>No tiene dependencias.</i>
Descripción	<i>Este actor representa a todos los habitantes de la zona que no tienen un conocimiento empírico o profesional sobre la identificación de aves a simple vista. Es quién se registrará en la aplicación móvil, podrá editar perfil, iniciar sesión, identificar un ave y consultar lista de aves identificadas.</i>
Comentarios	<i>Sin comentarios.</i>
<id>002	<i><Ornitólogo></i>
[Versión]	<i><0100> (<01/09/20>)</i>
[Dependencias]	<i>No tiene dependencias.</i>

Descripción	<i>Este actor representa a todos los usuarios que tienen conocimiento profesional y/o empírico para identificar aves a simple. Es quién se registrará en la aplicación móvil, podrá editar perfil, iniciar sesión, identificar un ave y consultar lista de aves identificadas.</i>
Comentarios	<i>En trabajos futuros tendrán la opción de realizar una clasificación manual por solicitudes con el fin de mejorar el modelo clasificador que se entregará en este proyecto.</i>
<id>003	<Administrador>
[Versión]	<0100> (<01/09/20>)
[Dependencias]	<i>No tiene dependencias.</i>
Descripción	<i>Es quien administra la base de datos del sistema desde el sistema web, tendrá la opción de realizar el CRUD de aves, de usuarios y recuperar contraseña.</i>
Comentarios	<i>Sin comentarios.</i>

Fuente: Autor.

8.3.2 CATÁLOGO DE REQUISITOS DEL SISTEMA A DESARROLLAR

8.3.2.1 Requisitos Generales del Sistema

Tabla 6 Requisitos Generales del Sistema

<RG>001	<i>Gestionar las identificaciones de aves</i>
[Versión]	0100 (01/09/20)
Descripción	<i>El sistema deberá permitir tomar y escoger una foto desde el dispositivo móvil y mostrar la información general del ave que estaba en la foto. Además, mostrar una lista de fotos con las aves ya identificadas y un mapa donde se marque la ubicación del avistamiento del ave.</i>
Requisitos hijos	<i>No tiene</i>
[Importancia]	<i>Muy importante</i>

[Prioridad]	<i>Alta</i>
Comentarios	<i><comentarios adicionales sobre el requisito general></i>
<RG>002	<i>Gestionar los usuarios</i>
[Versión]	<i>0100 (01/09/20)</i>
Descripción	<i>El sistema deberá permitir registrar, actualizar o eliminar información de los usuarios.</i>
Requisitos hijos	<i>No tiene</i>
[Importancia]	<i>Muy importante</i>
[Prioridad]	<i>Alta</i>
Comentarios	<i>En la aplicación móvil de vista al usuario general, solo se le permitirá registrar y actualizar su información.</i>
<RG>003	<i>Gestionar aves</i>
[Versión]	<i>0100 (01/09/20)</i>
Descripción	<i>El sistema deberá permitir registrar, actualizar o eliminar información de las aves, incluyendo la familia y fotos.</i>
Requisitos hijos	<i>No tiene</i>
[Importancia]	<i>Muy importante</i>
[Prioridad]	<i>Alta</i>
Comentarios	<i>En la aplicación móvil de vista al usuario general, solo se le permitirá consultar la información del ave con alguna identificación.</i>

Fuente: Autor.

8.3.2.2 Casos de uso del Sistema

8.2.2.2.1 Diagramas de Casos de Uso del Sistema

Se identificaron 10 casos de uso, los cuales se encuentran descritos en la tabla 1.

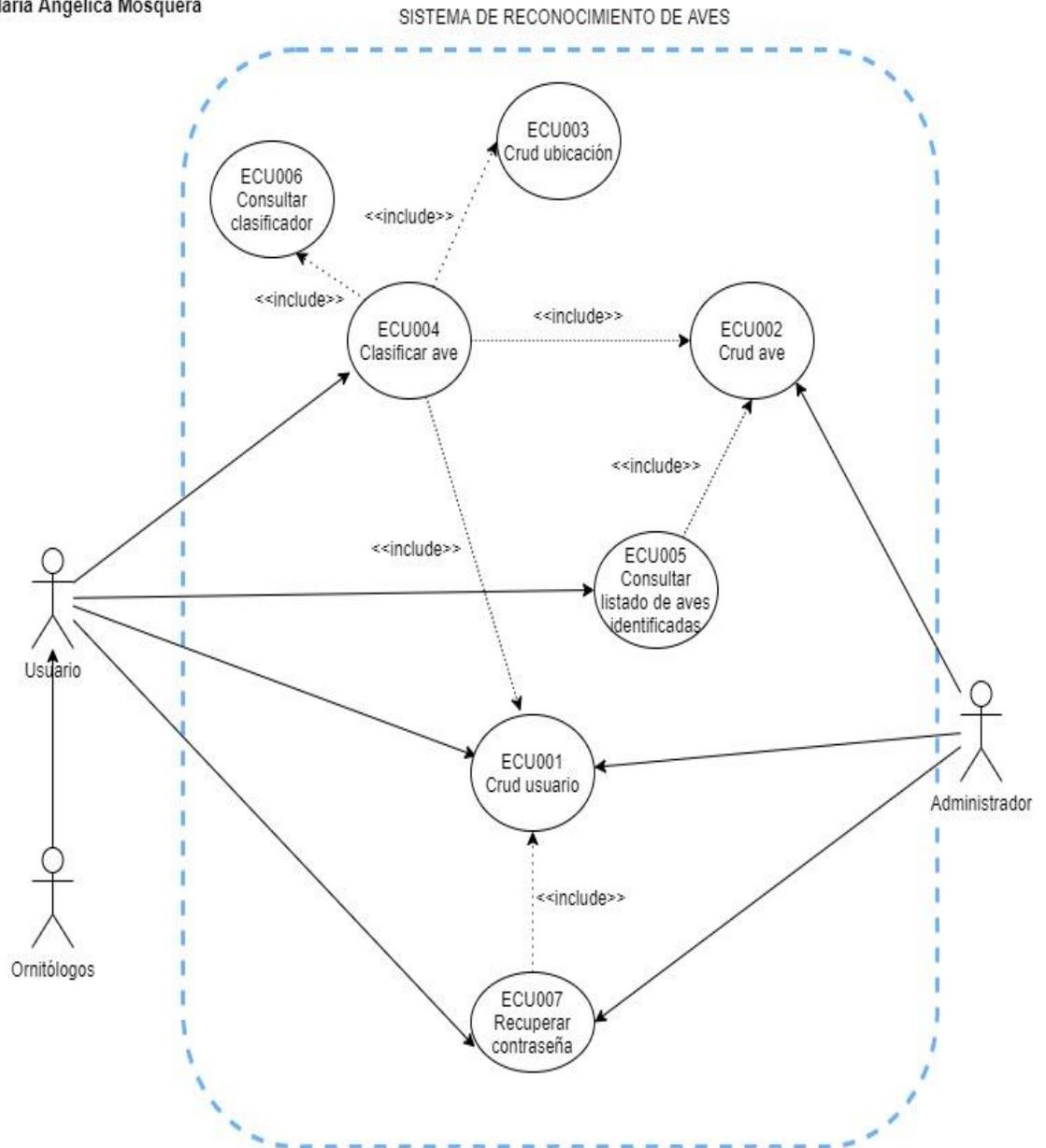
Tabla 7 Descripción de Casos de Usos del Sistema

ID	NOMBRE	ACTORES INVOLUCRADOS	INCLUDE	EXTENDS
ECU001	Crud Usuario	Administrador, Ornitólogo	Usuario, -	-
ECU002	Crud Ave	Administrador	-	-
ECU003	Crud Ubicación	-	-	-
ECU004	Clasificar Ave	Usuario, Ornitólogo.	ECU001, ECU002, ECU003, ECU007	-
ECU005	Consultar listado de aves identificadas	Usuario, Ornitólogo.	ECU002	-
ECU006	Recuperar Contraseña	Administrador, ornitólogo.	Usuario, ECU001	-
ECU007	Consultar clasificador	-	-	-

Fuente: Autor.

Ilustración 30 Diagrama de Casos de Uso

Name: AveDex
Package: Use Case Model
Version: 1.0
Author: María Angélica Mosquera



Fuente: Autor.

8.2.2.2.2 Especificación de Casos de Uso del Sistema

Tabla 8 Especificación de Casos de Uso del Sistema

<CU>001-1	<i>CRUD Usuario (Crear usuario) o (Registro Usuario)</i>	
[Versión]	1.0 (01/09/20)	
[Dependencias]	<ul style="list-style-type: none"> <i>RG-002 Gestionar usuarios del sistema</i> 	
Escenario	<i>El administrador crea un nuevo usuario o un usuario general/especializado se registra en la aplicación móvil.</i>	
Precondición	<i>El usuario administrador ha creado un usuario padre y se ha logueado en el sistema.</i>	
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando <i>el usuario administrador o un usuario general/especializado crea un nuevo usuario desde el sistema principal</i>	
Secuencia Normal	Paso	Acción
	1	El <i>administrador o usuario general/especializado</i> solicita al sistema comenzar el proceso de añadir (registrar) un nuevo <i>usuario</i> .
	2	El sistema solicita que ingrese los datos esenciales del usuario a registrar.
	3	El <i>administrador o usuario general/especializado</i> proporcional al sistema los datos esenciales del <i>usuario</i> a registrar.
	4	El <i>administrador o usuario general/especializado</i> pulsa el botón guardar información o registrar
	5	Si alguno de los campos se encuentra vacío,
	5.1	<i>El sistema señalará los campos vacíos y se regresará al paso 3 de esta secuencia.</i>
6	El sistema validará la información ingresada e informa de que el registro se ha realizado correctamente.	
Postcondición	<i>El sistema ha registrado el nuevo usuario.</i>	
Excepciones	Paso	Acción
	2	<i>Si el usuario a registrar ya se encuentra registrado en el sistema</i>
	E.1	<i>El sistema informa de la situación que impide realizar el registro.</i>
E.2	<i>Se cancela el caso de uso.</i>	
[Importancia]	<i>Alto</i>	
[Prioridad]	<i>Alto</i>	

Comentarios	<i>Sin comentarios</i>	
<CU>001-2	<i>CRUD Usuario (Editar usuario/Editar Perfil)</i>	
[Versión]	<i>1.0 (01/09/20)</i>	
[Dependencias]	<ul style="list-style-type: none"> <i>RG-002 Gestionar usuarios del sistema</i> 	
Escenario	<i>Se necesitan actualizar los datos de un usuario registrado en el sistema</i>	
Precondición	<i>El usuario administrador ha creado un usuario padre, un usuario general/especializado se ha logueado en el sistema y el usuario a editar existe en el sistema (CU001-1).</i>	
Descripción	<i>El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario administrador o un usuario general/especializado edite un usuario desde el sistema principal o la app.</i>	
Secuencia Normal	Paso	Acción
	1	<i>El administrador o un usuario general/especializado solicita al sistema comenzar el proceso de editar un usuario (editar perfil)</i>
	2	<i>El sistema carga los datos previamente guardados en el sistema</i>
	3	<i>El sistema solicita que ingrese los datos actualizados del usuario.</i>
	4	<i>El administrador o un usuario general/especializado proporcional al sistema los datos actualizados del usuario.</i>
	5	<i>El administrador o un usuario general/especializado pulsa el botón actualizar información</i>
	6	<i>Si alguno de los campos se encuentra vacío,</i>
	6.1	<i>El sistema señalará los campos vacíos y se regresará al paso 3 de esta secuencia.</i>
7	<i>El sistema validará la información ingresada e informa de que la actualización se ha realizado correctamente.</i>	
Postcondición	<i>El sistema ha editado un usuario.</i>	
Excepciones	Paso	Acción
	2	<i>Si el usuario a editar no se encuentra registrado en el sistema</i>
	E.1	<i>El sistema informa de la situación que impide realizar la edición.</i>
	E.2	<i>El sistema recomienda crear un usuario (CU001-1)</i>
	E.3	<i>Se cancela el caso de uso.</i>
[Importancia]	<i>Alto</i>	
[Prioridad]	<i>Alto</i>	

Comentarios	<i>Sin comentarios</i>	
<CU>001-3	<i>CRUD Usuario (Eliminar usuario)</i>	
[Versión]	<i>1.0 (01/09/20)</i>	
[Dependencias]	<ul style="list-style-type: none"> <i>RG-002 Gestionar usuarios del sistema</i> 	
Escenario	<i>Se necesita eliminar un usuario registrado en el sistema</i>	
Precondición	<i>El usuario administrador ha creado un usuario padre, se ha logueado en el sistema y el usuario a eliminar existe en el sistema (CU001-1).</i>	
Descripción	<i>El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario administrador elimine un usuario desde el sistema principal</i>	
Secuencia Normal	Paso	Acción
	1	El <i>administrador</i> solicita al sistema comenzar el proceso de eliminar un <i>usuario</i> .
	2	El sistema carga los datos previamente guardados en el sistema.
	3	El sistema pregunta al administrador si está seguro de eliminar ese usuario.
	4	El <i>administrador</i> proporciona al sistema la confirmación.
	5	El sistema validará la información ingresada e informa de que se ha eliminado al usuario correctamente.
Postcondición	<i>El sistema ha eliminado un usuario.</i>	
Excepciones	Paso	Acción
	2	<i>Si el usuario a eliminar no se encuentra registrado en el sistema</i>
	E.1	<i>El sistema informa de la situación que impide eliminar.</i>
	E.2	<i>El sistema recomienda crear un usuario (CU001-1)</i>
	E.3	<i>Se cancela el caso de uso.</i>
[Importancia]	<i>Alto</i>	
[Prioridad]	<i>Alto</i>	
Comentarios	<i>Sin comentarios</i>	
<CU>001-4	<i>CRUD Usuario (Consultar usuario)</i>	
[Versión]	<i>1.0 (01/09/20)</i>	
[Dependencias]	<ul style="list-style-type: none"> <i>RG-002 Gestionar usuarios del sistema</i> 	
Escenario	<i>Se necesitan mostrar los datos de un usuario registrado en el sistema</i>	

Precondición	<i>El usuario administrador ha creado un usuario padre, o un usuario general/especializado se ha logueado en el sistema y el usuario a consultar existe en el sistema (CU001-1).</i>	
Descripción	<i>El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario administrador o un usuario general/especializado consulte un usuario desde el sistema principal</i>	
Secuencia Normal	Paso	Acción
	1	El <i>administrador</i> o un <i>usuario general/especializado</i> solicita al sistema comenzar el proceso de ver un <i>usuario</i> .
	2	En caso de que se haga desde la app el sistema carga los datos previamente guardados en el sistema como perfil. En caso de que se haga desde el sistema web el sistema carga los datos previamente guardados en el sistema como una tabla.
Postcondición	<i>El sistema ha consultado un usuario.</i>	
Excepciones	Paso	Acción
	2	Si el <i>usuario a consultar</i> no se encuentra registrado en el sistema
	E.1	El sistema informa de la situación que impide realizar la consulta.
	E.2	El sistema recomienda crear un usuario (CU001-1)
	E.3	Se cancela el caso de uso.
[Importancia]	<i>Alto</i>	
[Prioridad]	<i>Alto</i>	
Comentarios	<i>Sin comentarios</i>	
<CU>002-1	<i>CRUD ave (Crear ave)</i>	
[Versión]	<i>1.0 (01/09/20)</i>	
[Dependencias]	<ul style="list-style-type: none"> <i>RG-003 Gestionar aves</i> 	
Escenario	<i>El administrador crea una nueva ave en el sistema principal.</i>	
Precondición	<i>El usuario administrador ha creado un usuario padre y se ha logueado en el sistema.</i>	
Descripción	<i>El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario administrador crea una nueva ave desde el sistema principal.</i>	
Secuencia Normal	Paso	Acción
	1	El <i>administrador</i> solicita al sistema comenzar el proceso de añadir un ave.
	2	El sistema solicita que ingrese los datos esenciales del ave a registrar, incluyendo la familia y las fotos.
	3	El <i>administrador</i> proporciona al sistema los datos esenciales del ave a registrar.

	4	El <i>administrador</i> pulsa el botón guardar información.	
	5	Si alguno de los campos se encuentra vacío,	
	5.1	El sistema señalará los campos vacíos y se regresará al paso 3 de esta secuencia.	
	6	El sistema validará la información ingresada e informa de que el registro se ha realizado correctamente.	
Postcondición	El sistema ha registrado la nueva ave.		
Excepciones	Paso	Acción	
	2	Si el ave a registrar ya se encuentra registrado en el sistema	
	E.1	El sistema informa de la situación que impide realizar el registro.	
	E.2	Se cancela el caso de uso.	
[Importancia]	Alto		
[Prioridad]	Alto		
Comentarios	Sin comentarios		
<CU>002-2	CRUD ave (Editar ave)		
[Versión]	1.0 (01/09/20)		
[Dependencias]	<ul style="list-style-type: none"> RG-003 Gestionar aves 		
Escenario	Se necesitan actualizar los datos de un ave registrado en el sistema		
Precondición	El usuario administrador ha creado un usuario padre, se ha logueado en el sistema y el ave a editar existe en el sistema (CU002-1).		
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario administrador edite un ave desde el sistema principal.		
Secuencia Normal	Paso	Acción	
	1	El <i>administrador</i> solicita al sistema comenzar el proceso de editar un ave.	
	2	El sistema carga los datos previamente guardados en el sistema.	
	3	El sistema solicita que ingrese los datos actualizados del ave.	
	4	El <i>administrador</i> proporciona al sistema los datos actualizados del ave.	
	5	El <i>administrador</i> pulsa el botón actualizar información.	
	6	Si alguno de los campos se encuentra vacío,	
6.1	El sistema señalará los campos vacíos y se regresará al paso 3 de esta		

			<i>secuencia.</i>
	7	El sistema validará la información ingresada e informa de que la actualización se ha realizado correctamente.	
Postcondición	<i>El sistema ha actualizado un ave.</i>		
Excepciones	Paso	Acción	
	2	<i>Si el ave a editar no se encuentra registrado en el sistema</i>	
	E.1	<i>El sistema informa de la situación que impide realizar la edición.</i>	
	E.2	<i>El sistema recomienda crear un ave (CU002-1)</i>	
	E.3	<i>Se cancela el caso de uso.</i>	
[Importancia]	<i>Alto</i>		
[Prioridad]	<i>Alto</i>		
Comentarios	<i>Sin comentarios</i>		
<CU>002-3	<i>CRUD Ave (Eliminar ave)</i>		
[Versión]	<i>1.0 (01/09/20)</i>		
[Dependencias]	<ul style="list-style-type: none"> <i>RG-003 Gestionar aves</i> 		
Escenario	<i>Se necesita eliminar un ave registrado en el sistema</i>		
Precondición	<i>El usuario administrador ha creado un usuario padre, se ha logueado en el sistema y el usuario a eliminar existe en el sistema (CU002-1).</i>		
Descripción	<i>El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario administrador elimine un ave desde el sistema principal</i>		
Secuencia Normal	Paso	Acción	
	1	El <i>administrador</i> solicita al sistema comenzar el proceso de eliminar un <i>ave</i> .	
	2	El sistema carga los datos previamente guardados en el sistema.	
	3	El sistema pregunta al administrador si está seguro de eliminar esa <i>ave</i> .	
	4	El <i>administrador</i> proporcional al sistema la confirmación.	
	5	El sistema validará la información ingresada e informa de que se ha eliminado el <i>ave</i> correctamente.	
Postcondición	<i>El sistema ha eliminado un ave.</i>		
Excepciones	Paso	Acción	
	2	<i>Si el ave a eliminar no se encuentra registrado en el sistema</i>	

	E.1	El sistema informa de la situación que impide eliminar.
	E.2	El sistema recomienda crear un ave (CU002-1)
	E.3	Se cancela el caso de uso.
[Importancia]	Alto	
[Prioridad]	Alto	
Comentarios	Sin comentarios	
<CU>002-4	CRUD Ave (Consultar ave)	
[Versión]	1.0 (01/09/20)	
[Dependencias]	<ul style="list-style-type: none"> RG-003 Gestionar aves 	
Escenario	Se necesitan mostrar los datos de un ave registrado en el sistema	
Precondición	El usuario administrador ha creado un usuario padre, o un usuario general/especializado se ha logueado en el sistema y el ave a consultar existe en el sistema (CU001-1).	
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario administrador o un usuario general/especializado consulte un ave desde el sistema principal o la app.	
Secuencia Normal	Paso	Acción
	1	El administrador o un usuario general/especializado solicita al sistema comenzar el proceso de ver la información de un ave.
	2	En caso de que se haga desde la app el sistema carga los datos previamente guardados en el sistema como interfaz de información del ave. En caso de que se haga desde el sistema web el sistema carga los datos previamente guardados en el sistema como una tabla.
Postcondición	El sistema ha consultado un ave.	
Excepciones	Paso	Acción
	2	Si el ave a consultar no se encuentra registrado en el sistema
	E.1	El sistema informa de la situación que impide realizar la consulta.
	E.2	El sistema recomienda crear un ave (CU002-1)
	E.3	Se cancela el caso de uso.
[Importancia]	Alto	
[Prioridad]	Alto	
Comentarios	Sin comentarios	

<CU>003-1	<i>CRUD Ubicación (Crear ubicación)</i>		
[Versión]	1.0 (01/09/20)		
[Dependencias]	<ul style="list-style-type: none"> RG-001 Gestionar las identificaciones de aves 		
Escenario	Se está identificando un ave.		
Precondición	Que se presenten algunos de los escenarios.		
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso el usuario general/especializado está clasificando un ave por foto de la cámara o de la galería.		
Secuencia Normal	Paso	Acción	
	1	El sistema recibe la ubicación (latitud, longitud) del usuario al momento de realizar la clasificación.	
	2	El sistema validará la información capturada o ingresada e informa de que el registro se ha realizado correctamente.	
Postcondición	El sistema ha registrado la ubicación		
Excepciones Postcondición	Paso	Excepciones	
	1	Si el móvil no tiene encendido el GPS	
	E.1	El sistema informa de la situación que impide realizar el registro y pide que se encienda el GPS	
	E.2	Se cancela el caso de uso.	
[Importancia]	Alto		
[Prioridad]	Alto		
Comentarios	Sin comentarios		
<CU>003-2	<i>CRUD Ubicación (Consultar Ubicación)</i>		
[Versión]	1.0 (01/09/20)		
[Dependencias]	<ul style="list-style-type: none"> RG-001 Gestionar las identificaciones de aves 		
Escenario	Se necesitan mostrar los datos de una ubicación registrado en el sistema		
Precondición	Un usuario general/especializado se ha logueado en el sistema y la ubicación a consultar existe en el sistema (CU003-1).		
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario general/especializado consulte el mapa de las especies identificadas o cuando se identifique un ave.		
Secuencia Normal	Paso	Acción	

	1	El <i>usuario general/especializado</i> solicita al sistema comenzar el proceso de ver la información de una ubicación.	
	2	Desde la app el sistema carga los datos previamente guardados en el sistema como interfaz de mapa de la ubicación del ave identificada.	
Postcondición	<i>El sistema ha consultado una ubicación.</i>		
Excepciones	Paso	Acción	
	2	Si la ubicación a consultar no se encuentra registrado en el sistema	
	E.1	El sistema informa de la situación que impide realizar la consulta.	
	E.2	El sistema recomienda crear una ubicación (CU003-1)	
	E.3	Se cancela el caso de uso.	
[Importancia]	Alto		
[Prioridad]	Alto		
Comentarios	Sin comentarios		
<CU>004	Clasificar ave		
[Versión]	1.0 (01/09/20)		
[Dependencias]	<ul style="list-style-type: none"> RG-001 Gestionar las identificaciones de aves 		
Escenario	El <i>usuario general o especializado</i> toman una foto desde la cámara o la galería y quiere identificar un ave en la aplicación móvil.		
Precondición	El <i>usuario general o especializado</i> se ha logueado en el sistema.		
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso el <i>usuario general/especializado</i> está clasificando un ave por foto de la cámara o de la galería.		
Secuencia Normal	Paso	Acción	
	1	Si se está clasificando un ave desde una foto de la cámara.	
	1.1	La aplicación mostrará los pasos que son necesarios para la realizar la clasificación de esta forma.	
	1.2	La aplicación abrirá la cámara.	
	1.3	El <i>usuario</i> debe tomar la foto, enfocando el ave que quiere identificar en la aplicación.	
	1.4	La aplicación guardará la información de la ubicación de la identificación (incluye caso CU003-1)	
	2	Si se está clasificando un ave desde una foto de la galería.	

	2.1	<i>La aplicación mostrará los pasos que son necesarios para la realizar la clasificación de esta forma.</i>
	2.2	<i>La aplicación abrirá la galería.</i>
	2.3	<i>El usuario debe elegir la imagen del ave que quiere identificar.</i>
	2.4	<i>La aplicación abrirá el mapa, en donde puede escoger la ubicación de donde se avistó el ave (incluye caso CU003-1)</i>
	3	Se incluye caso CU006 (consultar clasificador)
	4	Se incluye el caso de uso CU002-1 (consultar ave)
	5	La aplicación guarda la información correspondiente de la clasificación.
Postcondición	<i>El usuario ha clasificado un ave.</i>	
Excepciones	Paso	Acción
	1	<i>Si el móvil no tiene encendido el GPS</i>
	E.1	<i>La aplicación informa de la situación que impide realizar el registro y pide que se encienda el GPS</i>
	E.2	<i>Se cancela el caso de uso.</i>
	2	<i>Si la aplicación no tiene los permisos de (galería, cámara, internet y ubicación)</i>
	E.1	<i>La aplicación informa que no cuenta con los permisos para realizar la identificación.</i>
	E.2	<i>Se cancela el caso de uso.</i>
[Importancia]	<i>Alto</i>	
[Prioridad]	<i>Alto</i>	
Comentarios	<i>Sin comentarios</i>	
<CU>005	<i>Consultar listado de aves</i>	
[Versión]	<i>1.0 (01/09/20)</i>	
[Dependencias]	<ul style="list-style-type: none"> <i>RG-001 Gestionar las identificaciones de aves</i> 	
Escenario	<i>El usuario general o especializado quieren ver las aves que identifica o han sido identificadas en la aplicación</i>	
Precondición	<i>El usuario general o especializado se han logueado en el sistema.</i>	
Descripción	<i>El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario general o especializado quieran consultar todas las aves identificadas en el sistema</i>	

Secuencia Normal	Paso	Acción
	1	El <i>usuario general/especializado</i> pulsa el botón ver mapa de aves identificadas o lista.
	2	Si el usuario dio clic en lista: el sistema consulta todas las aves que se pueden identificar en la aplicación. Si el usuario dio clic en ver mapa de aves: el sistema consulta todas las aves identificadas por ese usuario específico
	3	Si el usuario dio clic en ver mapa de aves: el sistema carga la información como las imágenes de las aves en la interfaz de mapa. Si el usuario dio clic en lista: el sistema carga la información por tarjeta de cada ave.
	4	Si el usuario le da clic a una de las tarjetas (en la interfaz de Lista)
	4.1	El sistema consulta el caso de uso CU002-1 (consultar ave)
Postcondición	<i>El sistema ha consultado las aves</i>	
Excepciones	Paso	Acción
	2	<i>Si el usuario general o especializado no tiene aves clasificadas</i>
	E.1	<i>El sistema informa de la situación que impide realizar consulta</i>
	E.2	<i>Se cancela el caso de uso.</i>
[Importancia]	<i>Medio</i>	
[Prioridad]	<i>Medio</i>	
Comentarios	<i>Sin comentarios</i>	
<CU>006	<i>Consultar clasificador</i>	
[Versión]	<i>1.0 (01/09/20)</i>	
[Dependencias]	<ul style="list-style-type: none"> <i>RG-001 Gestionar las identificaciones de aves</i> 	
Escenario	<i>El sistema ha capturado la imagen y lo envía al modelo clasificador para obtener una predicción.</i>	
Precondición	<i>El usuario general o especializado se han logueado en el sistema.</i>	
Descripción	<i>El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el sistema consulte el modelo clasificador para dar una predicción.</i>	
Secuencia Normal	Paso	Acción
	1	El <i>sistema</i> crea una conexión http con el sistema web en flask.
	2	El sistema envía la imagen por la conexión.
	3	El sistema web recibe la imagen y la captura del método POST o GET.

	4	El sistema web envía la imagen al modelo.	
	5	El modelo valida la imagen y da una predicción.	
	6	El sistema web devuelve la predicción y el mensaje de exitoso o fallido en un JSON a la aplicación móvil.	
	7	El sistema móvil recibe la información y da continuidad al proceso del caso de uso CU004.	
Postcondición	<i>El sistema ha dado una predicción</i>		
Excepciones	Paso	Acción	
	2	Si el sistema no realiza la conexión correctamente.	
	E.1	El sistema informa de la situación que impide realizar la conexión.	
	E.2	Se cancela el caso de uso.	
[Importancia]	Alto		
[Prioridad]	Alto		
Comentarios	Sin comentarios		
<CU>007	<i>Recuperar contraseña</i>		
[Versión]	1.0 (01/09/20)		
[Dependencias]	<ul style="list-style-type: none"> RG-002 Gestionar los usuarios del sistema 		
Escenario	<i>El usuario general, especializado o administrador se les olvida la contraseña</i>		
Precondición	<i>El usuario general, especializado o administrador debe estar registrado en el sistema</i>		
Descripción	<i>El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario general, especializado o administrador intenten recuperar la contraseña</i>		
Secuencia Normal	Paso	Acción	
	1	El usuario general/especializado o administrador pulsa el botón recuperar contraseña	
	2	El sistema solicita ingresar el email registrado en el sistema del usuario	
	3	El sistema consulta el usuario incluye CU001-4 (consultar usuario)	
	4	El sistema envía la contraseña al correo registrado en el sistema	
Postcondición	<i>El sistema ha recuperado la contraseña</i>		
Excepciones	Paso	Acción	
	2	Si el usuario general, especializado o administrador no está registrado en el sistema	

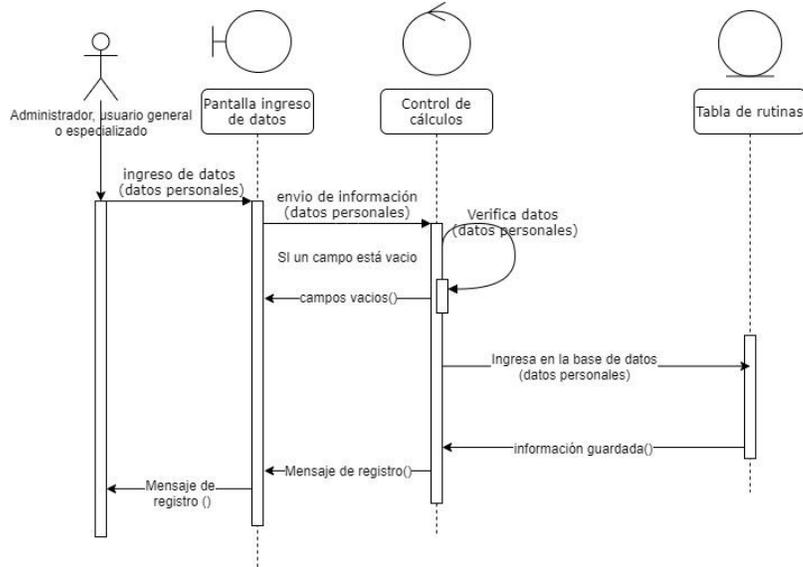
	E.1	El sistema informa de la situación que impide recuperar la contraseña
	E.2	Se cancela el caso de uso.
[Importancia]	Medio	
[Prioridad]	Medio	
Comentarios	Sin comentarios	
<CU>008	Ingresar al sistema	
[Versión]	1.0 (01/09/20)	
[Dependencias]	<ul style="list-style-type: none"> RG-002 Gestionar los usuarios del sistema 	
Escenario	El usuario general, especializado o administrador necesitan ingresar al sistema	
Precondición	El usuario general, especializado o administrador debe estar registrado en el sistema	
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario general, especializado o administrador intenten ingresar al sistema	
Secuencia Normal	Paso	Acción
	1	El usuario general/especializado o administrador pulsa el botón ingresar al sistema
	2	El sistema solicita los datos necesarios para ingresar al sistema
	3	El sistema valida la información ingresada
	4	El sistema redirecciona al panel principal
Postcondición	El usuario administrador, general u ornitólogo ingresan al sistema	
Excepciones	Paso	Acción
	2	Si el usuario general, especializado o administrador ingresan datos incorrectos
	E.1	El sistema informa de la situación que impide ingresar al sistema
	E.2	Se cancela el caso de uso.
	2	Si el usuario general, especializado o administrador no están registrados en el sistema
	E.1	El sistema informa la situación y recomienda registrarse en el sistema CU001-1
E.2	Se cancela el caso de uso.	
[Importancia]	Medio	
[Prioridad]	Medio	
Comentarios	Sin comentarios	

Fuente: Autor.

8.2.2.3 Diagramas de Secuencia del Sistema

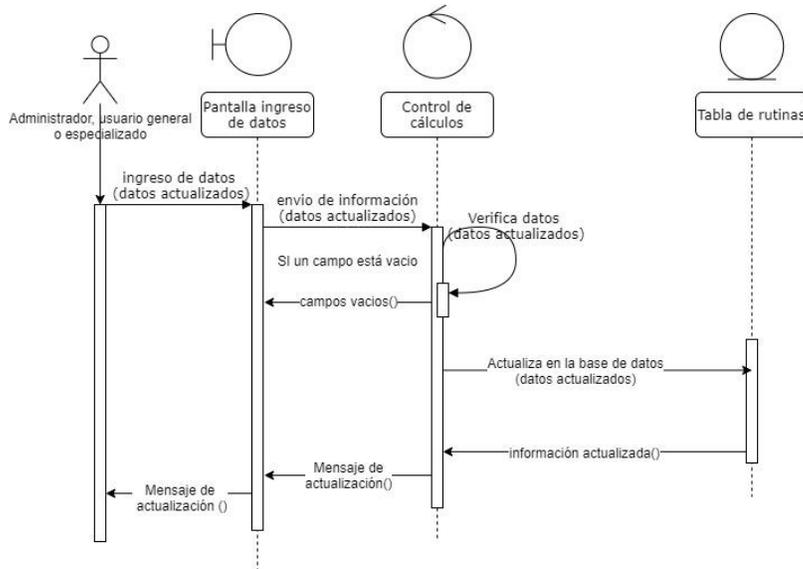
En esta sección se encuentran los diagramas de secuencia por cada caso de uso descrito en el ítem anterior.

Ilustración 31 Diagrama de Secuencia CU001-1



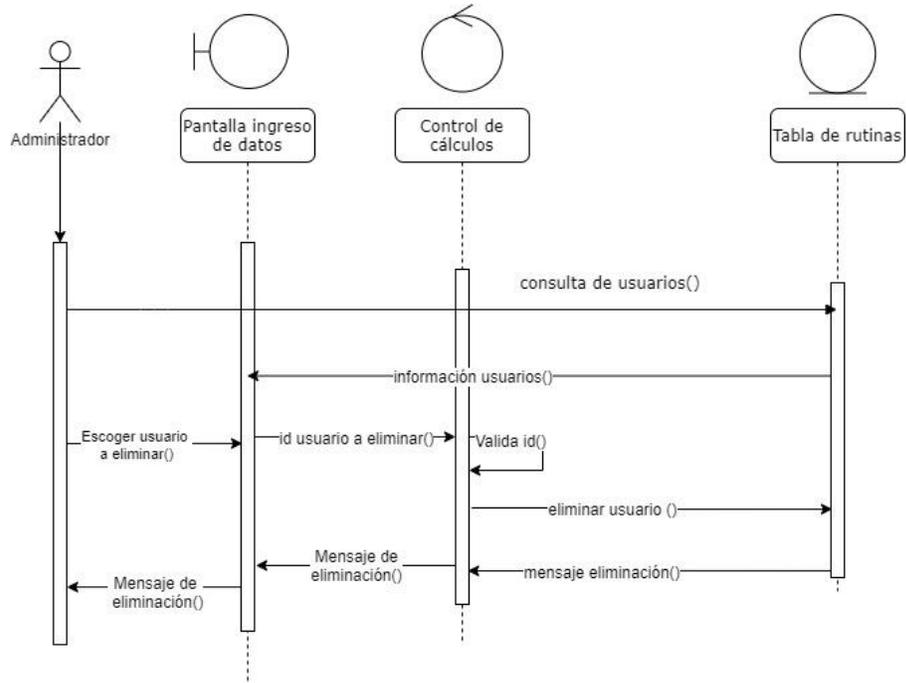
Fuente: Autor.

Ilustración 32 Diagrama de Secuencia CU001-2



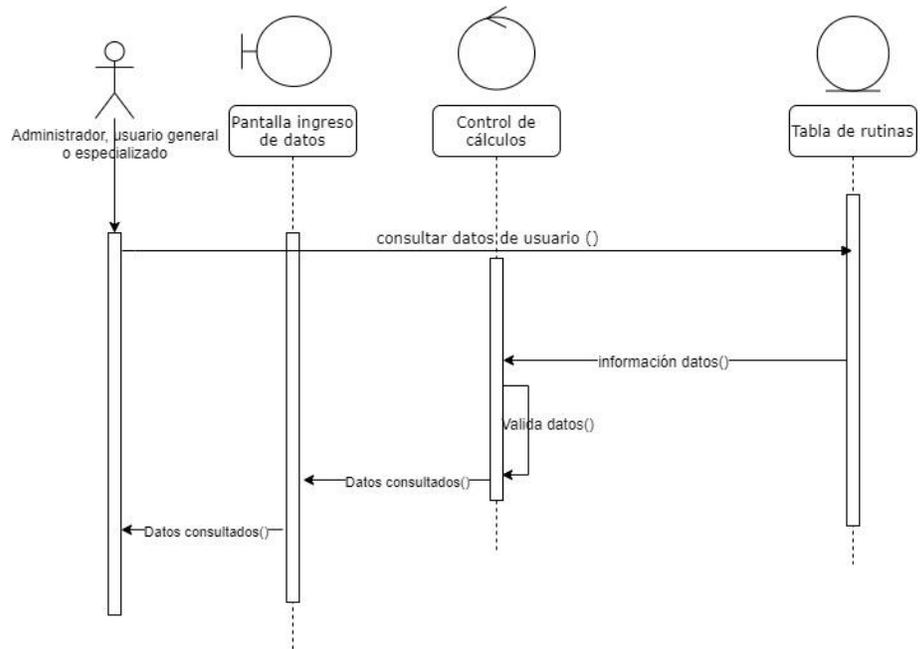
Fuente: Autor.

Ilustración 33 Diagrama de Secuencia CU001-3



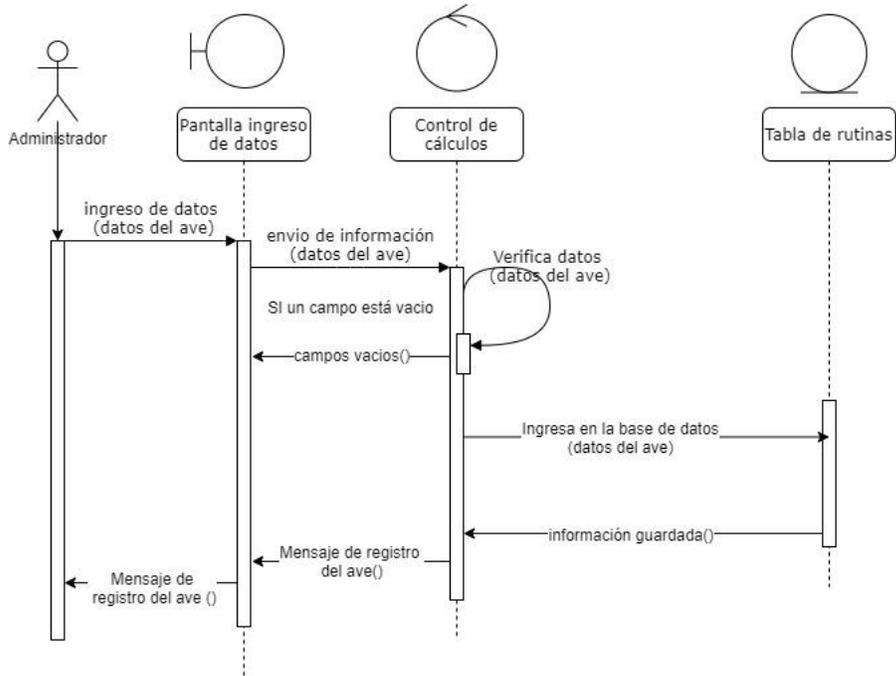
Fuente: Autor.

Ilustración 34 Diagrama de Secuencia CU001-4



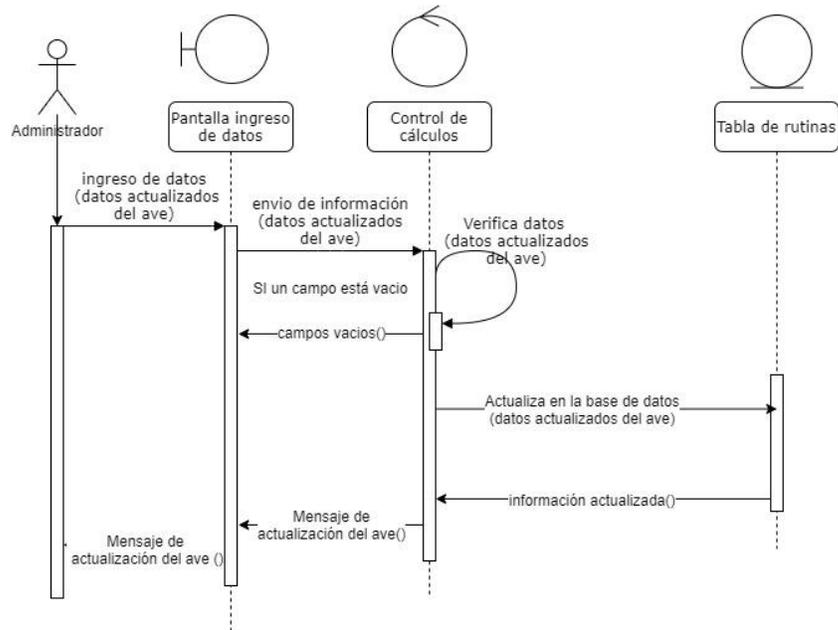
Fuente: Autor.

Ilustración 35 Diagrama de Secuencia CU002-1



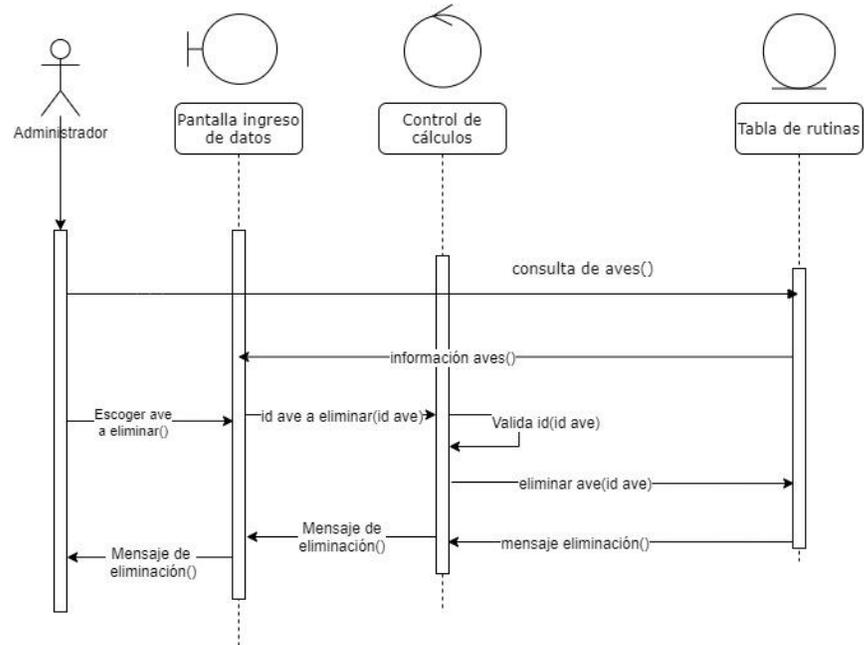
Fuente: Autor.

Ilustración 36 Diagrama de Secuencia CU002-2



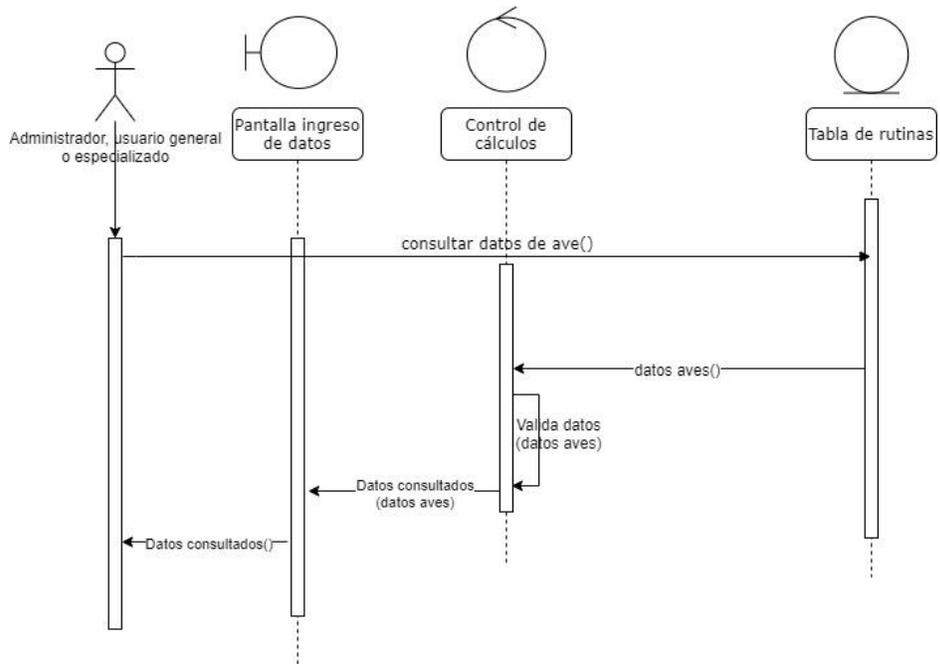
Fuente: Autor.

Ilustración 37 Diagrama de Secuencia CU002-3



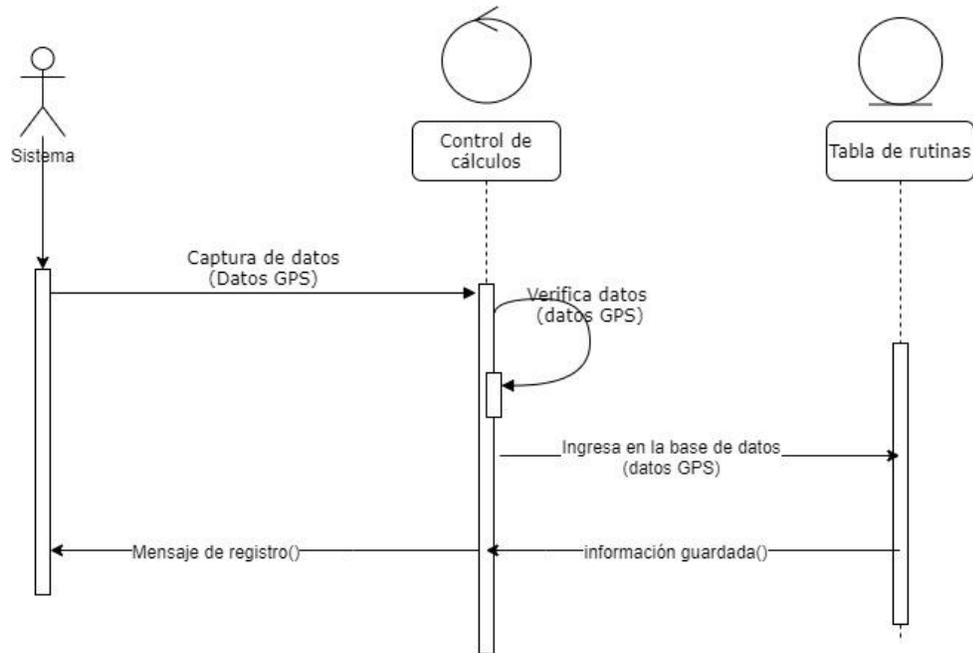
Fuente: Autor.

Ilustración 38 Diagrama de Secuencia CU002-4



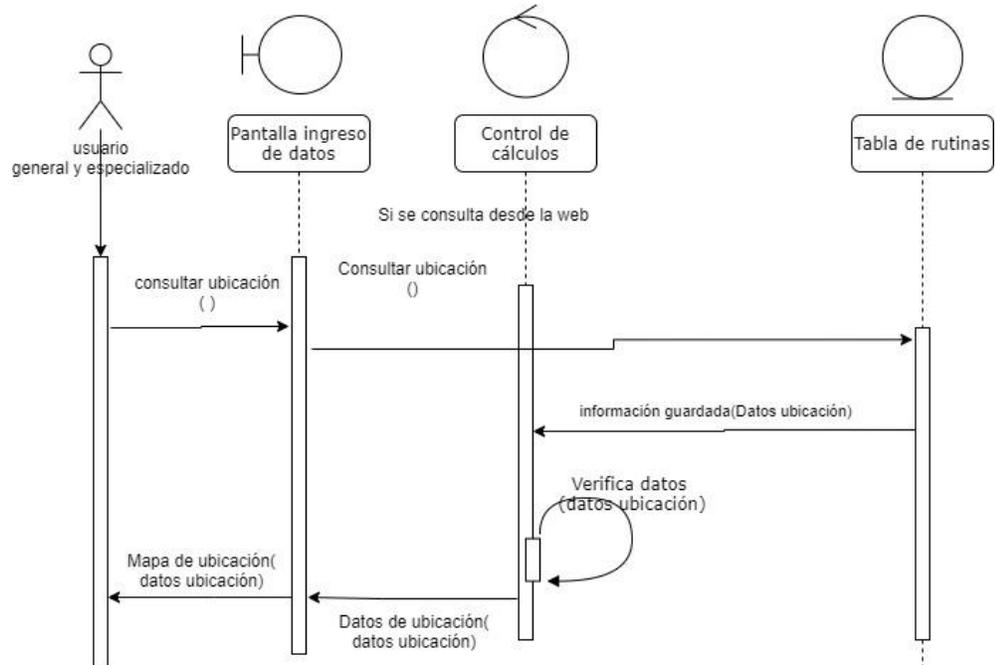
Fuente: Autor.

Ilustración 39 Diagrama de Secuencia CU003-1



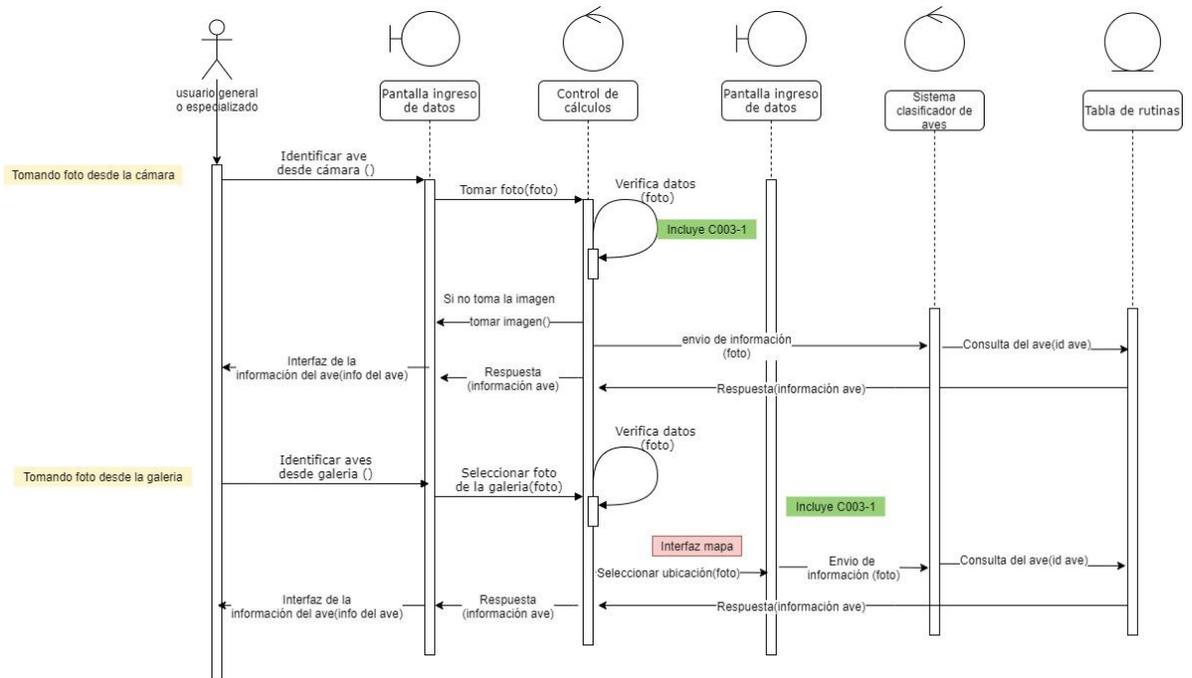
Fuente: Autor.

Ilustración 40 Diagrama de Secuencia CU003-2



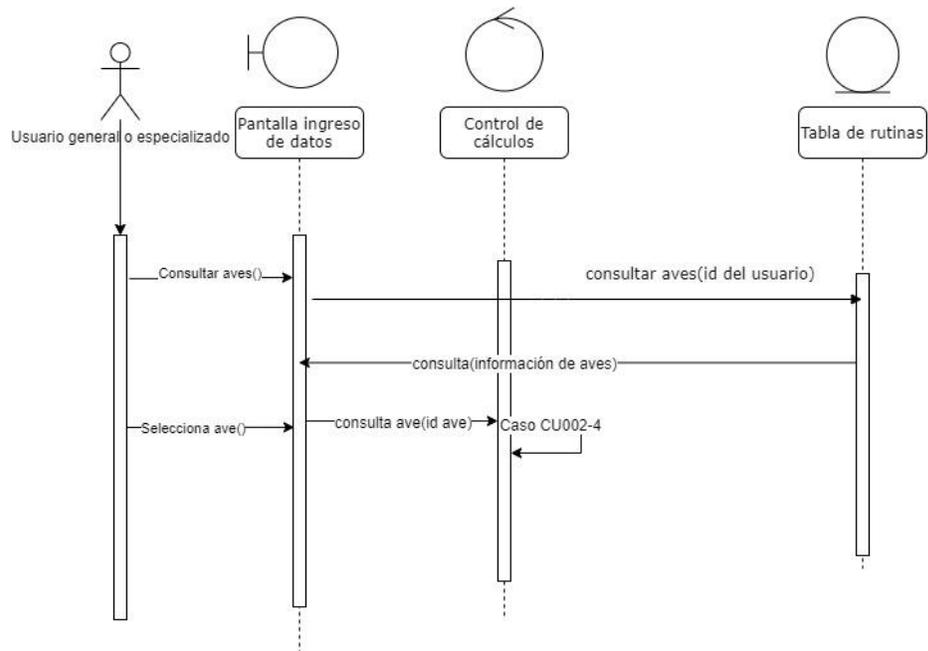
Fuente: Autor.

Ilustración 41 Diagrama de Secuencia CU004



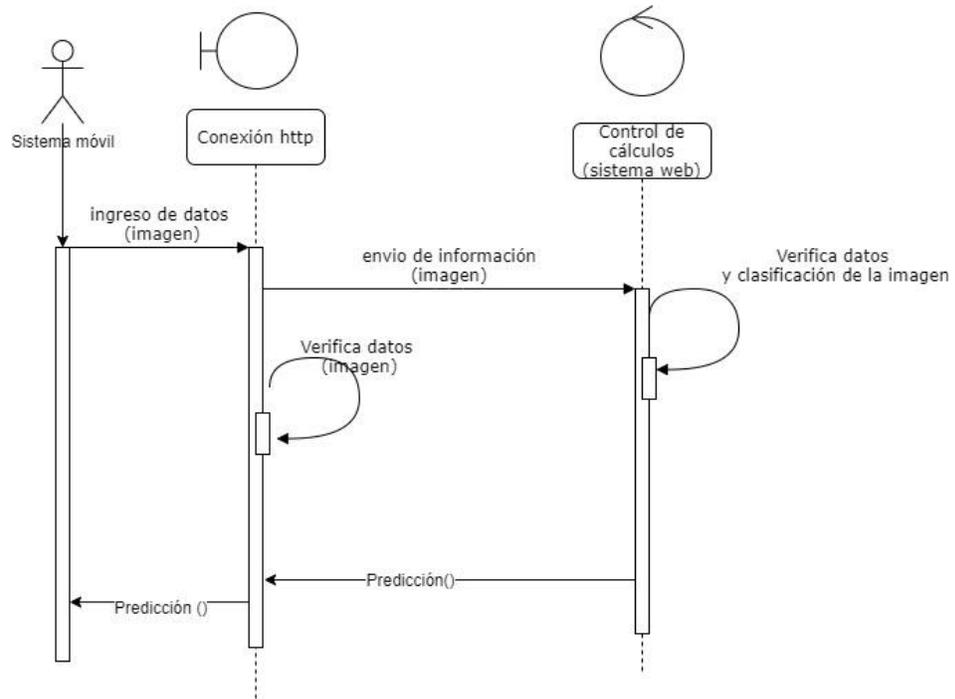
Fuente: Autor.

Ilustración 42 Diagrama de Secuencia CU005



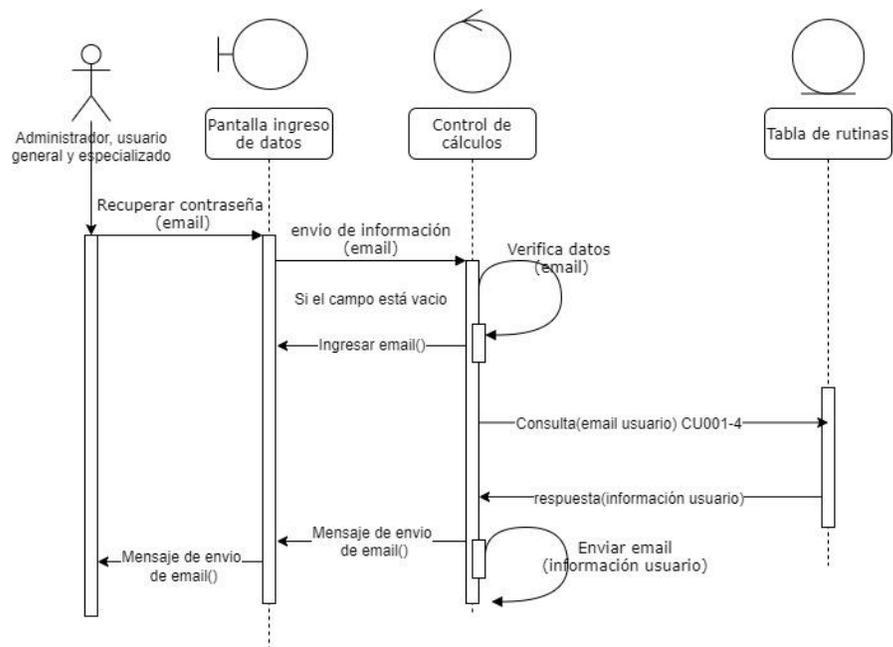
Fuente: Autor.

Ilustración 43 Diagrama de Secuencia CU006



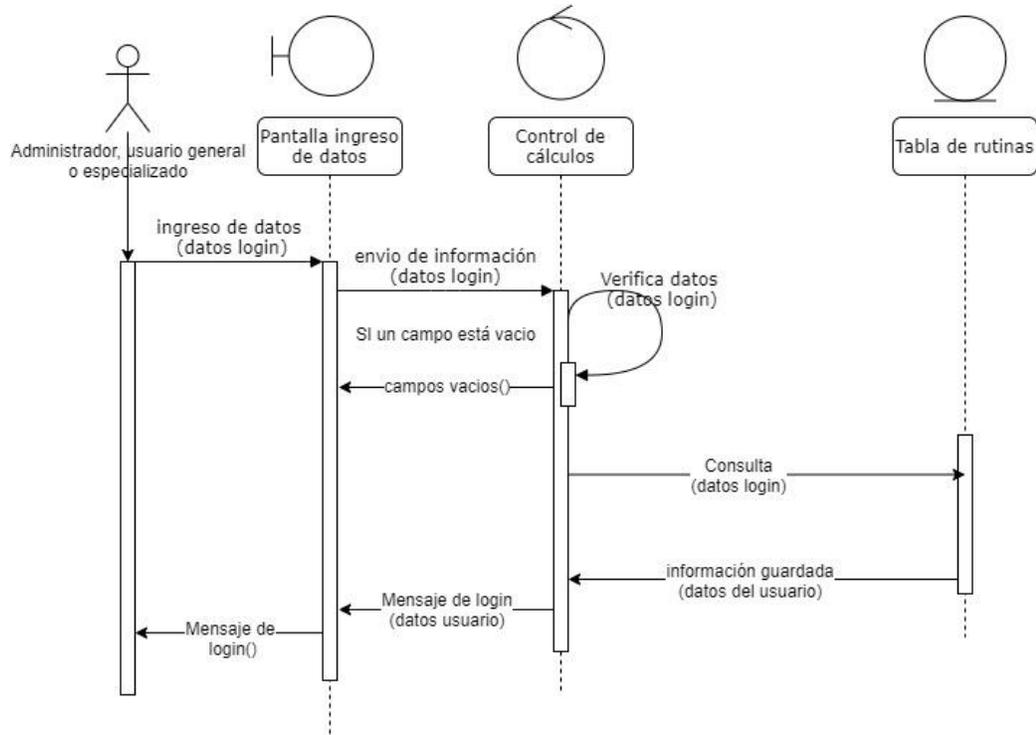
Fuente: Autor.

Ilustración 44 Diagrama de Secuencia CU007



Fuente: Autor.

Ilustración 45 Diagrama de Secuencia CU008



Fuente: Autor.

8.2.2.4 Requisitos No Funcionales del Sistema

Tabla 9 Requisitos no Funcionales del Sistema

<RN>001	<i>Seguridad lógica y de datos.</i>
[Versión]	<i><1.0> (<10/09/20>)</i>
[Dependencias]	<ul style="list-style-type: none"> <i>Gestión de usuarios.</i>
Descripción	<i>El sistema deberá permitir cambiar los permisos de acceso al sistema solamente por el administrador de acceso a datos.</i>
[Importancia]	<i>Alta.</i>
[Prioridad]	<i>Media.</i>
Comentarios	<i>Sin comentarios.</i>
<RN>002	<i>Usabilidad</i>

[Versión]	<1.0> (<10/09/20>)
[Dependencias]	<ul style="list-style-type: none"> • <i>Gestión de usuarios.</i> • <i>Identificación de aves.</i>
Descripción	<i>El sistema deberá contar con manuales de usuario estructurados adecuadamente.</i>
[Importancia]	<i>Media.</i>
[Prioridad]	<i>Media.</i>
Comentarios	<i>Sin comentarios.</i>
<RN>003	<i>Usabilidad</i>
[Versión]	<1.0> (<10/09/20>)
[Dependencias]	<ul style="list-style-type: none"> • <i>Gestión de usuarios.</i> • <i>Gestión de identificación de aves.</i>
Descripción	<i>El sistema debe contener interfaces gráficas bien formadas que permitan una apropiación intuitiva e instantánea por parte del usuario.</i>
[Importancia]	<i>Alta.</i>
[Prioridad]	<i>Alta.</i>
Comentarios	<i>Sin comentarios.</i>
<RN>004	<i>Eficiencia</i>
[Versión]	<1.0> (<10/09/20>)
[Dependencias]	<ul style="list-style-type: none"> • <i>Gestión de identificación de aves.</i>
Descripción	<i>El sistema debe ser capaz de responder en menos de 60 segundos la identificación de un ave.</i>
[Importancia]	<i>Alta.</i>
[Prioridad]	<i>Alta.</i>
Comentarios	<i>Sin comentarios.</i>
<RN>005	<i>Usabilidad</i>

[Versión]	<1.0> (<10/09/20>)
[Dependencias]	<ul style="list-style-type: none"> • <i>Gestión de usuarios.</i> • <i>Gestión de identificación de aves.</i>
Descripción	<i>El sistema deberá proporcionar mensajes de error que sean informativos y orientados a usuario final.</i>
[Importancia]	<i>Alta.</i>
[Prioridad]	<i>Alta.</i>
Comentarios	<i>Sin comentarios.</i>
<RN>006	<i>Portabilidad hardware (producto)</i>
[Versión]	<1.0> (<10/09/20>)
[Dependencias]	<ul style="list-style-type: none"> • <i>Gestión de usuarios.</i> • <i>Gestión de identificación de aves.</i>
Descripción	<i>La aplicación móvil se ejecutará en dispositivos iOS (iPhone 4S o más reciente) y dispositivos Andrid ARM.</i>
[Importancia]	<i>Alta.</i>
[Prioridad]	<i>Alta.</i>
Comentarios	<i>Sin comentarios.</i>
<RN>007	<i>Portabilidad software (producto)</i>
[Versión]	<1.0> (<10/09/20>)
[Dependencias]	<ul style="list-style-type: none"> • <i>Gestión de usuarios.</i> • <i>Gestión de identificación de aves.</i>
Descripción	<i>La aplicación web correrá localmente en computadores que tengan sistema operativo Windows, cuente con la instalación de visual Studio Code u Anaconda. Además, debe contar con la instalación de la versión de Python mayor a la 3.7 y todas las dependencias para ejecutar flask en dicho computador.</i>
[Importancia]	<i>Alta.</i>
[Prioridad]	<i>Alta.</i>

<RN>008	<i>Portabilidad software (producto)</i>
[Versión]	<1.0> (<10/09/20>)
[Dependencias]	<ul style="list-style-type: none"> • <i>Gestión de usuarios.</i> • <i>Gestión de identificación de aves.</i>
Descripción	<i>La aplicación móvil se ejecutará en sistemas operativos móviles: Android Jelly Bean, V16, 4.1 o más reciente e iOS 8 o más reciente.</i>
[Importancia]	<i>Alta.</i>
[Prioridad]	<i>Alta.</i>
Comentarios	<i>Sin comentarios.</i>
<RN>009	<i>Portabilidad hardware (producto)</i>
[Versión]	<1.0> (<10/09/20>)
[Dependencias]	<ul style="list-style-type: none"> • <i>Gestión de usuarios.</i> • <i>Gestión de identificación de aves.</i>
Descripción	<i>La aplicación web se ejecutará en computadores que tengan un procesador corei5, memoria ram mínimo 8Gb.</i>
[Importancia]	<i>Alta.</i>
[Prioridad]	<i>Alta.</i>
Comentarios	<i>Sin comentarios.</i>

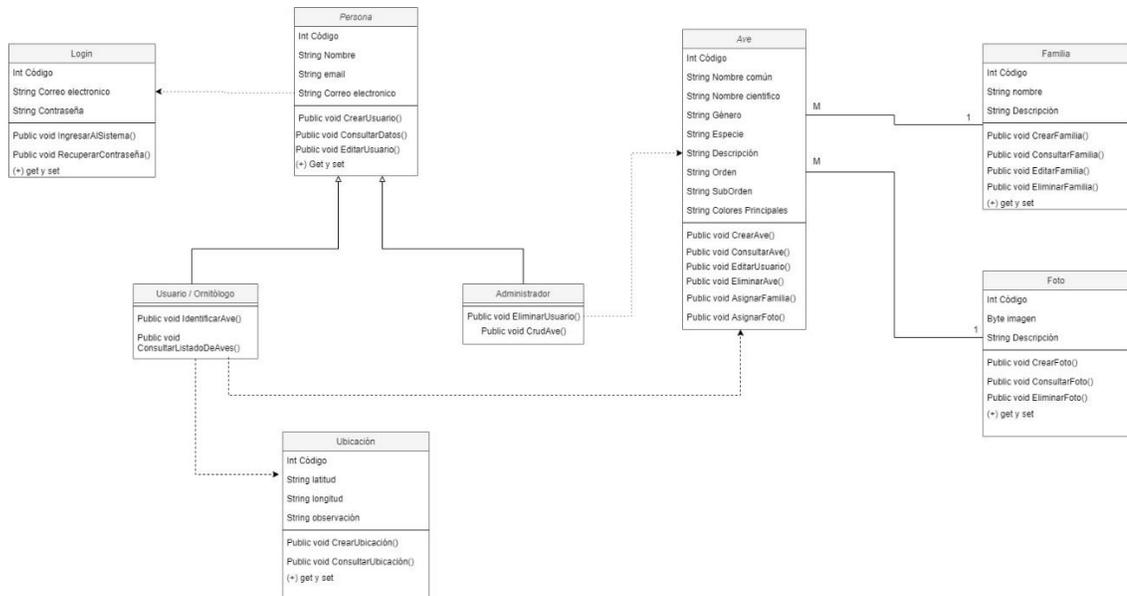
Fuente: Autor.

8.3.3 MODELO DE CLASES DEL SISTEMA

8.3.3.1 Diagramas de Clases del Sistema

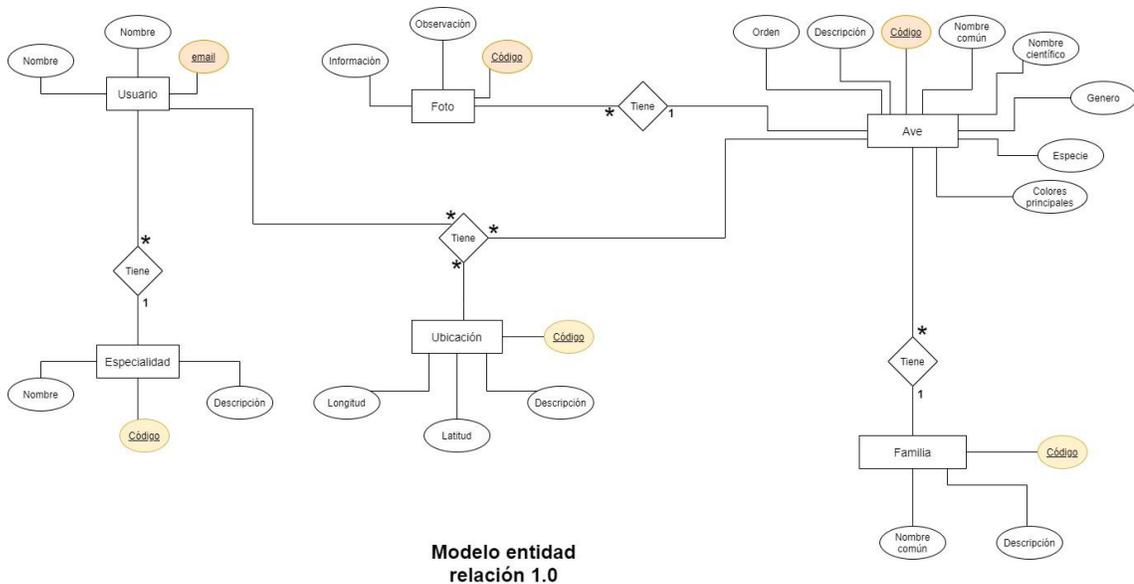
En el siguiente diagrama se describe las clases que debe contener el sistema, dado que el lenguaje seleccionado para realizar el desarrollo no está orientado a objetos, se realizan modificaciones en la codificación del sistema web y móvil.

Ilustración 46 Diagrama de Clases



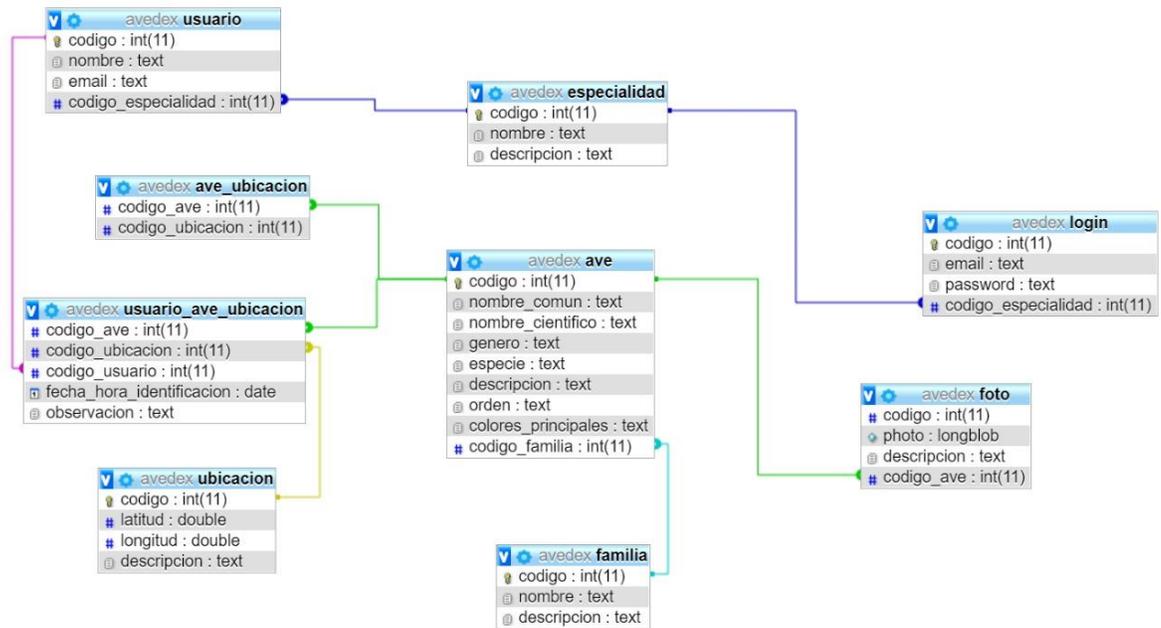
Fuente: Autor.

Ilustración 47 Modelo Entidad - Relación



Fuente: Autor.

Ilustración 48 Modelo Relacional



Fuente: Autor.

8.3.3.2 Descripción de las Clases del Sistema

Clase Ave: Esta clase permitirá la gestión de las aves (creación, consulta y edición), se determinó que debía existir otra información para complementar la del ave que es la familia y las fotos. Además, tiene una asociación con los métodos de los usuarios: administrador, general y ornitólogo.

tiene como atributos: código unitario de tipo int, Nombre común tipo String, Nombre científico tipo String, Género tipo String, Especie tipo String, Descripción tipo String, Orden tipo String, Suborden tipo String y colores principales tipo String.

Clase Persona: permitirá la gestión de los usuarios del sistema, se tiene una generalización del administrador, y general/ornitólogo porque no todos tienen las mismas funcionalidades en el sistema, en general comparten: creación de usuario, consulta de usuario, edición de usuario.

En específico los siguientes métodos:

- General / Ornitólogo: Identificar ave y consultar listado de aves.
- Administrador: eliminar usuario y Crud Ave.

Los atributos determinados para esta clase son: código unitario de tipo int, nombre de tipo String, y correo electrónico tipo String. Finalmente, esta clase se encuentra asociada con una clase Login.

Clase Login: permitirá el manejo de las sesiones de los usuarios en el sistema,

tiene métodos como: ingresar al sistema y recuperar contraseña. Los atributos pertenecientes a esta clase son: código unitario tipo int, correo electrónico tipo String y contraseña tipo String.

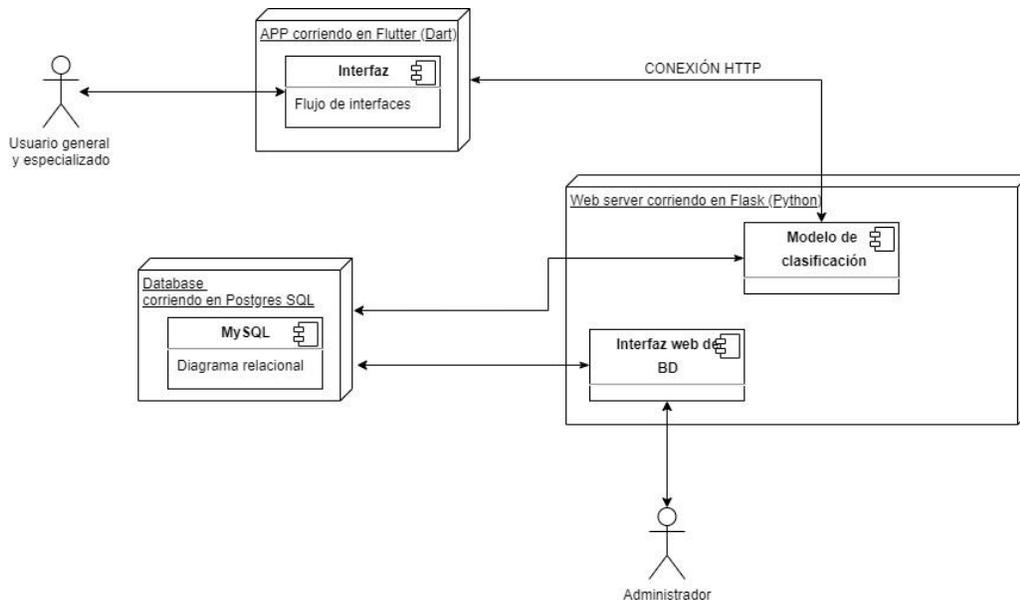
Clase Ubicación: gestionará las la identificación del ave, tiene métodos como: creación y consulta de ubicaciones.

Tiene los siguientes atributos: código unitario tipo int, latitud tipo String, longitud tipo String y una descripción tipo String.

8.3.4 ARQUITECTURA DEL SISTEMA

En esta sección encontrará el diagrama de despliegue fusionando la arquitectura lógica y física del sistema propuesto teniendo en cuenta todos las acciones e interacciones que deben tener entre ellos.

Ilustración 49 Diagrama de Despliegue



Fuente: Autor.

8.3.4.1 Descripción de los nodos del sistema

Se identificaron tres nodos:

1. Nodo de app: de vista al usuario normal y especializado, tendrá las funcionalidades descritas en los casos de uso, se encuentra realizada con el nodo de servicio web.
2. Nodo de Servicio Web: manejará el modelo clasificador de ave y el sistema web de la base de datos, en caso de que en la app se quiera identificar un ave, se envía una petición http a este nodo para ser enviado al sistema

clasificador que se relacionará con el sistema de la base de datos que a su vez se comunica con el nodo del motor de la BD. Finalmente, este nodo es el encargado de devolver al nodo App la información solicitada por el usuario de manera eficiente.

3. **Nodo de la base de datos:** es el motor de la base de datos, en donde se tendrá las tablas de rutina definidas en el diseño de esta, es quien tramitará las consultas pedidas por el nodo de servicio web y encargada de dar una respuesta de la información solicitada.

8.4. DISEÑO DE INTERFACES Y MANUAL DE USUARIO DEL SISTEMA.

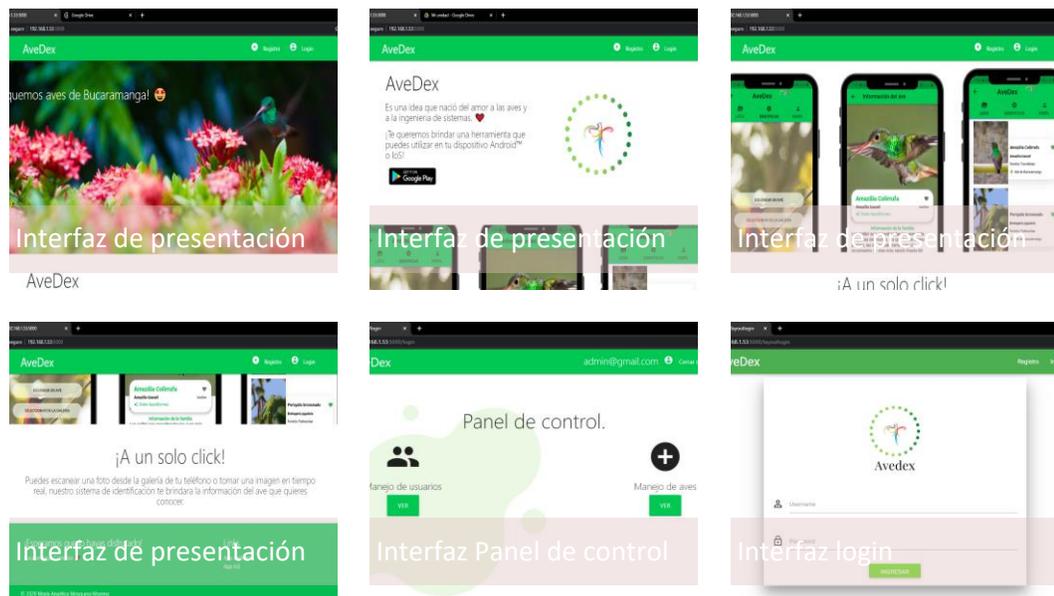
8.4.1. MAPA DEL SISTEMA

8.4.1.1. Modelo Lógico:

El sistema está compuesto tanto por la aplicación móvil, cómo por la aplicación web para la administración de datos. Por ello, está compuesto por los siguientes módulos:

- **Subsistema web:** es el sistema administrador, descrito en los casos de uso, es aquel que permitirá la administración de la información de aves y usuarios.

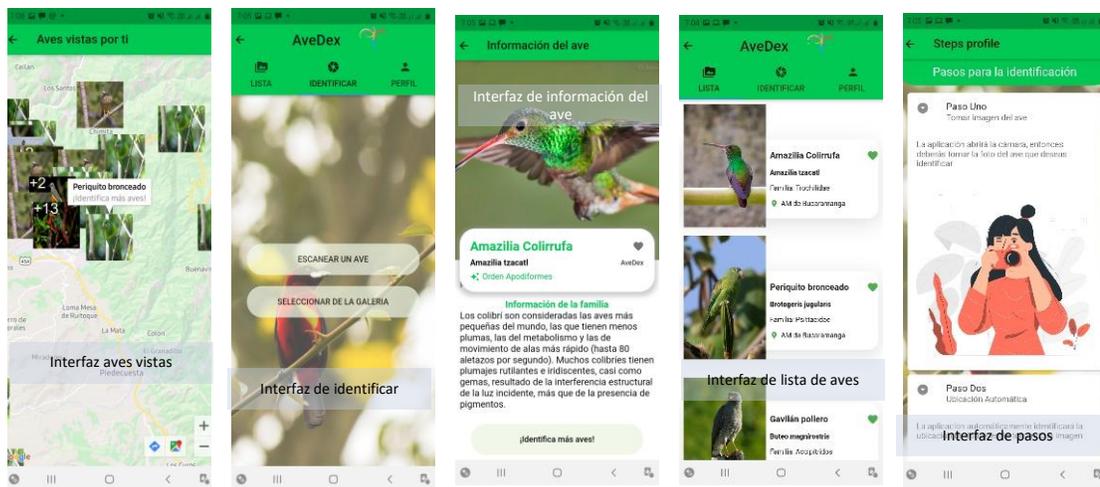
Ilustración 50 Interfaces Sistema Web



Fuente: Autor.

1. Módulo de usuarios: en este módulo se realizan todos los procesos sobre los usuarios que se encuentran registrados en el sistema.
 2. Módulo de aves: aquí se encuentran todos los procesos para realizar el crud de familias, fotos e información del ave.
- **Subsistema móvil:** es la aplicación móvil que tendrá el usuario (ornitólogo y general). A partir de ella, se podrán clasificar las aves desde una imagen obtenida del teléfono. En este sistema no se identificaron módulos.

Ilustración 51 Interfaces Aplicación Móvil

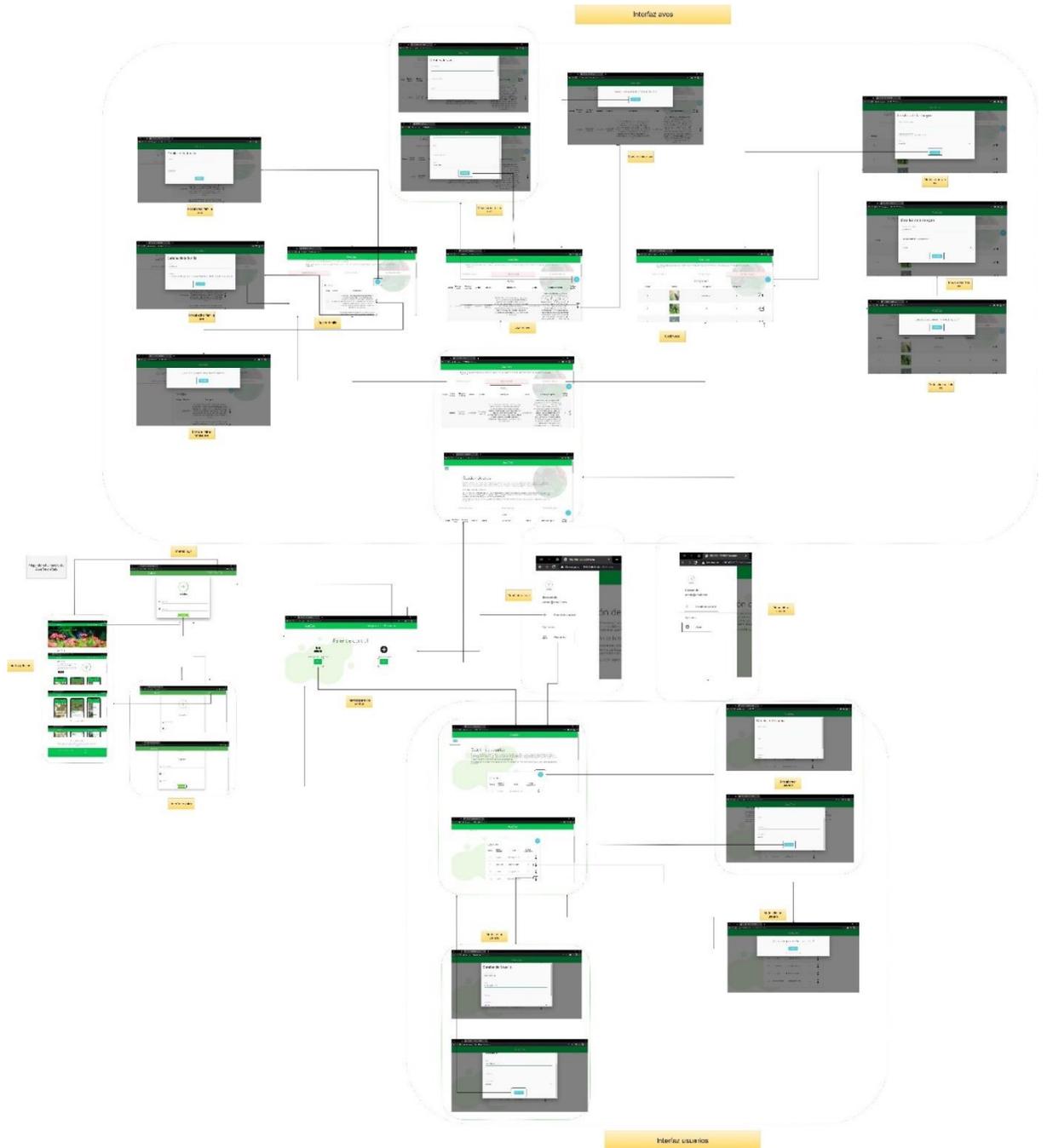


Fuente: Autor.

8.4.1.2. Navegación:

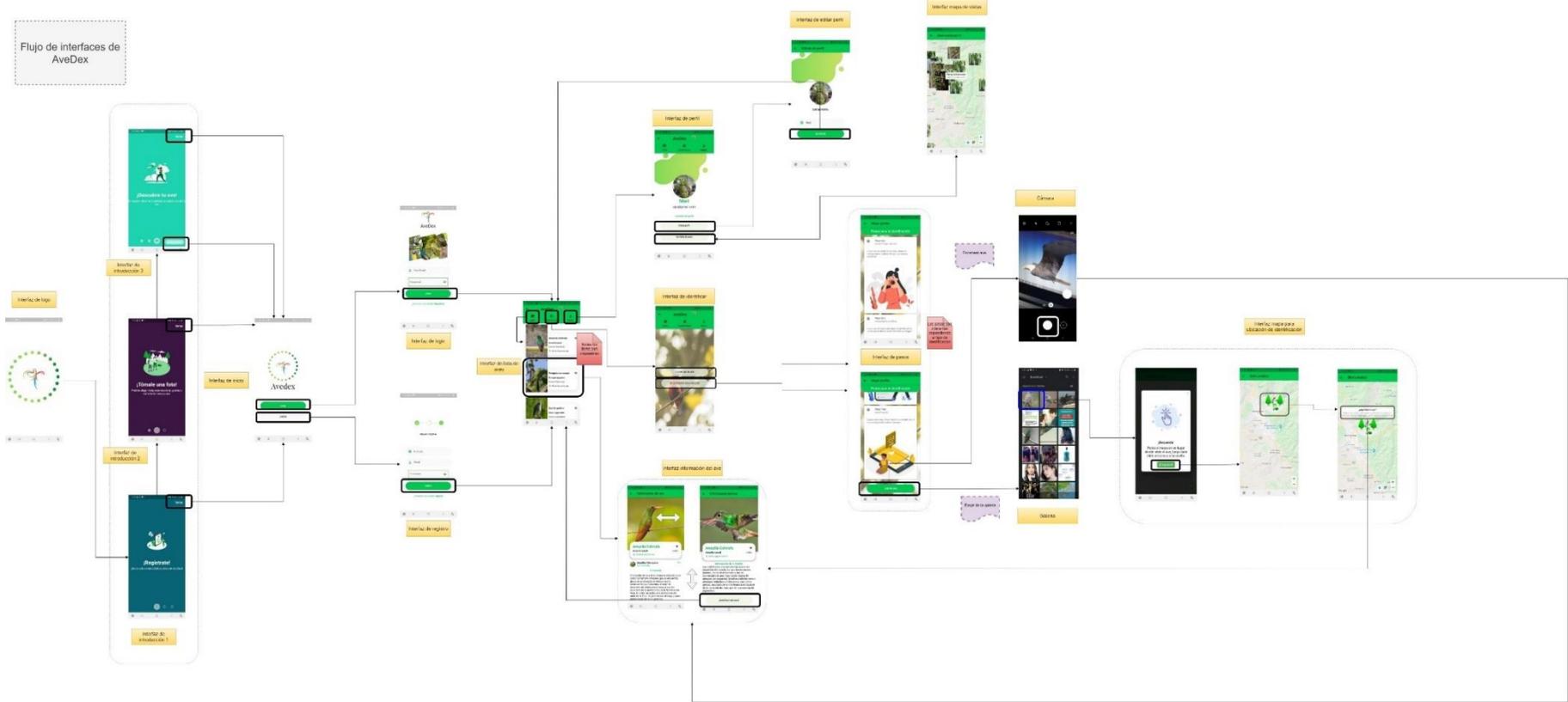
En la ilustración 49 se encuentra la navegación entre las interfaces de la aplicación móvil, también en la 50 se puede apreciar la de la aplicación móvil. En estas se detalla el flujo que tiene cada una de las interfaces al momento de que un usuario interactúe con cada una de ellas.

Ilustración 52 Flujo de Interfaces Aplicación Web



Fuente: Autor.

Ilustración 53 Flujo de Interfaces Aplicación Móvil



Fuente: Autor.

9. ANÁLISIS DE RESULTADOS

9.1. ANÁLISIS EXPLORATORIO

Inicialmente, se construyeron 9 carpetas, cada una con el nombre de su especie (Amazilia tzacatl, Brotogeris jugularis, Buteo magnirostris, Columbina talpacoti, Coragyps atratus, Melanerpes rubricapillus, Pitangus sulphuratus, Tiaris bicolor, Tyrannus melancholicus) con un total de 500 imágenes por cada una de ellas. Estas imágenes fueron obtenidas de la aplicación de un scraping a Flickr, el cual es un sitio web donde se comparten imágenes y videos de alta calidad.

A partir de esta data, se realizó una limpieza manual, identificando imágenes con un formato no legible, las que no tenían un ave en su contenido o muchas letras. Dejando un total de: Amazilia tzacatl – 496, Brotogeris jugularis-498, Buteo magnirostris- 500, Columbina talpacoti 491, Coragyps atratus-472, Melanerpes rubricapillus-491, Pitangus sulphuratus-482, Tiaris bicolor-490 y Tyrannus melancholicus con 494. Seguidamente, se tomó la especie con el número de imágenes menor para tener un estándar para todas las especies y poder realizar una división equivalente entre los datos de entrenamiento, validación y testeo. La información detallada se encuentra en la tabla 10.

Tabla 10 Descripción de Datos por Especie

Especie	Número de imágenes Inicial	Primera limpieza	Segunda limpieza	Imágenes de entrenamiento (70%)	Imágenes de validación (15%)	Imágenes de testeo (15%)
<i>Amazilia tzacatl</i>	500	496	472	330	71	71
<i>Brotogeris jugularis</i>	500	498	472	330	71	71
<i>Buteo magnirostris</i>	500	500	472	330	71	71
<i>Columbina talpacoti</i>	500	491	472	330	71	71
<i>Coragyps atratus</i>	500	472	472	330	71	71
<i>Melanerpes rubricapillus</i>	500	491	472	330	71	71
<i>Pitangus sulphuratus</i>	500	482	472	330	71	71
<i>Tiaris bicolor</i>	500	490	472	330	71	71
<i>Tyrannus melancholicus</i>	500	494	472	330	71	71

Fuente: Autor.

Luego se hizo la carga de cada una de las carpetas y se obtuvieron las estadísticas correspondientes divididas por un dataframe resumen (el cual contiene todos los datos), dataframe train, dataframe test y dataframe validate. Se realizaron dos pruebas diferentes, por lo cual, en el notebook adjunto a este documento puede encontrar imágenes diferentes, esto se realizó con el fin de tener un mayor número de imágenes vistas con su respectivo histograma.

Se resalta que el dataset construido no es un dataset estratificado porque no hay una jerarquización de cada una de las especies, como se puede observar en el número de imágenes de cada especie, todas tienen el mismo valor, por lo tanto el dataset está uniformemente distribuido, porque se espera que el sistema pueda predecir por cada especie por igual.

9.1.1 DATAFRAMES

El dataframe resumen fue construido por cuatro columnas: name_class (nombre de la clase), train (datos de entrenamiento), validate (datos de validación), test (datos de testeo), en las últimas tres columnas se almacenaron el número de datos correspondiente a la sección (ilustración 54).

Ilustración 54 Descripción de datos por clases Dataframe Resumen

	name_class	train	validate	test
0	Amazilia_tzacatl	330	71	71
1	Brotogeris_jugularis	330	71	71
2	Buteo_magnirostris	330	71	71
3	Columbina_talpacoti	330	71	71
4	Coragyps_atratus	330	71	71
5	Melanerpes_rubricapillus	330	71	71
6	Pitangus_sulphuratus	330	71	71
7	Tiaris_bicolor	330	71	71
8	Tyrannus_melanolicus	330	71	71

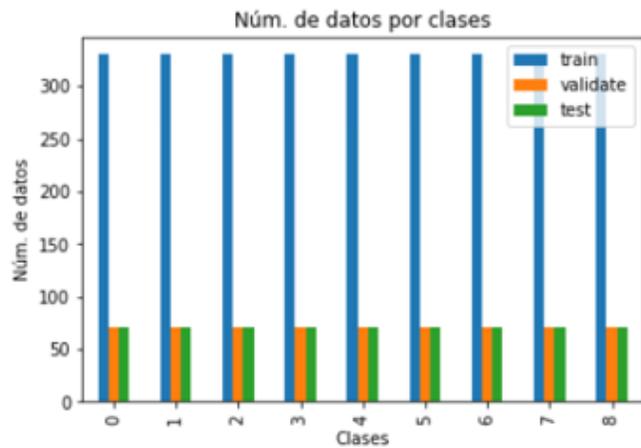
Fuente: Autor.

Seguidamente, se obtuvieron las estadísticas básicas de los dataframe: número de datos, media, mínimo (numero), máximo (numero), el 25% 50% 75% de los datos. Además, la graficación de los datos para verificar que las especies son equivalentes con su número de datos por sección. Explícitamente los encuentra en la ilustración 56, ilustración 57, ilustración 58 e ilustración 59. Por otro lado, se graficó cada una de las especies con su respectivo nombre científico (ilustración 60). En conclusión, las especies en general cuentan con un numero de imágenes

aceptable para obtener un modelo que pueda clasificar de forma óptima. Cabe resaltar que los conjuntos de imágenes contienen las especies con diferente número de cada una de ellas en una sola imagen.

Ilustración 55 Estadísticas Básicas Dataframe Resumen

	train	validate	test
count	9.0	9.0	9.0
mean	330.0	71.0	71.0
std	0.0	0.0	0.0
min	330.0	71.0	71.0
25%	330.0	71.0	71.0
50%	330.0	71.0	71.0
75%	330.0	71.0	71.0
max	330.0	71.0	71.0



Fuente: Autor.

Ilustración 56 Dataframe Train

	width	height
count	330.000000	330.000000
mean	1681.239394	1255.803030
std	1220.913983	887.393855
min	317.000000	255.000000
25%	1024.000000	683.000000
50%	1024.000000	801.000000
75%	1920.000000	1534.500000
max	8040.000000	5612.000000

Fuente: Autor.

Ilustración 57 Dataframe Validate

	width	height
count	71.000000	71.000000
mean	2016.760563	1454.098592
std	1390.142567	861.516125
min	681.000000	416.000000
25%	1024.000000	768.000000
50%	1402.000000	1024.000000
75%	2889.000000	1811.000000
max	6000.000000	3989.000000

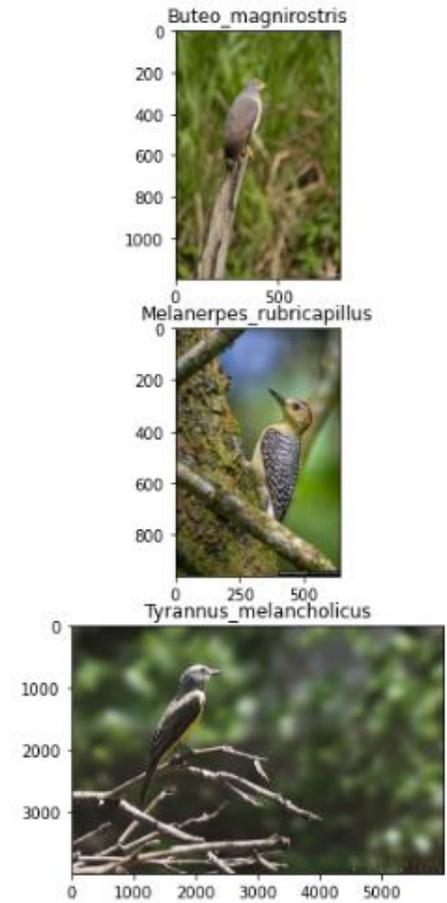
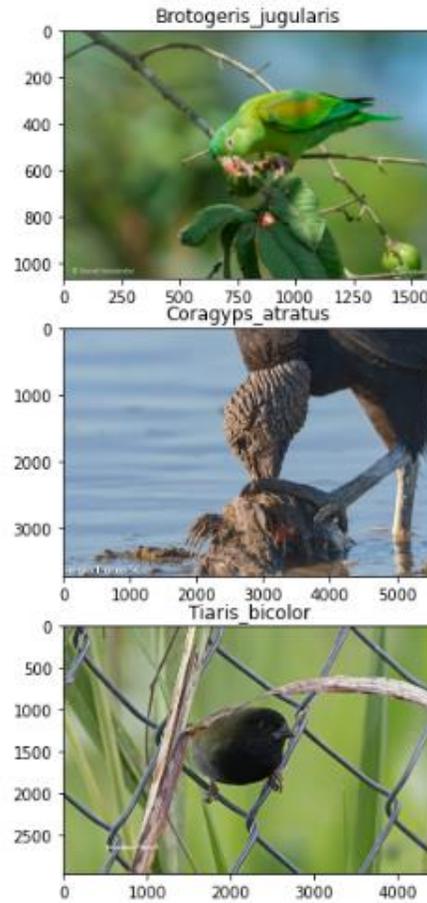
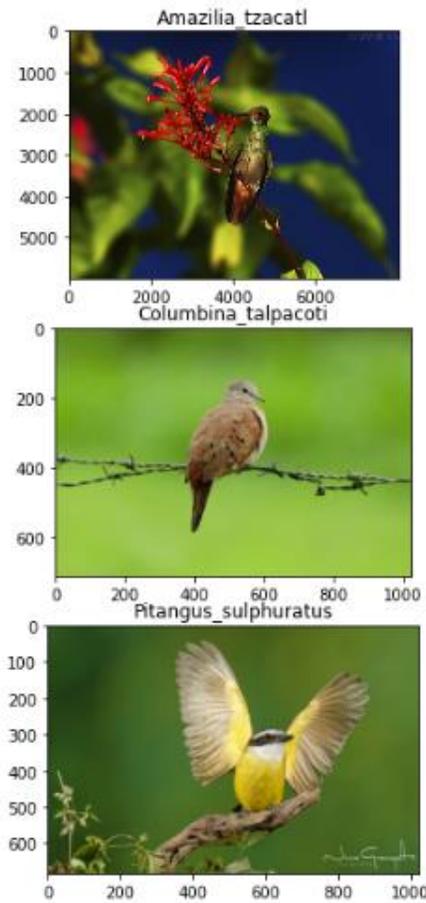
Fuente: Autor.

Ilustración 58 Dataframe Test

	width	height
count	71.000000	71.000000
mean	1842.563380	1383.647887
std	1255.088497	925.827307
min	768.000000	531.000000
25%	1024.000000	690.000000
50%	1132.000000	971.000000
75%	2468.500000	2000.000000
max	5568.000000	4278.000000

Fuente: Autor.

Ilustración 59 Imagen y Nombre de Cada Especie



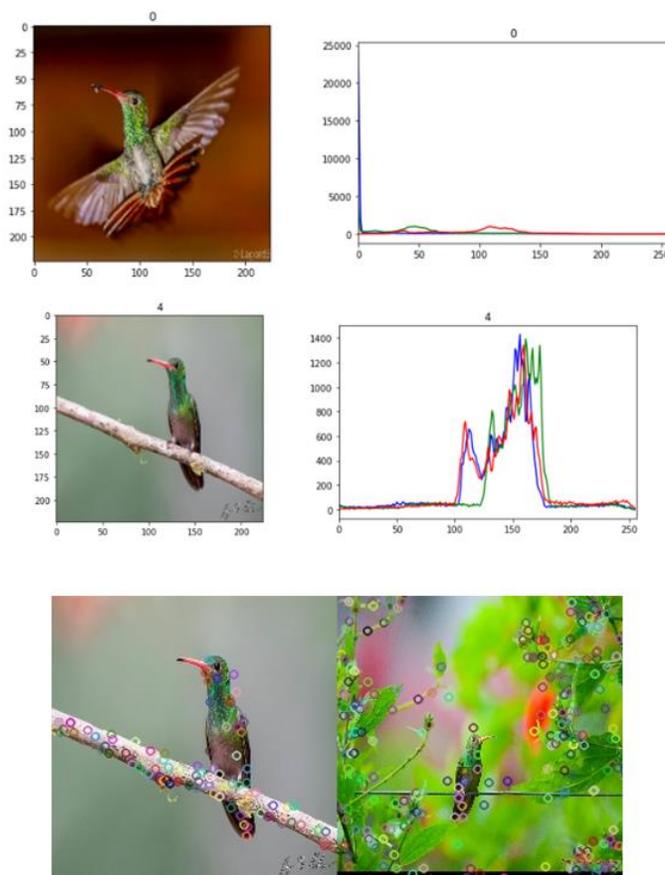
Fuente: Autor.

9.1.2. ANÁLISIS POR ESPECIE

Para realizar este análisis se tomó una muestra de 20 imágenes por cada especie, fueron leídas por la librería de OpenCV, redimensionadas y posteriormente se obtuvieron los histogramas. Por cada especie se tomaron aleatoriamente 5 imágenes. Además, se les aplicó un algoritmo de semejanza para encontrar puntos similares entre imágenes de la misma especie, lo que se refiere a “Good matches” en la salida del algoritmo.

Amazilia tzacatl:

Ilustración 60 Análisis Especie *Amazilia tzacatl*



Fuente: Autor.

Las imágenes tienen el mismo tamaño y canal

42722

Las imágenes no son iguales

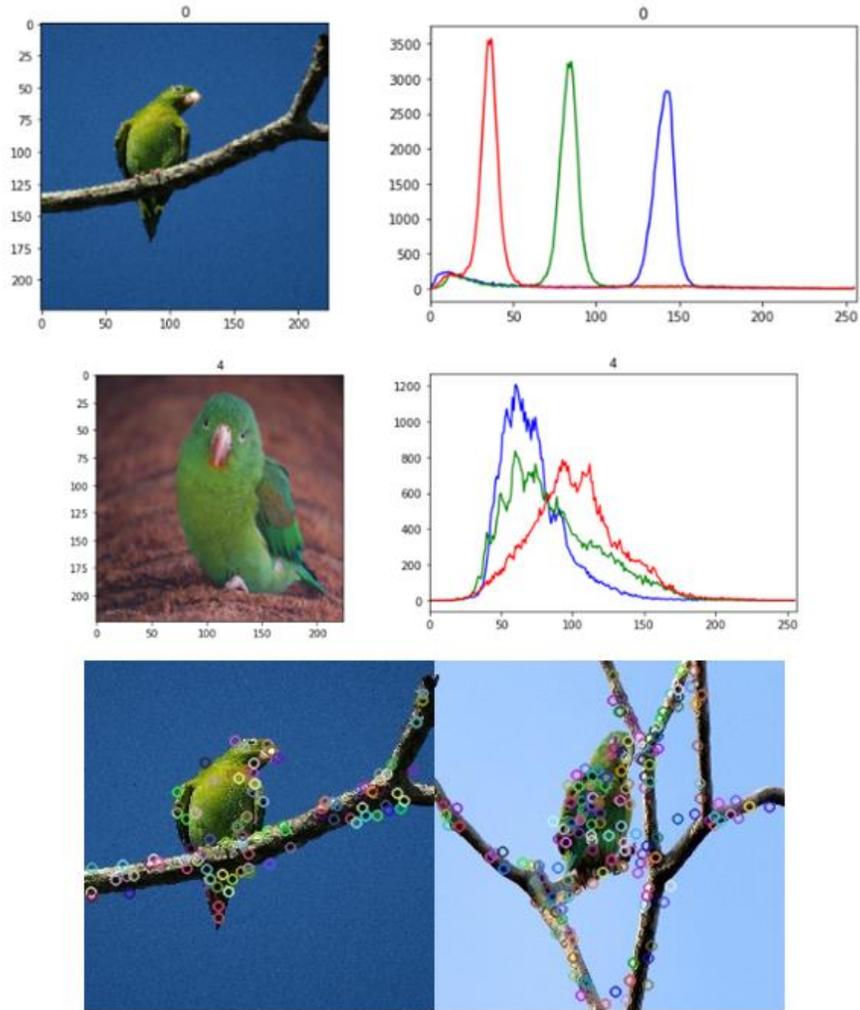
Keypoints 1st image 98

Keypoints 2nd image 280

GOOD matches 0

***Brotogeris jugularis*:**

Ilustración 61 Análisis de Especie *Brotogeris jugularis*



Fuente: Autor.

Las imágenes tienen el mismo tamaño y canal

6814

Las imágenes no son iguales

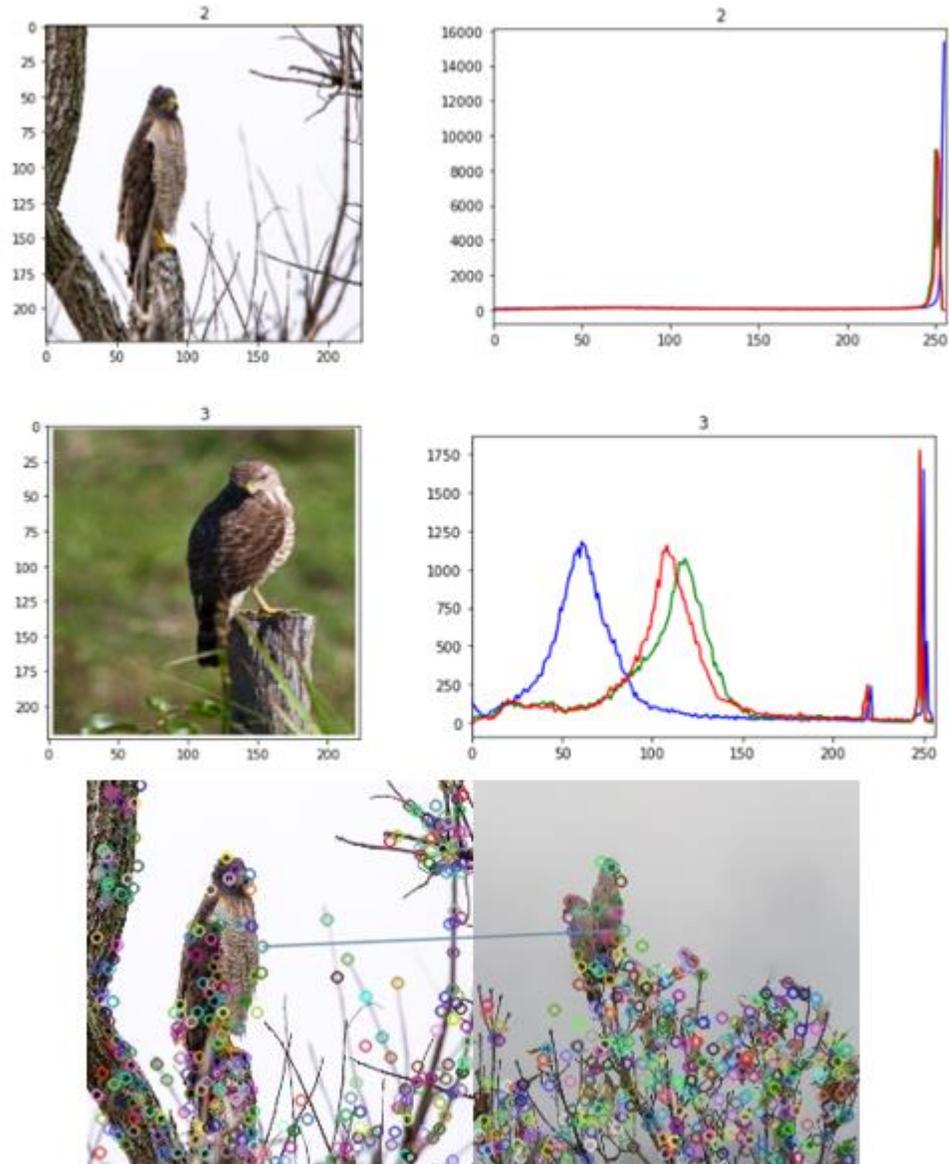
Keypoints 1st image 113

Keypoints 2st image 189

GOOD matches 0

Buteo magnirostris:

Ilustración 62 Análisis de Especie *Buteo magnirostris*

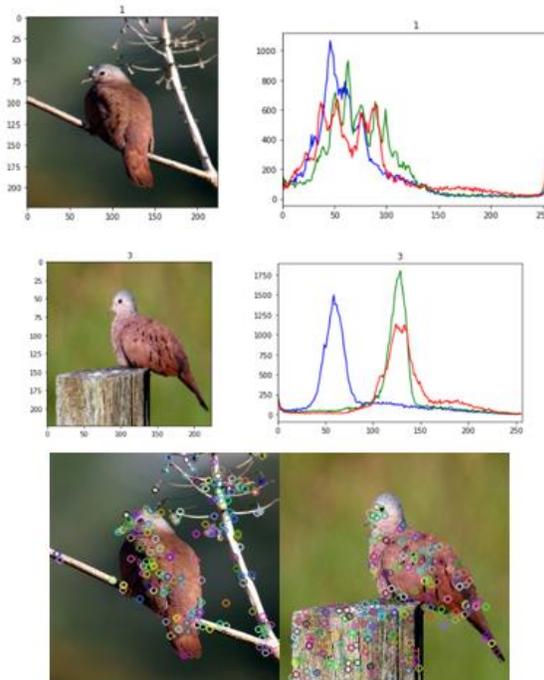


Fuente: Autor.

Las imágenes tienen el mismo tamaño y canal
36926
Las imágenes no son iguales
Keypoints 1st image 363
Keypoints 2st image 435
GOOD matches 1

Columbina talpacoti

Ilustración 63 Análisis de Especie *Columbina talpacoti*



Fuente: Autor.

Las imágenes tienen el mismo tamaño y canal
22722

Las imágenes no son iguales

Keypoints 1st image 209

Keypoints 2st image 237

GOOD matches 0

Ilustración 64 Especie Cuatro Aplicando Algoritmo de Semejanza, Imagen Tres



Fuente: Autor.

Las imágenes tienen el mismo tamaño y canal

0

Las imágenes son completamente iguales

Keypoints 1st image 262

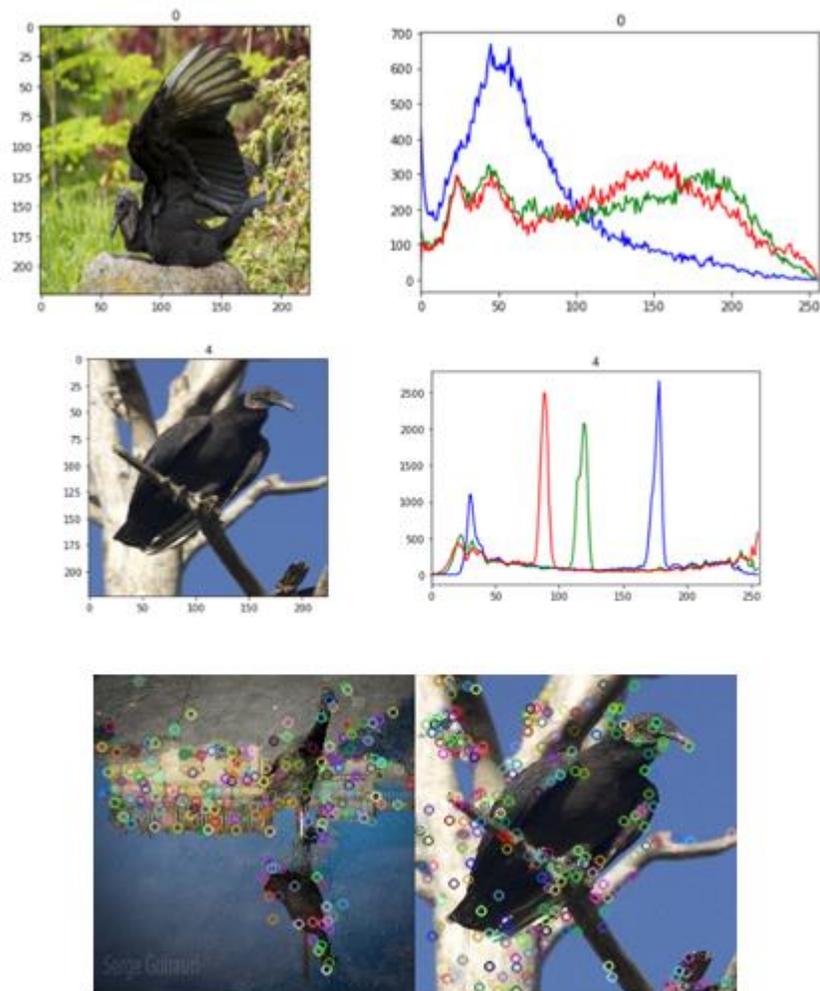
Keypoints 2st image 262

GOOD matches 262

Esta comparación en específico se realizó para que algoritmo encontrara las semejanzas entre dos imágenes totalmente iguales, en esta prueba las salidas fueron totalmente satisfactorias, ya que dio una salida acertada.

Coragyps atratus:

Ilustración 65 Análisis de Especie Coragyps Atratus



Fuente: Autor.

Las imágenes tienen el mismo tamaño y canal
17437

Las imágenes no son iguales

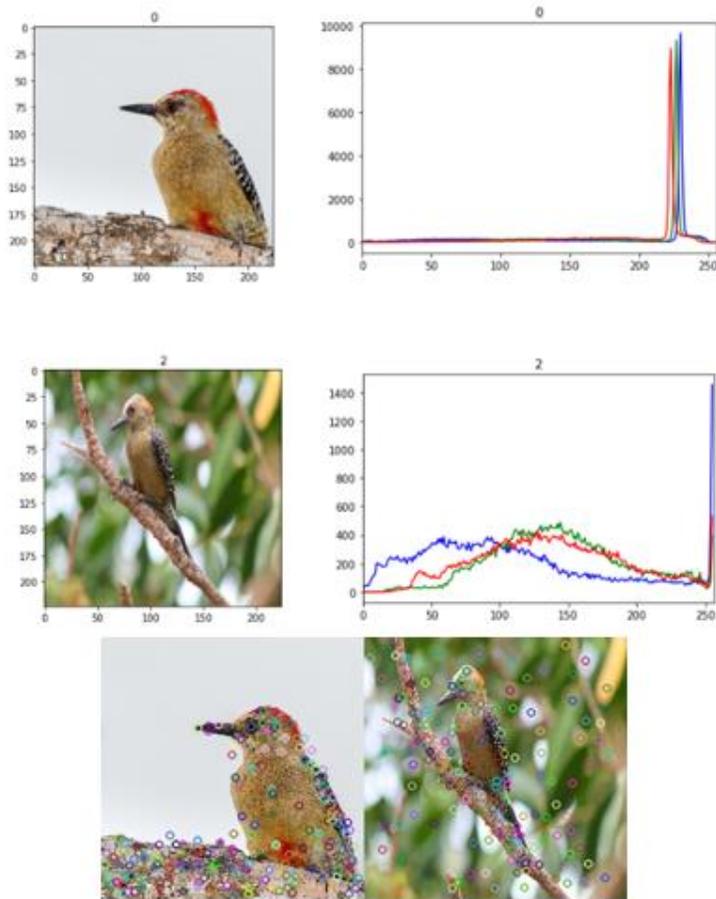
Keypoints 1st image 246

Keypoints 2st image 274

GOOD matches 0

Melanerpes rubricapillus:

Ilustración 66 Análisis de Especie *Melanerpes Rubricapillus*



Fuente: Autor.

Las imágenes tienen el mismo tamaño y canal
38262

Las imágenes no son iguales

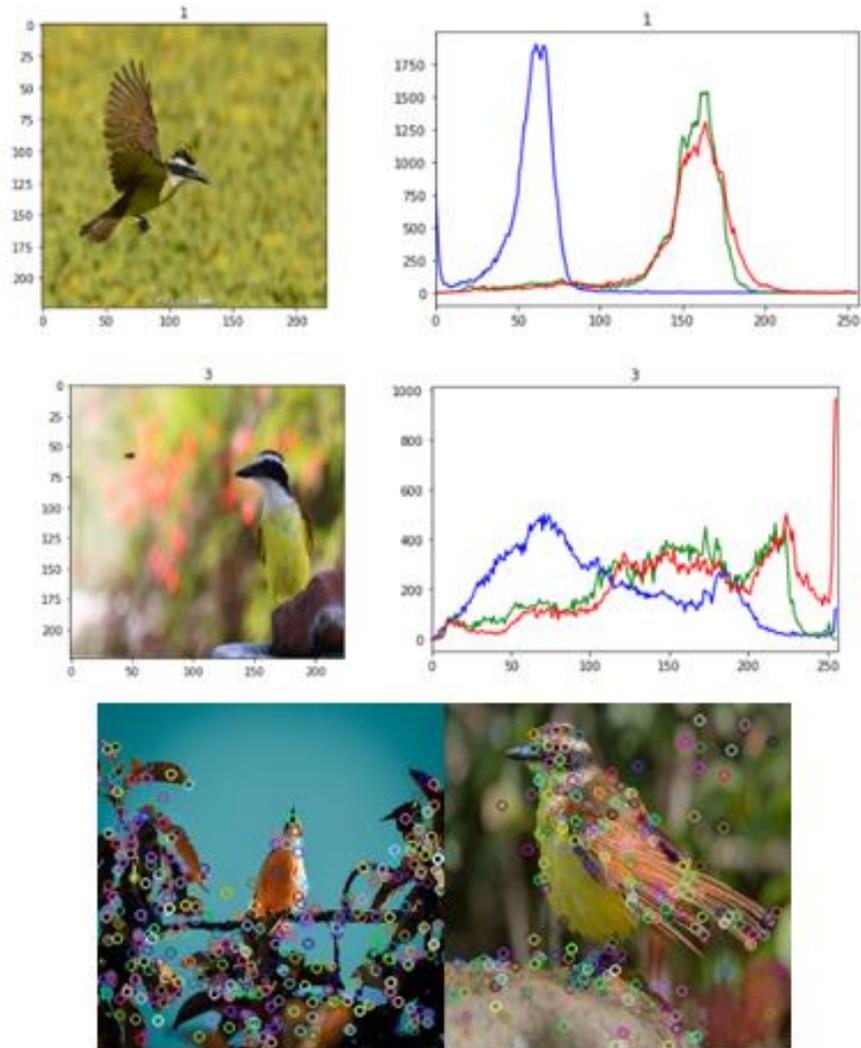
Keypoints 1st image 282

Keypoints 2st image 214

GOOD matches

***Pitangus sulphuratus*:**

Ilustración 67 Análisis de Especie Pitangus Sulphuratus



Fuente: Autor.

Las imágenes tienen el mismo tamaño y canal
32085

Las imágenes no son iguales

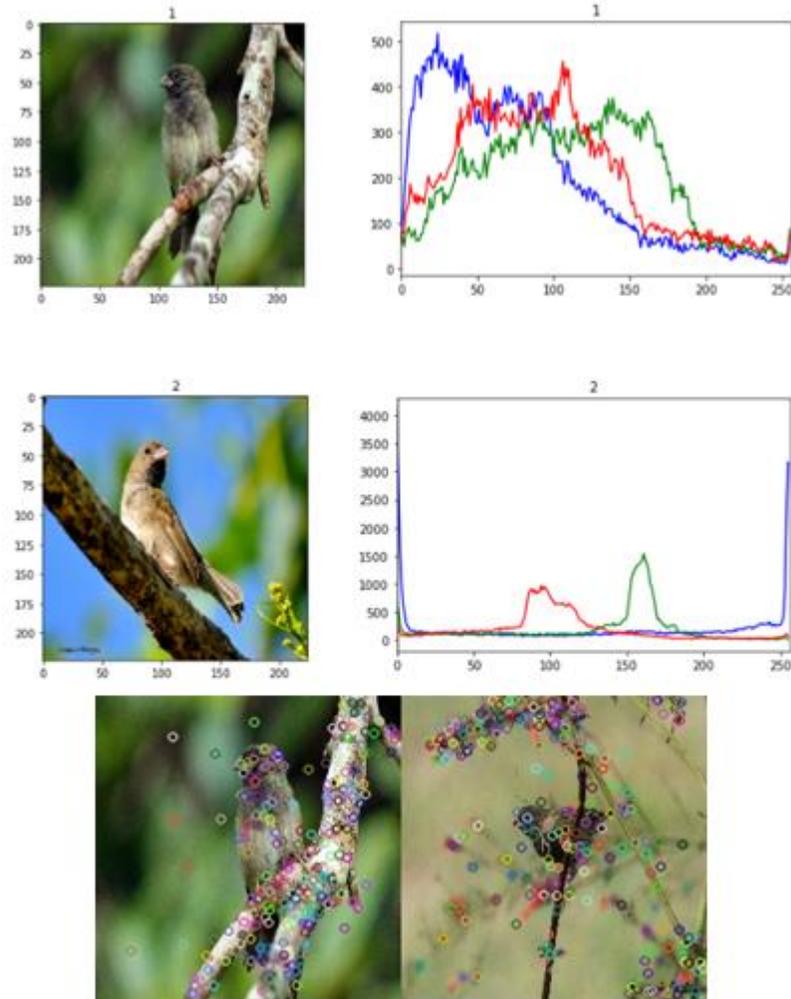
Keypoints 1st image 387

Keypoints 2st image 235

GOOD matches 0

Tiaris bicolor:

Ilustración 68 Análisis de Especie Tiaris Bicolor



Fuente: Autor.

Las imágenes tienen el mismo tamaño y canal

14120

Las imágenes no son iguales

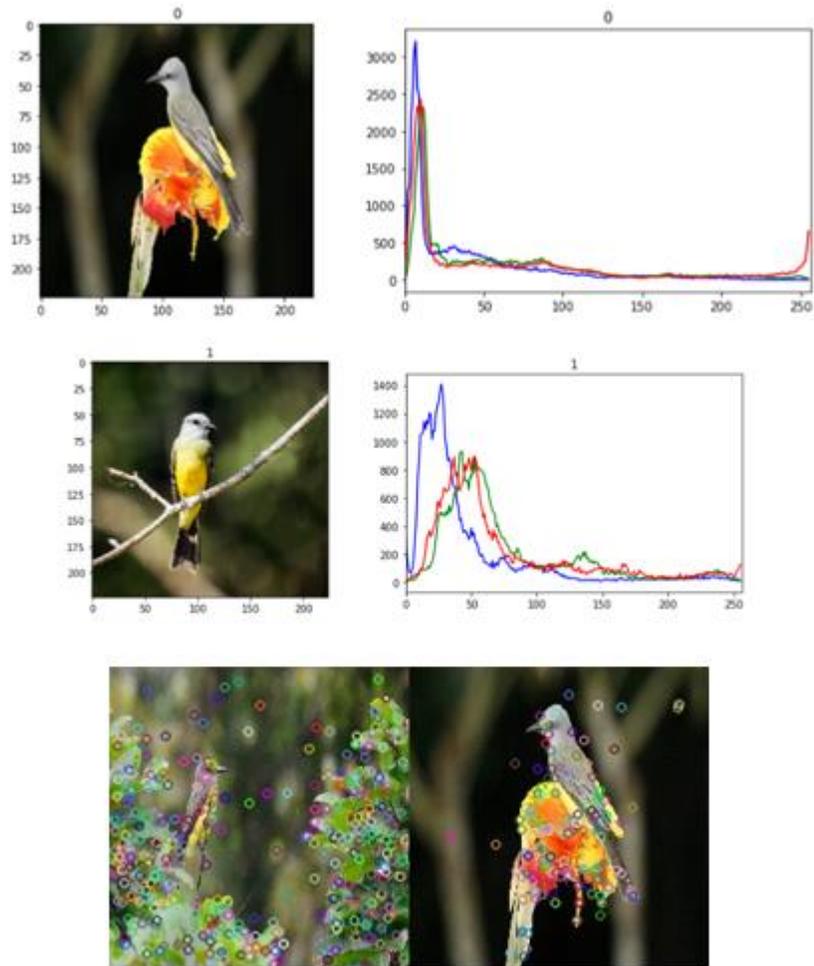
Keypoints 1st image 236

Keypoints 2st image 288

GOOD matches 0

***Tyrannus melancholicus*:**

Ilustración 69 Análisis de Especie Tyrannus Melancholicus



Fuente: Autor.

Las imágenes tienen el mismo tamaño y canal

36865

Las imágenes no son iguales

Keypoints 1st image 468

Keypoints 2nd image 143

GOOD matches 0

9.2. ANÁLISIS DE DESARROLLO DEL MODELO DE IA.

Se realizó un plan de pruebas con la arquitectura ResNet para obtener el mejor modelo. En la siguiente tabla se registró el accuracy de cada una de las pruebas, adjuntando el código y el modelo generado.

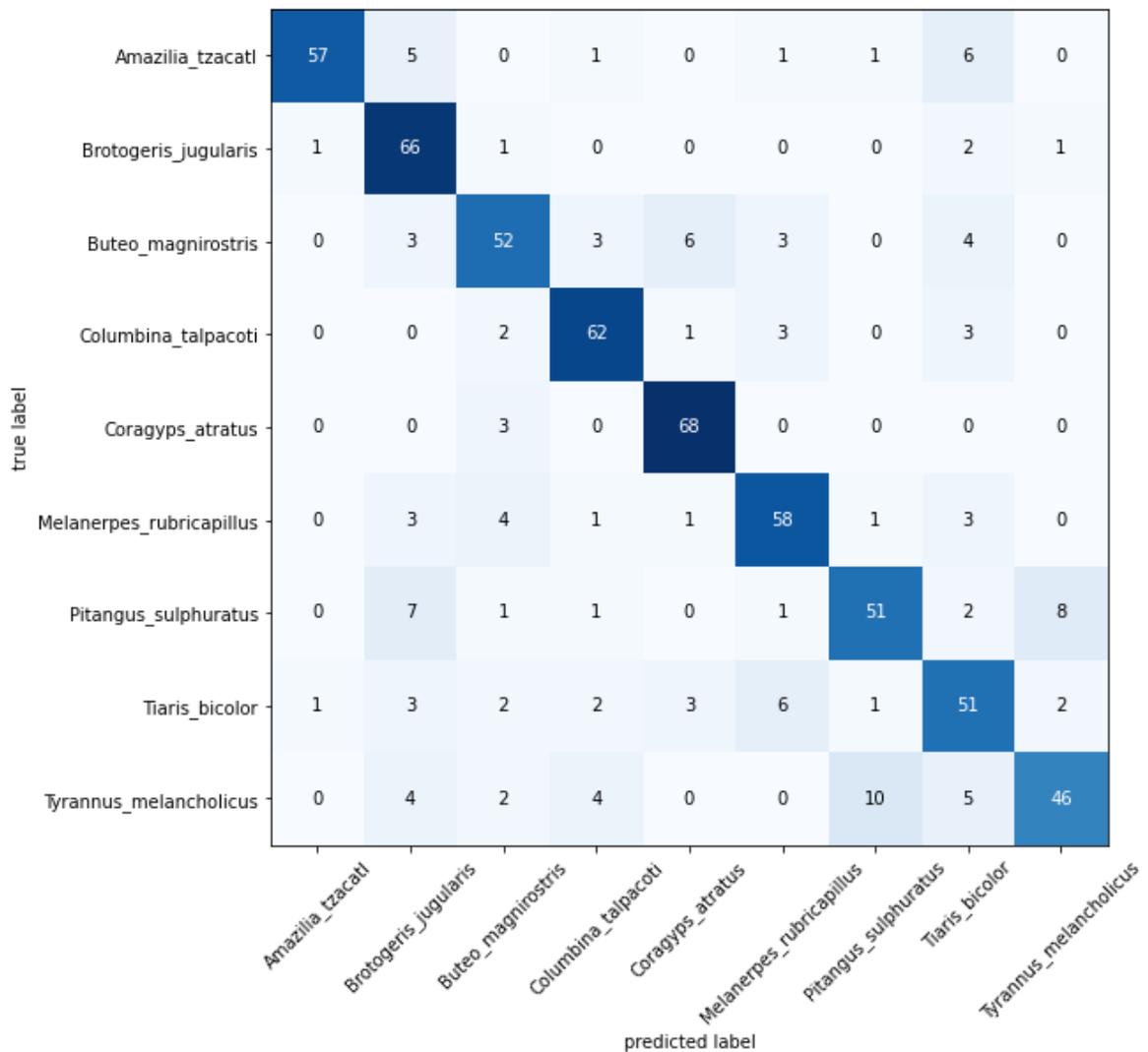
Tabla 11 Plan de Pruebas de Modelo de IA

NÚMERO DE PRUEBA	ÉPOCAS	PASOS POR ÉPOCA	TASA DE APRENDIZAJE	DE ACCURACY	LOSS	REFERENCIA DE ARCHIVO
1	10	10	0,01	0,1643	0,1011	Notebook Prueba Uno
2	50	10	0,01	0,6548	0,0613	Notebook Prueba Dos
3	100	10	0,01	0,7993	0,0425	Notebook Prueba Tres
4	79	15	0,01	0,82	0,0393	Notebook Prueba Cuatro
5	100	15	0,01	0,8771 (0,8905)	0,0285(0,0283)	Notebook Prueba Cinco

Fuente: Autor.

El modelo que se conectará a la aplicación será el de la prueba número 5, el cual tiene 0,87 de exactitud.

Para la evaluación del modelo resultante se tiene la siguiente matriz de confusión la cual es la visualización de los datos reales vs los datos predichos de cada una de las especies. En la diagonal principal tenemos las evaluaciones correctas o llamado verdaderos positivo. Por ejemplo, si nos situamos en la primera casilla de izquierda a derecha y de arriba hacia abajo perteneciente a la clase amazilia tzacatl el numero 57 son los verdaderos positivos de esa especie, el resto de la diagonal serian verdaderos negativos, es decir, esto se decide según la especie en donde estemos situados.



Ahora bien, en la matriz también encontramos falsos positivos y falsos negativos, en donde los falsos positivos son aquellos que se asignaron a una clase pero que no corresponden a esta, para evaluarla se deben mirar de manera vertical, o sea, los datos de la predicción. Siguiendo el ejemplo, en la especie amazilia tzacatl,

tenemos que hizo la predicción que una imagen perteneciente esta se clasificó como la especie brotogeris juglaris, entonces, este sería un falso positivo. Por otro lado, los falsos negativos se evidencian de manera horizontal, es decir, los datos verdaderos de las especies, continuando el ejemplo, en la relación de la especie amazilia tzacatl con la especie brotogeris se tiene un 5, estos son falsos negativos, se clasificaron como negativos para esa clase, pero en realidad si pertenecía a la clase amazilia. Entonces, los falsos positivos y negativos son en realidad evaluaciones incorrectas del modelo.

Dicho esto, se presentaron las siguientes métricas por cada especie: precisión, recall (exhaustividad o sensibilidad) y el F1-score.

	Precision	Recall	F1-score	Support
0	0.9661	0.8028	0.8769	71
1	0.7253	0.9296	0.8148	71
2	0.7761	0.7324	0.7536	71
3	0.8378	0.8732	0.8552	71
4	0.8608	0.9577	0.9067	71
5	0.8056	0.8169	0.8112	71
6	0.7969	0.7183	0.7556	71
7	0.6711	0.7183	0.6939	71
8	0.8070	0.6479	0.7188	71
Total	0.8051	0,7996	0,7985	

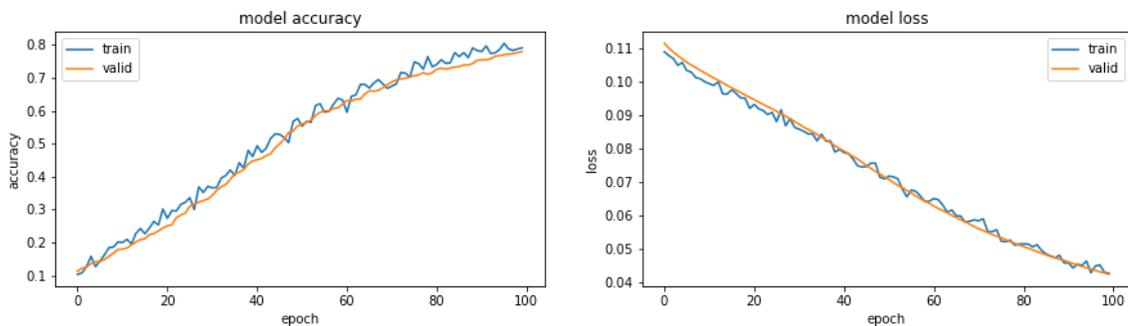
La precisión es la división entre los verdaderos positivos y los verdaderos positivos más los falsos positivos, es decir se evalúa la predicción. Por otro lado, el recall es la división entre los verdaderos positivos y los verdaderos positivos más los falsos negativos, es decir se evalúan los datos reales. Entre tanto el f1-score es un promedio armónico entre precisión y recall. Por lo tanto, dado que el que f1-score total está muy cercano a 1, se puede decir que es un buen modelo, dado que la meta era mejorar el 70% que tenía el modelo inicial.

De igual importancia se tiene el overfitting o sobre-entrenamiento, el cual se presenta cuando el modelo no es capaz de generalizar adecuadamente nuevos datos, es decir, cuando se utilice con datos que no fueron datos de entrenamiento, el modelo no será capaz de clasificarlo de forma adecuada. Esto se puede

observar en la gráfica de pérdida de entrenamiento y validación, cuando la brecha entre ellas es muy grande.

En la ilustración 70 se presenta la gráfica de pérdida y exactitud, nos centraremos en la de loss (pérdida) en la parte derecha, encontramos pérdida entrenamiento y validación, como se puede observar, las graficas convergen de con valores similares, y no existe una brecha grande entre ellas que afirme que hay un overfitting, es más se observa que el modelo aprendió frecuentemente.

Ilustración 70 Gráfica de pérdida y exactitud del modelo



Fuente: autor.

Para el entrenamiento del modelo, se utilizó como estrategia configurar un early-stopping para la pérdida de validación por 3 épocas, es decir, si no se encontraba una mejora seguida en tres épocas parara el entrenamiento, esto implicaría que el aprendizaje del modelo ya no sería útil con datos que estén por fuera de los usados en el entrenamiento. Al nunca dispararse el evento, se confirma que el aprendizaje fue constante y significativo durante todo el proceso, sin señales de algún posible overfitting.

9.3. ANÁLISIS DE TEST, PRUEBAS DE FUNCIONALIDAD Y ACEPTACIÓN.

Siguiendo la ficha detalla (ver anexo 4) de las pruebas de funcionalidad y aceptación, se aplicó inicialmente un test con las aves que actualmente puede identificar el prototipo antes de utilizar la aplicación (ver tabla 20) sin que mostrara las respuestas correctas, esto con el fin de evaluar la ayuda que brindaría la aplicación a los usuarios de prueba. Después de utilizar la aplicación se realizó el mismo test, las respuestas se pueden observar en la tabla 13. Finalmente, se aplicó el test de funcionalidad y aceptación para evaluar los casos de uso que debe tener el prototipo.

La información demográfica obtenida para estos test fue la edad y nombre de los usuarios, en la tabla 12 se recopiló la información.

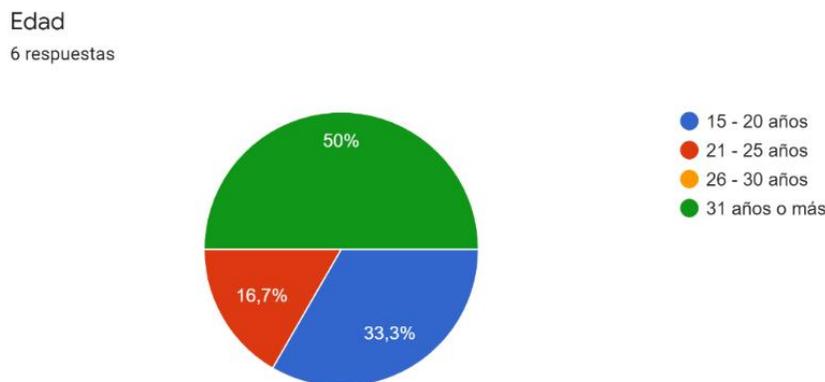
Tabla 12 Información de Usuarios de Prueba

Nombre completo	Correo
Marlon Díaz Jiménez	marlondiji@gmail.com
Andrea Carolina Camacho Julio	acamacho176@unab.edu.co
Ariel Ortiz	gnulion@gmail.com
Feisar Enrique Moreno Corzo	rasief@gmail.com
Fernando Javier Cediél Martínez	lukasced@gmail.com
Lisseth Andrea Hernández Flórez	lissethandreah@gmail.com

Fuente: Autor.

Como se puede analizar en la ilustración 71, el rango de edad de los usuarios que probaron la aplicación fue muy variado, esto con el fin de que tener una percepción específica por cada tipo de edad.

Ilustración 71 Edad de Usuarios de Prueba



Fuente: Autor.

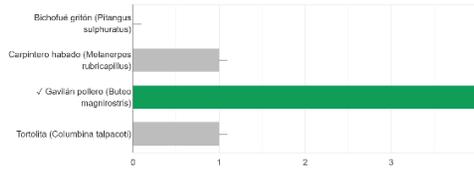
Tabla 13 Respuestas de Test de Pruebas

Pregunta	Respuestas recopiladas test antes	Respuestas recopiladas test después
----------	-----------------------------------	-------------------------------------

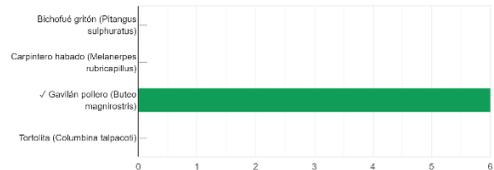


¿Cuál es el nombre de esta ave?

¿Cuál es el nombre de esta ave?
4 de 6 respuestas correctas

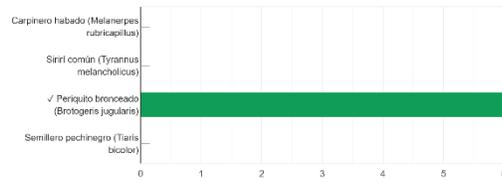


¿Cuál es el nombre de esta ave?
6 de 6 respuestas correctas

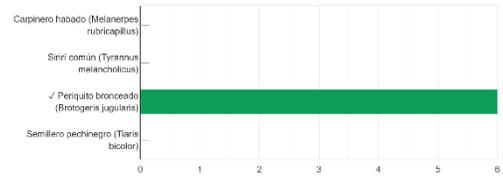


¿Cuál es el nombre de esta ave?

¿Cuál es el nombre de esta ave?
6 de 6 respuestas correctas



¿Cuál es el nombre de esta ave?
6 de 6 respuestas correctas

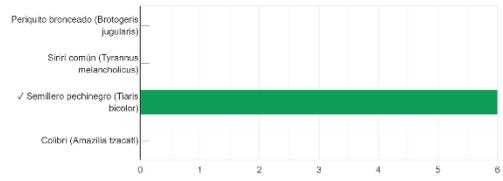


¿Cuál es el nombre de esta ave?

¿Cuál es el nombre de esta ave?
4 de 5 respuestas correctas

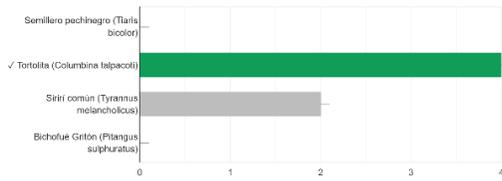


¿Cuál es el nombre de esta ave?
6 de 6 respuestas correctas

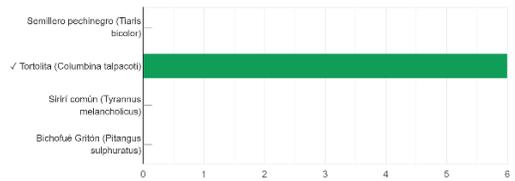


¿Cuál es el nombre de esta ave?

¿Cuál es el nombre de esta ave?
4 de 6 respuestas correctas

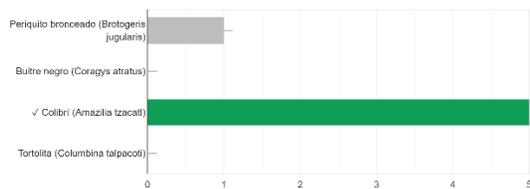


¿Cuál es el nombre de esta ave?
6 de 6 respuestas correctas

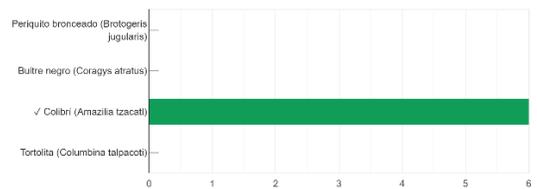


¿Cuál es el nombre de esta ave?

¿Cuál es el nombre de esta ave?
5 de 6 respuestas correctas



¿Cuál es el nombre de esta ave?
6 de 6 respuestas correctas



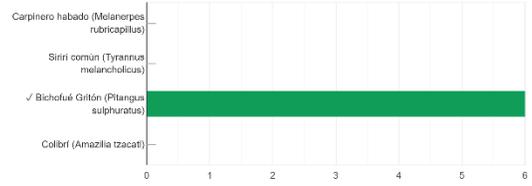


¿Cuál es el nombre de esta ave?

¿Cuál es el nombre de esta ave?
2 de 6 respuestas correctas

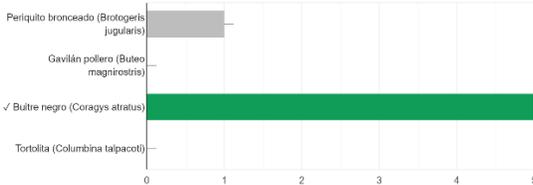


¿Cuál es el nombre de esta ave?
6 de 6 respuestas correctas

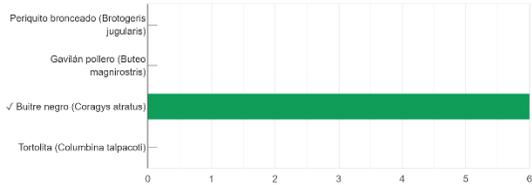


¿Cuál es el nombre de esta ave?

¿Cuál es el nombre de esta ave?
5 de 6 respuestas correctas

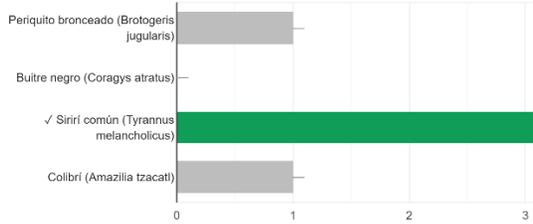


¿Cuál es el nombre de esta ave?
6 de 6 respuestas correctas

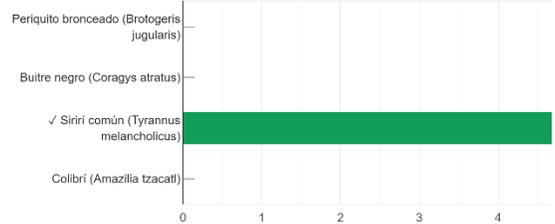


¿Cuál es el nombre de esta ave?

¿Cuál es el nombre de esta ave?
4 de 6 respuestas correctas

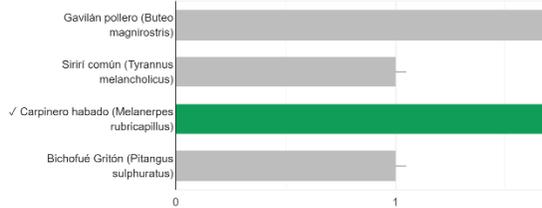


¿Cuál es el nombre de esta ave?
6 de 6 respuestas correctas

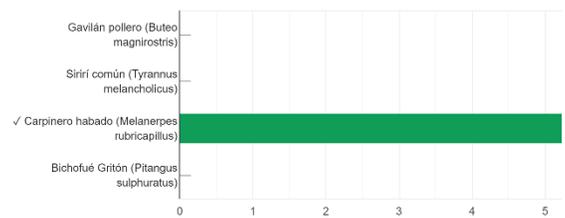


¿Cuál es el nombre de esta ave?

¿Cuál es el nombre de esta ave?
2 de 6 respuestas correctas



¿Cuál es el nombre de esta ave?
6 de 6 respuestas correctas



Fuente: Autor.

En total, los usuarios que realizaron el test antes de utilizar la aplicación obtuvieron una media de 7 de 9 puntos, donde cada punto es una pregunta. Estos

podieron alcanzar esta puntuación porque las aves de las imágenes son muy comunes en el día a día. A continuación, se muestra las respuestas por el número de encuestados antes de utilizar la aplicación.

Ilustración 72 Puntaje Obtenido por Usuarios Antes de Utilizar la Aplicación



Fuente: Autor.

En contraste con los puntajes obtenidos después de utilizar la aplicación, la media fue 9 de 9 puntos, esto también se puede ver reflejado en la tabla 13 por cada pregunta. Por lo tanto, es posible afirmar que el uso de la aplicación y la utilización como herramienta de identificación de aves eleva el conocimiento sobre las especies de la zona metropolitana de Bucaramanga.

ENCUESTA DE FUNCIONALIDAD Y ACEPTACIÓN:

A cada uno de los usuarios se aplicó la encuesta de funcionalidad y aceptación después de utilizar la aplicación (ver anexo 4) (ver tabla 21 y 22) las respuestas obtenidas en la sección de funcionalidad fueron las siguientes, donde cumplió o no cumplió son el número de usuarios que marcaron la opción.

Tabla 14 Respuestas de Encuesta de Funcionalidad

Pregunta	Cumplió	No cumplió
¿La aplicación le permitió ingresar? (Loguearse)	6	0
¿Se pudo registrar en la aplicación?	6	0
¿Pudo visualizar su información en la interfaz "perfil" de la aplicación?	6	0
¿Pudo editar sus datos?	6	0
¿Pudo visualizar la lista de aves que se pueden identificar en AveDex?	6	0

Al darle clic a cualquier item de la lista ¿Le permitió visualizar la información del ave?	6	0
¿Pudo identificar un ave tomando una foto desde su celular?	6	0
¿Pudo identificar un ave eligiendo una foto desde su galería?	6	0
En caso de estar identificando un ave desde una imagen de la galería, ¿Pudo visualizar el mapa y escoger la ubicación de donde vio el ave?	6	0
¿Pudo ver el mapa de aves identificadas por usted?	6	0
¿Pudo visualizar la información del ave al momento de identificar un ave desde las dos opciones (galería o cámara)?	6	0
¿Pudo recuperar la contraseña de su cuenta desde la aplicación?	5	1

Fuente: Autor.

A partir de estos resultados se obtuvo el cumplimiento de los casos de uso, calificados por completamente o parcialmente cumplidos (ver tabla 15).

Tabla 15 Cumplimiento de Casos de Uso

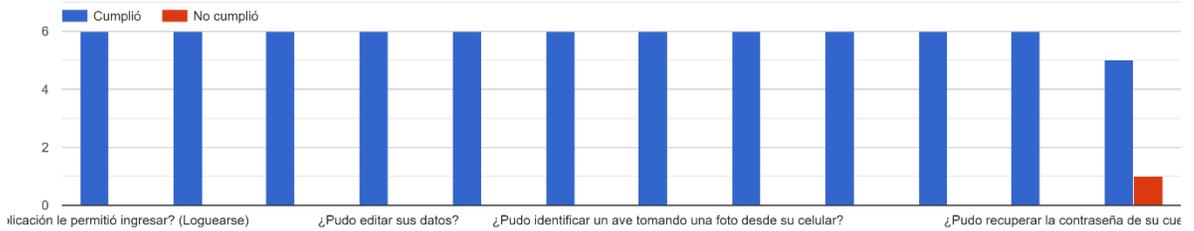
ID	Nombre	Cumplimiento
ECU001	Crud Usuario	Completo
ECU002	Crud Ave	Completo
ECU003	Crud Ubicación	Completo
ECU004	Clasificar Ave	Completo
ECU005	Consultar listado de aves identificadas	Completo
ECU006	Recuperar Contraseña	Parcial
ECU007	Consultar clasificador	Completo

Fuente: Autor.

Se identificó que el caso de uso ECU006 (recuperar contraseña) no pudo ser completado por un usuario de prueba, por lo tanto, se puede inferir que se tiene un cumplimiento del 85.71% de 100%. Por consiguiente, el cumplimiento total es de 6,85 lo que da un porcentaje de 97,86% total de casos de usos cumplidos en el prototipo.

Ilustración 73 Gráfica de Cumplimiento de Funcionalidades

Marque la casilla (Cumplió) , si la aplicación cumplió con la funcionalidad descrita. De lo contrario marque la casilla (No cumplió)



Fuente: Autor.

Ilustración 74 Observaciones de Usuarios Encuesta de Funcionalidad.

El flujo de trabajo es completamente funcional, pero se puede simplificar un poco de cara a versiones futuras, sobre todo en la parte de identificación de aves por cámara. ¡Excelente aplicación y bonito diseño!

La aplicación es intuitiva y cumple con su principal función; identificación de aves.

Muy intuitiva y funcionó super bien :3

La aplicación es muy fácil de usar. Sería bueno que mostrara que está procesando para que el usuario no crea que se bloqueó.

La aplicación está muy bien. Es fácil de usar, intuitiva y mas importante aún, identifica bien a las aves de Bucaramanga.

Me parece una aplicación intuitiva y contiene información acertada, al igual que me gusta que indica las aves de las cuáles se pueden realizar consultas.

Fuente: Autor.

Las observaciones obtenidas de cada uno de los usuarios en cuanto a la experiencia con las funcionalidades fueron las siguientes:

Dado que los usuarios que utilizaron la aplicación estaban en rango de edades diferentes, se puede deducir, que el flujo de las funcionalidades fue acertado.

Además, que el prototipo cumple con el objetivo principal que es la identificación de las aves por medio de imágenes. También, en las observaciones se hace hincapié en que este debería mostrar algún mensaje de que está procesando al momento de identificar.

De acuerdo a la percepción cualitativa al momento de realizar las funcionalidades se tienen las siguientes respuestas, estas hacen parte de la evaluación de aceptación hacia la aplicación, donde bueno, regular y malo tienen el número de puntos por usuarios.

Tabla 16 Respuestas de percepción Cualitativa por Funcionalidades

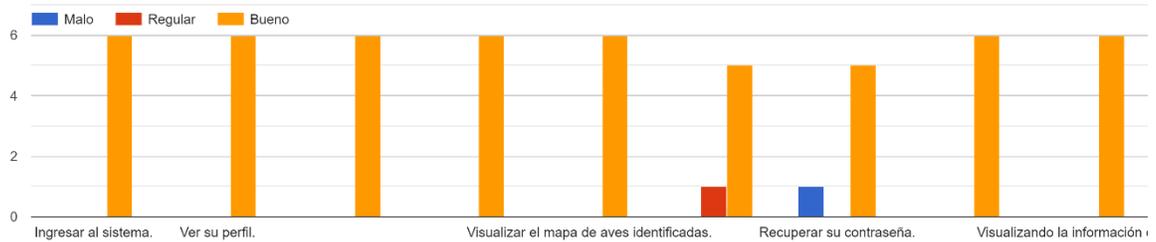
Acción	Bueno	Regular	Malo
Ingresar al sistema.	6	0	0
Ver su perfil.	6	0	0
Identificar un ave, desde una foto tomada de su teléfono.	6	0	0
Identificar un ave, desde una foto elegida de la galería de su celular.	6	0	0
Visualizar el mapa de aves identificadas.	6	0	0
Editar sus datos.	5	1	0
Recuperar su contraseña.	5	0	1
Visualizar la lista de aves que se pueden identificar en AveDex.	6	0	0
Visualizando la información del ave.	6	0	0
Los colores de la aplicación.	6	0	0
Los mensajes en los procesos de la aplicación.	6	0	0
La facilidad en el manejo de la aplicación.	6	0	0
El diseño en general de la aplicación.	6	0	0

Fuente: Autor.

A partir de los resultados de la tabla 16, se puede inferir que un usuario tuvo una experiencia no muy satisfactoria al momento de editar su información y recuperar contraseña. Pero, al momento de realizar las diferentes funcionalidades la experiencia fue buena, fácil e intuitiva.

Ilustración 75 Gráfica de percepción Cualitativa de la Aplicación.

¿Cómo fue su percepción cualitativa al momento de realizar estas acciones?

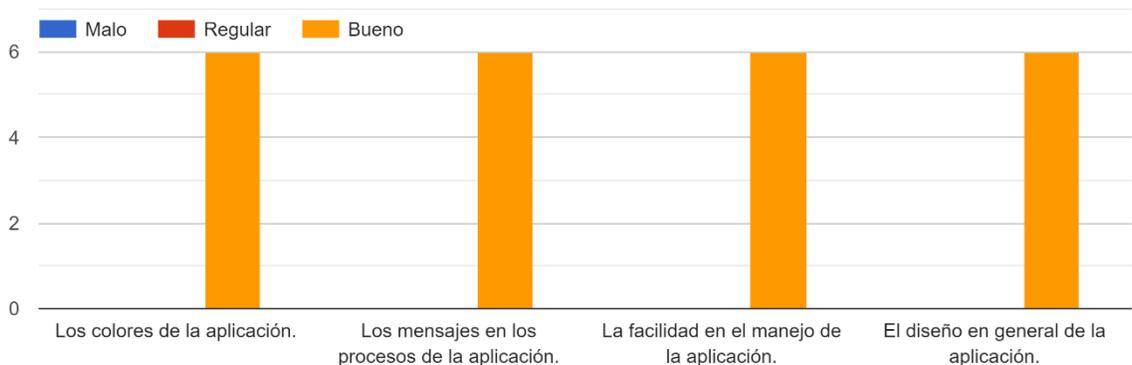


Fuente: Autor.

Por otro lado, se evaluaron diferentes aspectos como: los colores, diseño y mensajes en los procesos de la aplicación en general estos se encuentran en los 4 últimos registros de la tabla 16. A partir de ellos, se concluye que la aplicación tiene una aceptación buena en cuanto a la experiencia de los usuarios, armonía de diseño y colores.

Ilustración 76 Gráfica de percepción de aspectos de Experiencia de Usuario.

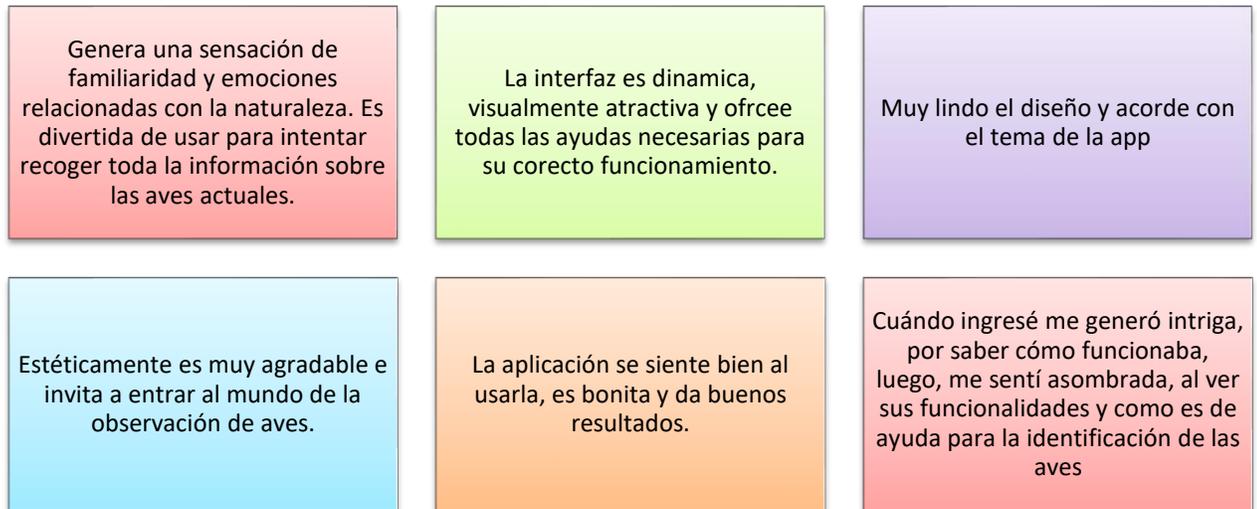
¿Cómo fue su percepción en estos aspectos?



Fuente: Autor.

A continuación, encontrará las observaciones sobre las emociones o sentimientos que experimentaron los usuarios al momento de usar la aplicación.

Ilustración 77 Observaciones de Usuario Encuesta de Percepción



Fuente: Autor.

Finalmente, se puede decir que la aplicación cumplió con su función principal que es la identificación de aves de una forma fácil, con un diseño intuitivo y experiencia de usuario adecuado para los usuarios de la zona metropolitana de Bucaramanga.

10. CONCLUSIONES Y RECOMENDACIONES

A partir del trabajo realizado en el presente proyecto, se permite resaltar la importancia de la educación y del cuidado sobre las aves de la zona metropolitana. Durante todo este proceso, se obtuvieron diferentes habilidades como la extracción, diseño y programación de las funcionalidades.

En el desarrollo de la investigación de estado del arte y proyectos semejantes al propuesto se determinó que la arquitectura para el entrenamiento del modelo adecuado era la ResNet, se confirma que se obtuvo un modelo con una precisión del 0,87, lo cual mejora el modelo inicial de este proyecto que era del 70%.

Por otro lado, sobre el conjunto de imágenes recopiladas, se concluye que es un número de datos aceptables para la construcción del modelo clasificador del presente prototipo, ya que existe equivalencia entre todas las especies, por lo cual el modelo no pudo sobre - aprender sobre especies específicas. Además, se obtuvo un porcentaje adecuado al momento de dividir el dataset entre entrenamiento, validación y testeo. En cuanto al algoritmo de semejanza utilizado para identificar patrones entre imágenes de las aves no se obtuvieron buenos resultados, esto se realizaba con el fin de identificar que especies de aves eran semejantes que podía permitir que el clasificador se confundiera, pero dado que el algoritmo recorre la imagen pixel por pixel para encontrar puntos clave entre ellas, dado que, en términos de pixel, cada imagen es totalmente diferente de otra, aunque contengan formas parecidas.

Finalmente, del análisis de las pruebas de aceptación y funcionalidad se concluye que el prototipo construido cumple con la función principal que es identificar aves del área metropolitana de Bucaramanga. Además, que la aplicación sirve de herramienta para obtener información de las aves, esto se afirma reflejado de los resultados de los test realizados a los usuarios de pruebas antes y después de probar la aplicación. El diseño, colores y experiencia hacia al usuario fue totalmente acertada. A pesar de, que el modelo predictor se encontraba local al momento de realizar las pruebas y las conexiones a la base de datos eran limitadas se pudo cumplir el objetivo general del presente proyecto. Asimismo, la pandemia del COVID-19 no permitió que la aplicación fuera testeada al aire libre o en zonas verdes.

TRABAJOS FUTUROS.

Se espera que el sistema web y el modelo clasificador sean desplegados en un dominio público desde donde pueda acceder la aplicación para hacer las predicciones. Además, de que la base de datos se pueda obtener con un mayor número de conexiones a ella, ya que actualmente tiene 20, lo que limita la cantidad de usuarios que quieran utilizar la aplicación.

También, el modelo debería ser accedido localmente a la aplicación, es decir, que venga incrustada en esta. En la actualidad existe tensorflow lite, para correr modelos de predicción de manera offline, pero, dado que la tecnología utilizada para el desarrollo fue flutter se obtuvieron obstáculos al momento de implementarlo, ya que no está lo suficientemente desarrollado para implementarlos en esta. Realizar esto, no limitaría a que los usuarios que la utilicen necesiten una conexión a internet estable para ingresar y realizar cualquier funcionalidad.

Por otro lado, el modelo clasificador debería ser reentrenado con un mayor número de aves, con una red neuronal mixta que permita una mayor estabilidad y precisión para prevenir que este se pueda confundir entre aves muy parecidas al momento de predecir.

Finalmente, el diseño hacia la experiencia de usuario debe estar en constante actualización, para generar una experiencia más satisfactoria al momento de utilizar la aplicación.

BIBLIOGRAFÍA

- Acevedo-Charry, O. C.-P.-Q. (2020). *Avifauna Colombiana*. Obtenido de Avifauna Colombiana: <http://reporte.humboldt.org.co/biodiversidad/2019/cap1/103/#seccion5>
- Akash, K. (2019). *GitHub*. Obtenido de GitHub: https://github.com/AKASH2907/bird_species_classification
- AQUAE Fundación. (2020). *AQUAE Fundación*. Obtenido de AQUAE Fundación: <https://www.fundacionaquae.org/el-cambio-climatico-reduce-el-tamano-de-las-aves/>
- Arispe, M., & Collarana, D. (5 de Marzo de 2017). *SCIELO*. Obtenido de SCIELO: http://www.scielo.org.bo/scielo.php?pid=S1683-07892017000100007&script=sci_arttext
- Avendaño, J., Bohórquez, C., Roselli, L., Arzuza-Buelvas, D., Estela, F. A., Cuervo, A. M., . . . Renjifo, L. M. (2017). Lista de chequeo de las aves de Colombia: Una síntesis del estado del conocimiento desde Hilty & Brown (1986). *Ornitología Colombiana*, 1-83.
- Ayerbe Quiñones, F. (2019). *Guía ilustrada de la Avifauna colombiana*. Bogotá: WSC.
- Berg, T., Liu, J., Woo Lee, S., Alexander, M., Jacobs, D., & Belhumeur, P. (2014). *IEEE Conference on Computer Vision and Pattern Recognition*. Obtenido de IEEE Conference on Computer Vision and Pattern Recognition: <https://ieeexplore-ieee.org/aure.unab.edu.co/document/6909656>
- Bhatnagar, A. (2020). *GitHub*. Obtenido de GitHub: https://github.com/aniket03/self_supervised_bird_classification
- Brownlee, J. (5 de Julio de 2019). *Machine Learning Mastery*. Obtenido de Machine Learning Mastery: <https://machinelearningmastery.com/what-is-computer-vision/>
- Cediel Martínez, F. (16 de Febrero de 2020). Investigación avifauna colombiana. (M. A. Mosquera Moreno, Entrevistador)
- Centro Integrado Politécnico "ETI". (s.f.). *Centro Integrado Politécnico "ETI"*. Obtenido de Centro Integrado Politécnico "ETI": <http://www.etitudela.com/celula/downloads/visionartificial.pdf>
- Cheng, L. Y. (2019). *GitHub*. Obtenido de <https://github.com/lychengr3x/Bird-Species-Classification-Using-Transfer-Learning>
- Contreras, F. (2016). *docPlayer.es*. Obtenido de docPlayer.es: <https://docplayer.es/74142729-Whitepaper-introduccion-a-machine-learning-por-fabian-a-contreras-arquitecto-de-soluciones-machine-learning-sunqu-2016.html>
- Cornell Lab of Ornithology. (2014). *NABirds Dataset*. Obtenido de NABirds Dataset: <https://dl.allaboutbirds.org/nabirds>
- Cornell lab Of Ornithology. (27 de Febrero de 2020). *Play Store*. Obtenido de Play Store: https://play.google.com/store/apps/details?id=com.labs.merlinbirdid.app&hl=es_CO
- Dart Dev. (2011). *Dart*. Obtenido de Dart: <https://dart.dev/brand>

- decide*. (2019). Obtenido de decide: <https://decidesoluciones.es/arquitectura-de-microservicios/>
- DeepAI. (2020). *DeepAI*. Obtenido de DeepAI: <https://deepai.org/machine-learning-glossary-and-terms/batch-normalization>
- Deloitte. (2017). *Deloitte Colombia - Consumo móvil en Colombia*. Obtenido de Deloitte Colombia - Consumo móvil en Colombia: [https://www2.deloitte.com/content/dam/Deloitte/co/Documents/technology-media-telecommunications/Consumo%20movil\(VF1\).pdf](https://www2.deloitte.com/content/dam/Deloitte/co/Documents/technology-media-telecommunications/Consumo%20movil(VF1).pdf)
- Deloitte. (2019). *Deloitte Colombia - Consumo móvil en Colombia*. Obtenido de Deloitte Colombia - Consumo móvil en Colombia: <https://www2.deloitte.com/content/dam/Deloitte/co/Documents/consumer-business/Reporte%20consumo%20movil%202019.pdf>
- Durán, J. (22 de Enero de 2020). *Medium*. Obtenido de Medium: <https://medium.com/metadatos/qu%C3%A9-es-la-transferencia-de-aprendizaje-y-c%C3%B3mo-aplicarla-a-tu-red-neuronal-e0e120156e40>
- Entre líneas*. (25 de Noviembre de 2008). Obtenido de <http://is3ados.blogspot.com/2008/11/los-ciclos-de-vida-del-software-ii.html>
- Erenli, K. (2013). *IEEEExplore. 2012 15th International Conference on Interactive Collaborative Learning (ICL)*. Villach, Austria: IEEE. Obtenido de IEEEExplore: <https://ieeexplore-ieee-org.aure.unab.edu.co/document/6402106>
- Escuela Superior de Ingeniería y Tecnología. (25 de Abril de 2005). *Escuela Superior de Ingeniería y Tecnología*. Obtenido de Escuela Superior de Ingeniería y Tecnología: cc.etsii.ull.es/ftp/antiguo/VISORD/temario/uno.doc
- Flask. (2010). *FLASK*. Obtenido de FLASK: <https://flask.palletsprojects.com/en/1.1.x/>
- Flutter-Dev. (2011). *Flutter*. Obtenido de Flutter: <https://flutter.dev/brand>
- Flutter-dex. (2020). *Flutter*. Obtenido de Flutter: <https://flutter.dev/docs/resources/architectural-overview>
- Generalitat de Catalunya. (Febrero de 2012). *Visión Artificial Aplicación práctica de la visión artificial en el control de procesos industriales*. Obtenido de Visión Artificial Aplicación práctica de la visión artificial en el control de procesos industriales: http://visionartificial.fpcat.cat/wp-content/uploads/UD_1_didac_Conceptos_previos.pdf
- González Diez, D. (25 de Julio de 2019). *Universidad de Valladolid*. Obtenido de Universidad de Valladolid: <https://core.ac.uk/download/pdf/250406384.pdf>
- González, A. (2019). *Cleverdata*. Obtenido de Cleverdata: <https://cleverdata.io/que-es-machine-learning-big-data/>
- Google Developers. (20 de Febrero de 2020). *Google Developers*. Obtenido de Google Developers: <https://developers.google.com/machine-learning/crash-course/introduction-to-neural-networks/anatomy?hl=es-419>
- Grinberg, M. (2018). *Flask Web Development Developing Web Applications With Python*. Estados Unidos: O'Reilly Media, Inc. Obtenido de https://drive.google.com/drive/folders/1mMpq2WluJKz08oQV_D80llntK26wf9tH

- Huang, C., Luo, B., Tang, L., Liu, Y., & Ma, J. (2013). *IEEE International Conference on Communications, Circuits and Systems (ICCCAS)*. Obtenido de IEEE International Conference on Communications, Circuits and Systems (ICCCAS): <https://ieeexplore-ieee-org.aure.unab.edu.co/document/6765346>
- Islam, M., Tasnim, N., & Bhatta Shuvo, S. (2019). *IEEE XPLORE*. Obtenido de IEEE XPLORE: <https://ieeexplore-ieee-org.aure.unab.edu.co/document/8944403>
- Iyawa, G., Masikara, W., Osakwe, J., & Oduor, C. (18 de Julio de 2019). *IEEEXplore*. Obtenido de IEEEXplore: <https://ieeexplore-ieee-org.aure.unab.edu.co/document/8764865/authors>
- Jungmo, A., Jeongyeup, P., & JeongGil, K. (2016). *IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications*. Obtenido de IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications: <https://ieeexplore-ieee-org.aure.unab.edu.co/document/7579939>
- Junta de Andalucía. (2017). *Ciclo de vida basado en prototipos*. Obtenido de http://agrega.juntadeandalucia.es/repositorio/20022017/6b/es-an_2017022012_9122843/53_ciclo_de_vida_basado_en_prototipos.html
- Kang, M.-S., & Hong, K.-S. (2018). *IEEE International Conference on Information and Communication Technology Convergence (ICTC)*. Obtenido de IEEE International Conference on Information and Communication Technology Convergence (ICTC): <https://ieeexplore-ieee-org.aure.unab.edu.co/document/8539463>
- Karim, R. (Julio de 2019). *Towards Data science*. Obtenido de Towards Data science: <https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d#e971>
- Kumar, A. (2020). *GitHub*. Obtenido de GitHub: https://github.com/birdsiitmandi/birds_image_classification
- Lee, S., Lee, M., Jeon, H., & Smith, A. (2019). *IEEE 6th International Conference on Control, Decision and Information Technologies (CoDIT)*. Obtenido de IEEE 6th International Conference on Control, Decision and Information Technologies (CoDIT): <https://ieeexplore-ieee-org.aure.unab.edu.co/document/8820331>
- Lemley, J., Bazrafkan, S., & Corcoran, P. (Abril de 2017). *ResearchGate*. Obtenido de ResearchGate: https://www.researchgate.net/publication/316748306_Transfer_Learning_of_Temporal_Information_for_Driver_Action_Classification/citations
- Liu, Y., Sun, P., Highsmith, M., Wergeles, N., Sartwell, J., Raedeke, A., . . . Shang, Y. (2018). *IEEE Third International Conference on Data Science in Cyberspace (DSC)*. Obtenido de IEEE Third International Conference on Data Science in Cyberspace (DSC): <https://ieeexplore-ieee-org.aure.unab.edu.co/document/8411873>
- Lohr, D. (2019). *GitHub*. Obtenido de GitHub: <https://github.com/djl70/cv-bird-id>

- Malav, B. (16 de Diciembre de 2017). *Medium*. Obtenido de Medium: <https://medium.com/startlovingyourself/microservices-vs-monolithic-architecture-c8df91f16bb4>
- Marini, A., Facon, J., & Koerich, A. (2013). *IEEE International Conference on Systems, Man, and Cybernetics*. Obtenido de IEEE International Conference on Systems, Man, and Cybernetics: <https://ieeexplore-ieee.org.aure.unab.edu.co/document/6722493>
- Marini, E. (Octubre de 2012). Obtenido de <https://www.linuxito.com/docs/el-modelo-cliente-servidor.pdf>
- Martinez Castillo, R. (2012). Ensayo crítico sobre educación ambiental. *Diálogos educativos*, 74-104.
- Ministerio de educación. (1962). *DECRETO 45 DE 1962*. Obtenido de DECRETO 45 DE 1962: https://www.mineducacion.gov.co/1759/articles-103679_archivo_pdf.pdf
- Mosquera Moreno, M. A. (Agosto de 2020). Conocimiento sobre la avifauna en la zona metropolitana de Bucaramanga. Bucaramanga, Colombia. Obtenido de https://docs.google.com/spreadsheets/d/1c8StBohC7Kc0RydQF36NAI_DFHGfJ__Fh5yhCx-WgOU/edit?usp=sharing
- Mufid, M. R., Al Rasyid, M. H., Basofi, A., Rokhim, A., & Rochimansyah, I. F. (2019). *IEEE Xplore 2019 International Electronics Symposium (IES)*. Obtenido de IEEE Xplore 2019 International Electronics Symposium (IES): <https://ieeexplore-ieee.org.aure.unab.edu.co/document/8901656>
- Nawaz, S., Calefati, A., Caraffini, M., Landro, N., & Gallo, I. (2019). *IEEE International Conference on Image and Vision Computing New Zealand (IVCNZ)*. Obtenido de IEEE International Conference on Image and Vision Computing New Zealand (IVCNZ): <https://ieeexplore-ieee.org.aure.unab.edu.co/document/8960960>
- Niemi, J., & Tanttu, J. (2019). *IEEE International Symposium ELMAR*. Obtenido de IEEE International Symposium ELMAR: <https://ieeexplore-ieee.org.aure.unab.edu.co/document/8124482>
- Ordieres-Meré, J., Castejón Limas, M., Martínez de Pisón Ascacibar, F. J., Alba-Elías, F., González-Marcos, A., Pernía-Espinoza, A., & Vergara, E. (2006). *Técnicas y algoritmos básicos de visión artificial*. España: Universidad de La Rioja. Servicio de Publicaciones.
- Pérez Roldán, I. (Julio de 2019). *Archivol Digital Universidad Politécnica de Madrid*. Obtenido de Archivol Digital Universidad Politécnica de Madrid: http://oa.upm.es/56163/1/TFG_ISMAEL_PEREZ_ROLDAN.pdf
- Pineda, L. (2009). *Inteligencia Artificial*. Obtenido de Inteligencia Artificial: http://turing.iimas.unam.mx/~luis/cursos/IA09/slides/s1_inteligencia_artificial.pdf
- PlantSnap, Inc. (28 de Febrero de 2020). *Play Store*. Obtenido de Play Store: https://play.google.com/store/apps/details?id=com.fws.plantsnap2&hl=es_COhttps://play.google.com/store/apps/details?id=com.fws.plantsnap2&hl=es_CO
- Qiao, B., Zhou, Z., Yang, H., & Cao, J. (2017). *IEEE First International Conference on Electronics Instrumentation & Information Systems (EIIS)*. Obtenido de IEEE First

- International Conference on Electronics Instrumentation & Information Systems (EIS): <https://ieeexplore-ieee-org.aure.unab.edu.co/document/8298548>
- RAE. (2020). Obtenido de <https://dle.rae.es/avifauna>
- Revista semana. (11 de Mayo de 2018). *Semana Sostenible*. Obtenido de Semana Sostenible: <https://sostenibilidad.semana.com/medio-ambiente/articulo/aviturismo-tiene-un-enorme-potencial-economico-en-colombia/40965>
- Sancho Peña, M. (28 de Noviembre de 2017). *Repositorio Institucional de la Universidad de Alicante*. Obtenido de Repositorio Institucional de la Universidad de Alicante: <http://rua.ua.es/dspace/handle/10045/71527>
- Shahana, Z. (2018). *GitHub*. Obtenido de GitHub: <https://github.com/zahan97/Bird-Species-Classification>
- Siwalu Software GmbH. (6 de Marzo de 2020). *Play Store*. Obtenido de Play Store: https://play.google.com/store/apps/details?id=com.siwalusoftware.dogscanner&hl=es_CO
- Van Horn, G., Branson, S., Farrell, R., Haber, S., Barry, J., Ipeirotis, P., . . . Bolongie, S. (2015). *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Obtenido de IEEE Conference on Computer Vision and Pattern Recognition (CVPR): <https://ieeexplore-ieee-org.aure.unab.edu.co/document/7298658>
- Vázquez Rodríguez, V. (2019). *Desarrollo de aplicaciones móviles multiplataforma con Flutter*. Obtenido de Desarrollo de aplicaciones móviles multiplataforma con Flutter: http://repositorio.ual.es/bitstream/handle/10835/8010/TFG_VAZQUEZ%20RODRIGUEZ,%20VICTOR.pdf?sequence=1
- Vega Arias, J. (Junio de 2019). *Universitat Oberta de Catalunya*. Obtenido de Universitat Oberta de Catalunya: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/98767/6/jvegaarTFM0619memoria.pdf>
- Veyssier, L. (2020). *GitHub*. Obtenido de GitHub: https://github.com/LaurentVeyssier/Bird_Classifier_Tensorflow_Colab_Notebook
- Vlvek, K. (22 de Julio de 2019). *Analytics Insight*. Obtenido de Analytics Insight: <https://www.analyticsinsight.net/state-computer-vision-now-future/>
- Wah, C., Branson, S., Welinder, P., Perona, P., & Belongie, S. (2011). *The Caltech-UCSD Birds-200-2011 Dataset*. Obtenido de The Caltech-UCSD Birds-200-2011 Dataset.: http://www.vision.caltech.edu/visipedia/papers/CUB_200_2011.pdf
- Wei, X.-S., Wang, P., Liu, L., & Wu, J. (2019). *IEEE Transactions on Image Processing, vol. 28, no. 12, pp. 6116-6125*. Obtenido de IEEE Transactions on Image Processing, vol. 28, no. 12, pp. 6116-6125: <https://ieeexplore-ieee-org.aure.unab.edu.co/document/8752297>
- Zazo Millán, C. (15 de Octubre de 2019). *Migración de aplicaciones Android hacia Flutter, un framework para desarrollo de apps multiplataforma*. Obtenido de Migración de aplicaciones Android hacia Flutter, un framework para desarrollo de apps multiplataforma: <http://hdl.handle.net/10251/128486>

- Zhang, X., Xiong, H., Zhou, W., Lin, W., & Tian, Q. (2016). *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Obtenido de IEEE Conference on Computer Vision and Pattern Recognition (CVPR): <https://ieeexplore-ieee-org.aure.unab.edu.co/document/7780497>
- Zheng, H., Fu, J., Mei, T., & Luo, J. (2017). *IEEE International Conference on Computer Vision (ICCV)*. Obtenido de IEEE International Conference on Computer Vision (ICCV): <https://ieeexplore-ieee-org.aure.unab.edu.co/document/8237819>

**PROTOTIPO DE UN SISTEMA DE VISIÓN ARTIFICIAL BASADO EN
TÉCNICAS DE APRENDIZAJE DE MÁQUINA PARA EL RECONOCIMIENTO DE
AVES DEL ÁREA METROPOLITANA DE BUCARAMANGA**

ANEXO 1

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA-UNAB
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA
2021

DESARROLLO

A partir de los artículos revisados previamente citados en el estado del arte del documento principal, se obtuvo información relevante para la definición de la arquitectura de la red a entrenar para la predicción de aves por imágenes. La información se encuentra clasificada de la siguiente forma: nombre del algoritmo clasificador, nombre del conjunto de datos utilizado y lenguaje de programación utilizado (ver tabla 17).

Tabla 17 Cuadro de Clasificación

NÚMERO DEL ARTÍCULO	NOMBRE DEL ALGORITMO CLASIFICADOR	NOMBRE DEL CONJUNTO DE DATOS	LENGUAJE
1	<p>CNN</p> <p>A. Métodos de detección de objetos</p> <ul style="list-style-type: none"> ● Detector MultiBox de disparo único (SSD) ● Etapa única sin cabeza (SSH) ● You Only Look Once (YOLO) ● Tiny Face <p>B. Métodos de segmentación de instancias</p> <ul style="list-style-type: none"> ● U-Net ● La máscara R-CNN 	LBAI (Little Birds in Aerial Imagery)	Python
2	MobileNet (CNN) (tensorflow)	Creación propia (Bangladesh)	Python
3	SVM Logistic Regression Naive Bayes	James Reserve bird	Python
4	One-VS-Most: realizaron un entrenamiento omitiendo del conjunto de entrenamiento todas las imágenes de especies similares.	Birdsnap (500 especies de estados unidos)	Sin información

5	Red neuronal convolucional de atención múltiple (MA-CNN)	CUB-Birds, FGVC-Aircraft Stanford-Cars	Python
6	Red neuronal convolucional (CNN): normalizada por canal cruzado y la capa de ReLU SVM	Creación propia (Finlandia)	Python
7	GMS: graphical model with Saliency SVM: support vector machine MAP (medida de máxima probabilidad posterior)	Caltech-UCSD Birds 200	Python
8	Primero descomponen los colores RGB SVM Algoritmo de optimización mínima secuencial de Platt	CUB-200	Python
9	CNN: descriptores localizados a través de Vector de Fisher con ponderación espacial (SWFV-CNN)	CUB-200	Python
10	CNN profunda (DCNN) con tensorflow: GoogLeNet, VGGNet, ResNet	Creación propia (Corea)	Python
11	TensorFlow GMM: mezcla gaussiana adaptativa. V3 MobileNet NASNet ResNet	Caltech-UCSD Birds-200 NABirds FGVC-Aircraft IMAGENET	Python
12	CNN	NABirds CUB-200 IMAGENET	Python
13	Bilineal CNN.	cub birds	Python
14	Bilinear CNN	Caltech-UCSD	Python

	NTS-NET: red navigator entrenada	Birds	
15	Arbol de decision-SVM	Caltech-UCSD Birds	Python

Fuente: Autor.

Complementando con lo descrito en el estado del arte se pueden observar diferentes enfoques, desde la clasificación por partes hasta la eliminación de imágenes de aves con similitudes. Sin embargo, se encontró que el patrón de algoritmos utilizados es la red neuronal convolucional. Por lo anterior, se realizó una revisión de proyectos publicados en GitHub, pero, esta vez se le dio un enfoque más profundo, a partir de un artículo que ilustraba 10 arquitecturas CNN (Karim, 2019) se determinó el modelo/arquitectura, tamaño, accuracy obtenido, parámetros y profundidad de la red, a partir de estos datos se clasificaron los proyectos encontrados (ver tabla 19).

- 1.1. Descripción de proyectos:
 - 1.1.1. Bird species classification (Akash, 2019): es una implementación del desafío de clasificación de especies de aves presentado por IIT Mandi en ICCVIP Conference'18 en Python3 y keras con backend de Tensorflow. La arquitectura consta de modelos Mask R-CNN e ImageNet de extremo a extremo. Los modelos de imageNet utilizados son Inception V3 e Inception ResNet v2. Fue desarrollado especialmente para aves del Himalaya.
 - 1.1.2. Self supervised learning for fine grained classification in case of bird species identification (Bhatnagar, 2020): es una implementación que explora el aprendizaje auto supervisado para la clasificación de especies de aves por medio de un sistema en el que el objeto (la imagen o ave) esta formado por partes y de esa forma realizar un entrenamiento eliminando la dependencia de los modelos pre-entrenados de ImageNet.
 - 1.1.3. Automatic Identification of Bird Species Using Computer Vision (Lohr, 2019): es el proyecto final en visión por computadora en la Universidad Estatal de Texas, fue realizado mediante python y keras.
 - 1.1.4. Bird Species Classification (Shahana, 2018): es un proyecto que clasifica la especie de un Pájaro usando una red neuronal en tensorflow.
 - 1.1.5. Bird Classifier Tensorflow Colab notebook (Veyssier, 2020): fue desarrollado en tensorflow utilizando un modelo previamente entrenado

de Tensorflow Hub con la arquitectura de MobileNet V2 y ejecutándose en Google Colab.

- 1.1.6. Birds image classification (Kumar, 2020): es un proyecto de clasificación de imágenes de aves detallada utilizando el conjunto de datos de Caltech UCSD Birds, desarrollado en el marco de keras como en el de Pytorch.
- 1.1.7. Bird Species Classification Using Transfer Learning (Cheng, 2019): implementa la clasificación de especies de aves mediante el aprendizaje por transferencia, utilizando VGG16 y Resnet18.

Analizando las dos tablas se observa que lo más utilizado para solucionar este problema en estudios que anteceden el proyecto son las redes neuronales convolucionales, destacando de esta la arquitectura ResNet, Inceptionv3 y VGG16 las cuales tienen un porcentaje de accuracy lo suficientemente altos para dar una precisión aceptable. Dado esto, la arquitectura definida para realizar pruebas de entrenamiento y validación será la ResNet, en caso de que no dé resultados positivos se espera utilizar las otras dos dichas anteriormente. Finalmente, se hará el desarrollo utilizando como backend Tensorflow y Keras.

Tabla 18 Clasificación de Proyectos por Arquitecturas CN

MODELO	TAMAÑO	ACCURACY TOP 1	ACCURACY TOP 5	PARÁMETROS	PROFUNDIDAD	¿EN CUÁLES APLICAN?
VGG16	528 MB	0,713	0,901	138,357,544	23	<ul style="list-style-type: none"> • Proyecto descrito en el 1.1.1. • Proyecto descrito en el 1.1.7
INCEPTION V3	92 MB	0,779	0,937	23,851,784	159	<ul style="list-style-type: none"> • Proyecto descrito en el 1.1.1
RESNET 50	98 MB	0,749	0,921	25,636,712	-	<ul style="list-style-type: none"> • Proyecto descrito en el 1.1.1: utilizando v2, obteniendo un accuracy de 97,29 %. • Proyecto descrito en el 1.1.2: utilizando 18, obteniendo un accuracy de 65-77% • Proyecto descrito en el 1.1.3. • Proyecto descrito en el 1.1.7: utilizando resnet 18.
XCEPTION	88 MB	0,790	0,945	22,910,480	126	<ul style="list-style-type: none"> • Proyecto descrito en el 1.1.6
INCEPTION RESNET V2	215 MB	0,803	0,953	55,873,736	572	-
RESNEXT 50	96 MB	0,777	0,938	25,097,128	-	-

Fuente: Autor

**PROTOTIPO DE UN SISTEMA DE VISIÓN ARTIFICIAL BASADO EN
TÉCNICAS DE APRENDIZAJE DE MÁQUINA PARA EL RECONOCIMIENTO DE
AVES DEL ÁREA METROPOLITANA DE BUCARAMANGA**

ANEXO 2

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA-UNAB
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA
2020**

DESARROLLO.

El presente documento presenta el desarrollo de las actividades 4 “*Elaboración y aplicación de entrevista a usuario normal y especializado*” y 5 “*Análisis de diferentes soluciones tecnológicas en la clasificación de imágenes*” del cronograma propuesto para el proyecto. Encontrará la información sobre la entrevista realizada a los usuarios y la revisión de aplicaciones afines. Estas actividades tenían como fin la obtención de los requerimientos funcionales y no funcionales que tendría la aplicación móvil. Además, el análisis de las aplicaciones, permitieron un primer diseño de la aplicación final.

1. Entrevista:

Inicialmente, se redactó una serie de preguntas globales que determinaran algunos requerimientos. A continuación, se encuentra el guion aplicado a cada uno de los entrevistados, siendo el entrevistador la estudiante María Angélica Mosquera Moreno.

“Hola, buen día, tarde o noche; mucho gusto, mi nombre es María Angélica Mosquera, estudiante de ingeniería de sistemas de octavo semestre. Esta entrevista se realiza con el fin de obtener requerimientos funcionales y no funcionales para desarrollar una aplicación móvil para el reconocimiento de aves en la zona metropolitana de Bucaramanga.

Introducción: con este proyecto se busca desarrollar una aplicación móvil práctica para que cualquier persona pueda identificar un ave por medio de la cámara de su celular. La idea es que desde esta app usted pueda obtener la información de un ave por medio de un clasificador basado en inteligencia artificial. Además, de incentivarlo a que la utilice por medio de puntos y estrellas por ayudarnos en el proceso de clasificación y descubrimiento de aves en la zona metropolitana de Bucaramanga.

Por lo anterior, lo quiero invitar a responder las siguientes preguntas:

- 1. ¿Le gustaría que esta aplicación mostrara información complementaria como imágenes, descripción y gestores de ubicación del ave identificada?*
- 2. ¿Le gustaría interactuar con otras personas dentro de la aplicación que también están identificando aves?*
- 3. ¿Le gustaría ver las imágenes de las aves que otras personas identificaron en el área?*
- 4. ¿Considera atractivo un sistema gamificador basado en puntos por su participación en la identificación de aves?*
- 5. ¿Le gustaría tener su propio portafolio de aves ya identificadas en el área?*

6. ¿Cree necesaria la inclusión de una sección enfocada en mostrar zonas de alto avistamiento de aves en la aplicación?”

Se entrevistaron cuatro usuarios, dos de ellos especializados y dos usuarios normales de la zona metropolitana. Esta entrevista tiene como fin identificar los requerimientos funcionales y no funcionales con el que contaría la aplicación móvil.

Tabla 19 Información de los Usuarios Entrevistados

Información de los usuarios entrevistados.

<i>Nombre completo</i>	<i>Género</i>	<i>Tipo de Usuario</i>	<i>Correo de contacto</i>
<i>Andrea Carolina Camacho Julio</i>	Femenino	General	Acamacho176@unab.edu.co
<i>Jhon Alexander Valencia Hernández</i>	Masculino	General	-
<i>Fernando Cediel</i>	Masculino	Especializado	fdocediel@gmail.com
<i>Daniel Badillo Mojica</i>	Masculino	Especializado	-

Fuente: Autor.

A continuación, encontrará las respuestas dadas por los usuarios descritos en la tabla 1, las respuestas corresponden a las preguntas previamente descritas en el guion.

Andrea Carolina Camacho Julio, estudiante de ingeniería de sistemas de séptimo semestre de la UNAB:

1. Si, primero la imagen y luego de información, cuando esté tomando la foto al ave, pase a otra pantalla con la foto del ave.
2. No le gustaría interactuar con otras personas.
3. Si, le gustaría ver la publicación de otras personas.
4. Si le gustaría que la app coloque retos, pero solo cuando ella lo pida.
5. Ver como un álbum de fotos y que al tocar muestre la información del ave.
6. Si le gustaría ver las zonas en donde se hace avistamiento de aves

Otras recomendaciones: Le gustaría exportar la información a formato pdf y editar sus datos.

Jhon Alexander Valencia Hernández, graduado de ingeniería mecánica UIS y habitante de la zona metropolitana de Bucaramanga:

1. Si, primero la imagen (corroborar el ave) datos básicos y opción de ver más datos.

2. Si, quisiera saber cómo estar los demás de acuerdo con puntos
3. Si, me gustaría comentar y dar “me gusta”.
4. Si, pero podría ser un falso positivo.
5. Agrupadas por familia de especies y luego si las imágenes.
6. Si, porque incitaría a ir a esos lugares.

Fernando Cediel, Biólogo graduado de la UIS:

1. Si, imagen y la descripción. en comparación con la imagen que se tomó.
2. No, para eso existen otros grupos.
3. Si, la ubicación es importante
4. Si, interesante.
5. Si, listado, cuantas ha visto, cuales son raras o fáciles. Check list de las aves que ya están. Distribución de la especie.
6. Si, ver hospost. Ebird.

Otras recomendaciones: Acceder por el número de puntos a nuevas funcionalidades, como la de validar el ave.

Daniel Badillo Mojica, estudiante de biología de la UIS:

1. Si, primero las imágenes breve descripción, canto, tipo de hábitat, tipos de ambientes, no tanto físico. Mapas de distribución (no muy importante)
2. Si, chat.
3. Si.
4. Si.
5. Si, listado de aves que has identificado y las que les falta por ver (check list).
6. Si, sitios en el mapa o mapa con los puntos más calientes.

En conclusión, de las entrevistas realizadas se tiene que, los usuarios quieren ver la información complementaria del ave como: imágenes, familia, especie, descripción y ubicación. Además, les gustaría ver su propio portafolio de aves divididas tal vez entre las que ya ha identificado y las que le falta por identificar. Además, esperan que la aplicación coloque misiones de manera gradual y ver cómo influye en ellos este módulo.

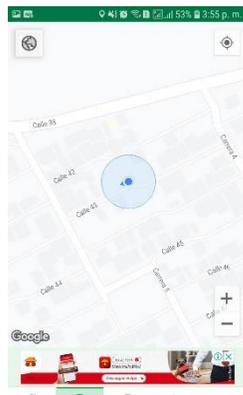
2. Análisis de aplicaciones:

Para el desarrollo de esta actividad se descargaron tres aplicaciones, en las cuales se pudiera identificar un objeto, persona o animal por medio de una imagen. El fin de esta actividad fue analizar el diseño de las interfaces y las funcionalidades que tenía cada una de estas, con esta información se complementarían las funcionalidades obtenidas de los usuarios y se podría sacar la versión de las interfaces.

PlantSnap: identifique plantas, flores y más.

¡Identifica al instante plantas de todo tipo en cualquier parte del mundo! Podrás reconocer rápidamente flores, árboles, suculentas, hongos y mucho más con PlantSnap, de Earth.com, la aplicación móvil que te ayudará a identificar flores, plantas y árboles en un abrir y cerrar de ojos. La enorme base de datos de PlantSnap cubre todas las partes del mundo para que puedas identificar plantas al instante, ¡estés donde estés! (PlantSnap, Inc., 2020)

Ilustración 78 Interfaz "explorar" PlantSnap



Fuente: Autor en base (PlantSnap, Inc., 2020).

Ilustración 79 Interfaz "Usuario" PlantSnap



Fuente: Autor en base (PlantSnap, Inc., 2020)

Ilustración 80 Interfaz "foto" PlantSnap



Fuente: Autor en base (PlantSnap, Inc., 2020).

Ilustración 81 Interfaz "Información de la Planta" PlantSnap



Fuente: Autor en base (PlantSnap, Inc., 2020).

FUNCIONALIDADES:

- Al momento de abrir la aplicación da una pequeña explicación sobre cómo se deben manejar las interfaces.
- La interfaz principal es “Foto” en donde se puede tomar y graduar la foto de la planta que se quiere identificar.
- En la opción “explorar” aparece el mapa con diferentes plantas identificadas cerca del lugar de la ubicación.
- En el “Feed” se encuentra una pequeña red social de fotografías de diferentes plantas identificadas por usuarios, se puede dar “me gusta” y hacer comentarios. Además, puedo revisar mis notificaciones y hacer mis propias publicaciones.
- Debo loguearme para acceder a todas las funcionalidades: correo o Facebook.
- En la interfaz de perfil puedo cambiar: nombre de visualización, dirección de correo electrónico y contraseña. También está la opción de “¿has olvidado la contraseña?” y la de cerrar sesión.
- En la opción de buscar, se puede filtrar por plantas, jardines y gente. En el filtro de gente: se busca por nombre de visualización y aparece la opción de añadir amigo o bloquear el usuario. Si busco en jardines, aparece un pequeño resumen, la categoría y los contactos del lugar. En planta, me aparecen las diferentes opciones de plantas y la información detallada de cada una.
- Información detallada de la planta: donde la puedo conseguir, descripción, reino, subreino, superdivisión, división, clase, subclase, orden, familia, género, hábitat, título, símbolo e imágenes.
- La aplicación demora entre 1 y 1:30 minutos en mostrar resultados de la clasificación de la planta.
- Tiene una demora de 3 y 2 minutos para tomar la foto y graduarla para que sea más fácil la identificación.

Merlin

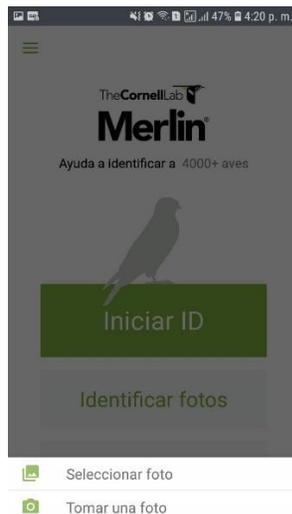
¿Qué es ese pájaro? Merlin ayuda a resolver el misterio en 5 preguntas, o con una foto de un pájaro.

Inicialmente, Merlin te hace unas cuantas preguntas sencillas. Luego, casi como por arte de magia, revela la lista de aves que mejor se adaptan a tu descripción. Elige tu especie y tendrás acceso a más fotos, sonidos y consejos de identificación sobre tu ave (Cornell lab Of Ornithology, 2020)

Funcionalidades:

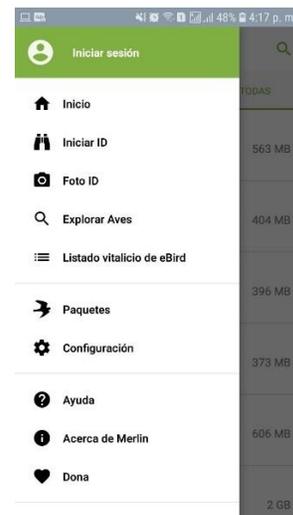
- En la interfaz principal está la opción de identificar fotos, la cual permite buscar o tomar una foto del ave que se quiere identificar, luego se gradúa y la aplicación la clasifica.
- Al momento de clasificar la imagen, aparecen las posibles opciones que más se asemejen al ave buscada, aparece su información detallada: pequeño resumen, sonidos, imágenes y mapa de su ubicación
- En la opción de explorar aves aparecen las posibles aves que se tienen en la zona de ubicación geográfica.
- La identificación toma alrededor de 30 a 60 segundos.
- Para que aparezcan las imágenes de las aves previamente hay que descargar un paquete de imágenes de la zona en la que se quiere identificar aves.
- Si quiero identificar sin tener una imagen un ave se utiliza el “iniciar ID” en el cual me hacen un conjunto de preguntas:
 1. ¿Dónde observaste el ave?
 2. ¿Cuándo observaste el ave?
 3. ¿De qué tamaño era el ave?
 4. ¿Cuáles eran sus principales colores?
 5. Que estaba haciendo el ave
- Debo tener una cuenta de Cornell Lab Ebird para iniciar sesión en la aplicación

Ilustración 82 Interfaz de Tomar o Seleccionar imagen del ave Merlin



Fuente: Autor.

Ilustración 83 Menú Lateral Merlin



Fuente: Autor.

Ilustración 84 Interfaz de Centrar el Ave en la Imagen Merlin



Fuente: Autor.

Ilustración 85 Información del Ave Merlin



Fuente: Autor.

Ilustración 86 Sonidos del ave Merlin



Fuente: Autor.

Ilustración 87 Paquete de Aves a Instalar de la Aplicación



Fuente: Autor.

Dog Scanner

¿Ha visto un perro, pero no sabe su raza? (Siwalu Software GmbH, 2020)

¡Hágale una foto y descúbralo! Esta aplicación buscará la raza automáticamente para usted.

¿Es una raza mixta?

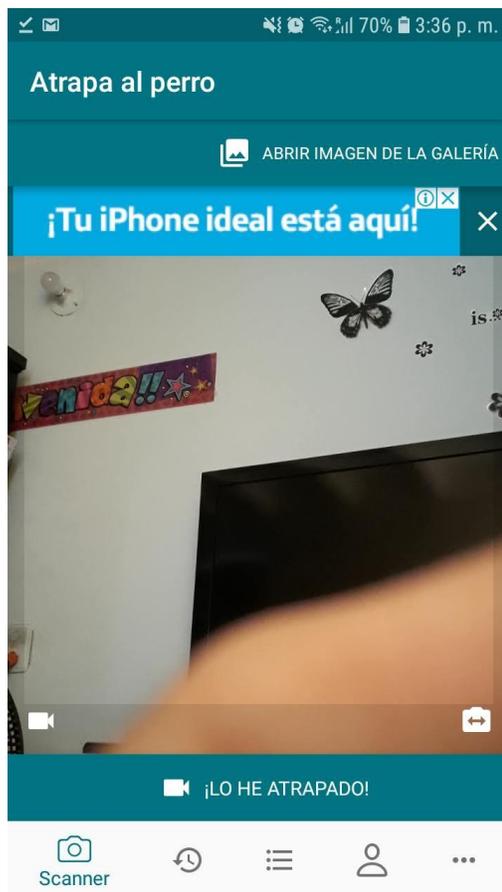
No hay problema, ¡le diremos que razas se han mezclado!

¿No hay perros cerca?

¡Solo escanee a un amigo suyo y descubra qué perro es!

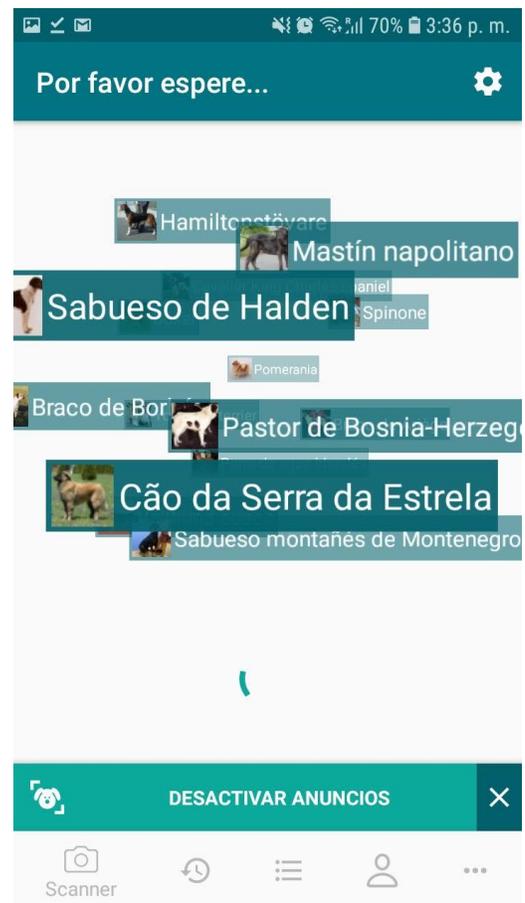
Es una aplicación para identificar razas de perros por medio de una imagen.

Ilustración 88 Interfaz "Atrapar al perro" DogScanner



Fuente: Autor.

Ilustración 89 Calculando la Raza del Perro DogScanner



Fuente: Autor.

Ilustración 90 Información de la Raza del Perro DogScanner



Fuente: Autor.

Ilustración 91 Predicción de la Raza del Perro DogScanner



Fuente: Autor

Funcionalidades:

- En la interfaz principal, permite buscar o tomar una foto de un perro para identificar su raza por medio de la aplicación, no permite graduarse.
- La aplicación toma entre 1 y 1:30 minutos en mostrar las posibles opciones de clasificación del perro.
- Tiene una interfaz para el perfil, inicialmente es uno provisional creado automático por la aplicación que muestra las razas que han sido identificadas por ti en la aplicación.
- Puedes ver el historial de las clasificaciones que hiciste dentro de la aplicación.
- La aplicación tiene una interfaz con las razas disponibles para hacer la clasificación.
- Tiene una interfaz de información del animal: pequeño resumen, historia y origen de la raza, cuidados, raza estándar, también conocido como, imagen y una opción para buscar más en Wikipedia.

**PROTOTIPO DE UN SISTEMA DE VISIÓN ARTIFICIAL BASADO EN
TÉCNICAS DE APRENDIZAJE DE MÁQUINA PARA EL RECONOCIMIENTO DE
AVES DEL ÁREA METROPOLITANA DE BUCARAMANGA**

ANEXO 3

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA-UNAB
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA
2021**

DESARROLLO

A partir de las aplicaciones revisadas y proyectos afines se identificaron 2 datasets de aves que están disponibles para ser utilizados en el entrenamiento del modelo clasificador.

1. Caltech-UCSD Birds.
2. NABirds.

Por lo cual, se realizó una búsqueda de características como: número de categorías, imágenes y país de creación, con el fin de identificar pautas para la creación de un nuevo dataset con solo especies de aves del área metropolitana de Bucaramanga.

Fichas técnicas:

Nombre	Caltech-UCSD Birds-200-2011 (Wah, Branson, Welinder, Perona, & Belongie, 2011)
<i>Descripción</i>	Es una versión extendida del conjunto de imágenes CUB-200 de aves, están organizadas por clasificación científica (orden, familia, género, especie).
<i>Número de categorías</i>	200
<i>Número de imágenes</i>	11,788
<i>Disponibilidad</i>	Disponible para descargar y utilizar en el entrenamiento.

Nombre	NABirds Dataset (Cornell Lab of Ornithology, 2014)
<i>Descripción</i>	Es conjunto de datos para reconocimiento detallado de especies de aves de Norte América, está organizado de forma taxonómica y por machos, hembra y juveniles. Está creado por la colaboración entre el Laboratorio de Ornitología de Cornell, Caltech, la Universidad Brigham Young y diferentes de fotógrafos colaboradores que visitan el sitio web All About Birds.
<i>Número de categorías</i>	400
<i>Número de imágenes</i>	48,000
<i>Disponibilidad</i>	Disponible para descargar y utilizar en el entrenamiento.

Finalmente, se decidió crear un dataset sobre 9 especies de aves del área metropolitana de Bucaramanga con el fin de que el entrenamiento fuera más transparente sobre las especies que podría clasificar.

**PROTOTIPO DE UN SISTEMA DE VISIÓN ARTIFICIAL BASADO EN
TÉCNICAS DE APRENDIZAJE DE MÁQUINA PARA EL RECONOCIMIENTO DE
AVES DEL ÁREA METROPOLITANA DE BUCARAMANGA**

ANEXO 4

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA-UNAB
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA
2021

PLAN DE PRUEBAS

Población: habitantes de la zona metropolitana de Bucaramanga.

Muestra: 5 habitantes usuarios normales y 1 habitantes usuarios especializados (biólogos o afines).

Inicialmente, se realizó un test para medir el impacto antes y después de utilizar la aplicación, las preguntas están enumeradas seguidamente.

Tabla 20 Preguntas test Antes y Después de Usar la App

Pregunta	Opciones de respuesta	Respuesta correcta
 ¿Cuál es el nombre de esta ave?	<ul style="list-style-type: none"> • Carpintero habado (Melanerpes rubricapillus) • Bichofué gritón (Pitangus sulphuratus) • Gavilán pollero (Buteo magnirostris) • Tortolita (Columbina talpacoti) 	Gavilán pollero (Buteo magnirostris)
 ¿Cuál es el nombre de esta ave?	<ul style="list-style-type: none"> • Semillero pechinegro (Tiaris bicolor) • Sirirí común (Tyrannus melancholicus) • Carpintero habado (Melanerpes rubricapillus) • Periquito bronceado (Brotogeris jugularis) 	Periquito bronceado (Brotogeris jugularis)
 ¿Cuál es el nombre de esta ave?	<ul style="list-style-type: none"> • Periquito bronceado (Brotogeris jugularis) • Sirirí común (Tyrannus melancholicus) • Semillero pechinegro (Tiaris bicolor) • Colibrí (Amazilia tzacatl) 	Semillero pechinegro (Tiaris bicolor)
 ¿Cuál es el nombre de esta ave?	<ul style="list-style-type: none"> • Tortolita (Columbina talpacoti) • Sirirí común (Tyrannus melancholicus) • Bichofué Gritón (Pitangus sulphuratus) • Semillero pechinegro (Tiaris bicolor) 	Tortolita (Columbina talpacoti)

 <p>¿Cuál es el nombre de esta ave?</p>	<ul style="list-style-type: none"> • Tortolita (<i>Columbina talpacoti</i>) • Periquito bronceado (<i>Brotogeris jugularis</i>) • Colibrí (<i>Amazilia tzacatl</i>) • Buitre negro (<i>Coragys atratus</i>) 	<p>Colibrí (<i>Amazilia tzacatl</i>)</p>
 <p>¿Cuál es el nombre de esta ave?</p>	<ul style="list-style-type: none"> • Sirirí común (<i>Tyrannus melancholicus</i>) • Bichofué Gritón (<i>Pitangus sulphuratus</i>) • Colibrí (<i>Amazilia tzacatl</i>) • Carpinero habado (<i>Melanerpes rubricapillus</i>) 	<p>Bichofué Gritón (<i>Pitangus sulphuratus</i>)</p>
 <p>¿Cuál es el nombre de esta ave?</p>	<ul style="list-style-type: none"> • Periquito bronceado (<i>Brotogeris jugularis</i>) • Gavilán pollero (<i>Buteo magnirostris</i>) • Buitre negro (<i>Coragys atratus</i>) • Tortolita (<i>Columbina talpacoti</i>) 	<p>Buitre negro (<i>Coragys atratus</i>)</p>
 <p>¿Cuál es el nombre de esta ave?</p>	<ul style="list-style-type: none"> • Colibrí (<i>Amazilia tzacatl</i>) • Sirirí común (<i>Tyrannus melancholicus</i>) • Buitre negro (<i>Coragys atratus</i>) • Periquito bronceado (<i>Brotogeris jugularis</i>) 	<p>Sirirí común (<i>Tyrannus melancholicus</i>)</p>

 <p>¿Cuál es el nombre de esta ave?</p>	<ul style="list-style-type: none"> • Carpinero habado (Melanerpes rubricapillus) • Sirirí común (Tyrannus melancholicus) • Gavilán pollero (Buteo magnirostris) • Bichofué Gritón (Pitangus sulphuratus) 	<p>Carpinero habado (Melanerpes rubricapillus)</p>
--	--	--

Fuente: Autor.

A continuación, se enumerarán las preguntas que den veracidad al cumplimiento de las funcionalidades de la aplicación móvil.

FUNCIONALIDAD:

Tabla 21 Plan de Pruebas de Funcionalidad

Pregunta	Caso de uso asociado	Check
¿La aplicación le permitió ingresar? (Loguearse)	-	Cumplió o no cumplió.
¿Se pudo registrar en la aplicación?	ECU001	Cumplió o no cumplió.
¿Pudo visualizar su información en la interfaz "perfil" de la aplicación?	ECU001	Cumplió o no cumplió.
¿Pudo editar sus datos?	ECU001	Cumplió o no cumplió.
¿Pudo visualizar la lista de aves que se pueden identificar en AveDex?	ECU002	Cumplió o no cumplió.
Al darle clic a cualquier ítem de la lista ¿Le permitió visualizar la información del ave?	ECU002	Cumplió o no cumplió.
¿Pudo identificar un ave tomando una foto desde su celular?	ECU004 ECU007	Cumplió o no cumplió.
¿Pudo identificar un ave eligiendo una foto desde su galería?	ECU004 ECU007	Cumplió o no cumplió.
En caso de estar identificando un ave desde	ECU004 ECU003	Cumplió o no cumplió.

una imagen de la galería, ¿Pudo visualizar el mapa y escoger la ubicación de donde vio el ave?		
¿Pudo ver el mapa de aves identificadas por usted?	ECU005	Cumplió o no cumplió.
¿Pudo visualizar la información del ave al momento de identificar un ave desde las dos opciones (galería o cámara)?	ECU002 ECU003 ECU004 ECU007	Cumplió o no cumplió.
¿Pudo recuperar la contraseña de su cuenta desde la aplicación?	ECU006	Cumplió o no cumplió.

Fuente: Autor.

ACEPTACIÓN:

Tabla 22 Plan de Pruebas de Aceptación

Acción	Check
Ingresar al sistema.	Malo, regular o bueno.
Ver su perfil.	Malo, regular o bueno.
Identificar un ave, desde una foto tomada de su teléfono.	Malo, regular o bueno.
Identificar un ave, desde una foto elegida de la galería de su celular.	Malo, regular o bueno.
Visualizar el mapa de aves identificadas.	Malo, regular o bueno.
Editar sus datos.	Malo, regular o bueno.
Recuperar su contraseña.	Malo, regular o bueno.
Visualizar la lista de aves que se pueden identificar en AveDex.	Malo, regular o bueno.
Visualizando la información del ave.	Malo, regular o bueno.
Los colores de la aplicación.	Malo, regular o bueno.
Los mensajes en los procesos de la aplicación.	Malo, regular o bueno.
La facilidad en el manejo de la aplicación.	Malo, regular o bueno.
El diseño en general de la aplicación.	Malo, regular o bueno.

Fuente: Autor.

**PROTOTIPO DE UN SISTEMA DE VISIÓN ARTIFICIAL BASADO EN
TÉCNICAS DE APRENDIZAJE DE MÁQUINA PARA EL RECONOCIMIENTO DE
AVES DEL ÁREA METROPOLITANA DE BUCARAMANGA**

ANEXO 5

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA-UNAB
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA
2021**

MANUAL DE USUARIO

¿Qué es AveDex?

AveDex app es una aplicación para identificar aves en el área metropolitana de Bucaramanga desde un sistema de IA que predice un ave a partir de una foto de esta, la versión beta cuenta con un accuracy del 89%. Por lo tanto, puede pasar que sus predicciones no sean exactas.

Actualmente, contamos con 9 especies registradas. Por ello, te pedimos, que solo ingreses imágenes de ellas.

¿Qué aves identifica AveDex?

A continuación, te mostraremos las aves que están registradas.

	<p>Semillero Pechinegro </p> <p>Tiaris bicolor</p> <p>Familia: Thraupidae</p> <p> AM de Bucaramanga</p>		<p>Sirirí común </p> <p>Tyrannus melancholicus</p> <p>Familia: Tyrannidae</p> <p> AM de Bucaramanga</p>
	<p>Bichofué Gritón </p> <p>Pitangus sulphuratus</p> <p>Familia: Tyrannidae</p> <p> AM de Bucaramanga</p>		<p>Carpintero habado </p> <p>Melanerpes rubricapillus</p> <p>Familia: Picidae</p> <p> AM de Bucaramanga</p>
	<p>Buitre negro </p> <p>Coragyps atratus</p> <p>Familia: Cathartidae</p> <p> AM de Bucaramanga</p>		<p>Tortolita </p> <p>Columbina talpacoti</p> <p>Familia: Columbidae</p> <p> AM de Bucaramanga</p>



Gavián pollero



Buteo magnirostris

Familia: Accipítridos

AM de Bucaramanga



Periquito bronceado



Brotogeris jugularis

Familia: Psittacidae

AM de Bucaramanga



Tortolita



Columbina talpacoti

Familia: Columbidae

AM de Bucaramanga

Pequeña guía.

Si quieres identificar aves.

Ingresas y ve a la pestaña identificar, **recuerda siempre tener el gps encendido.** podrás identificar por medio de:

- Escaneo.
- Seleccionar imagen de la galería.

Si quieres ver las aves que identifica aveDex y su información.

Ingresas y ve a la pestaña lista.

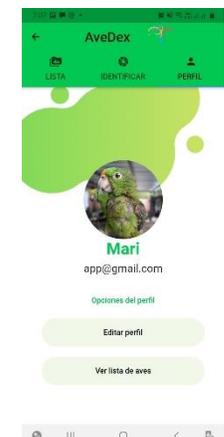
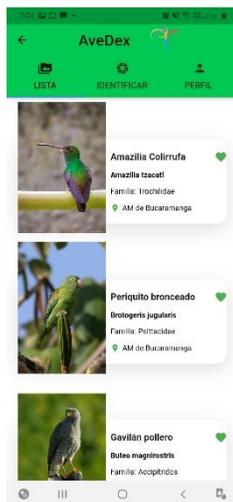
¡Recuerda!

Puedes clicar las tarjetas para ver la información del ave.

Si quieres ver las opciones del perfil.

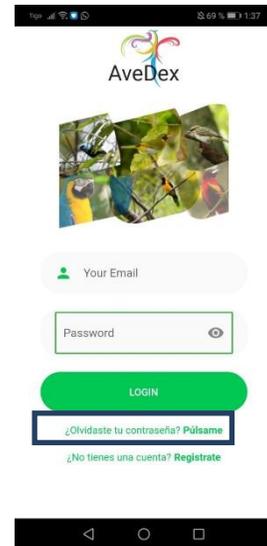
Ingresas y ve a la pestaña perfil.

- Edita tu perfil.
- Ver el mapa con tus identificaciones.



Si quieres recuperar tu contraseña

En la interfaz de ingreso, encontrarás la opción para recuperarla.



Gracias

¡Te agradecemos por ayudarnos a mejorar, esperamos que la experiencia haya sido única!