



UNIVERSIDAD TÉCNICA DE COTOPAXI
FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

CARRERA DE INGENIERIA EN SISTEMAS DE INFORMACIÓN

PROYECTO DE INVESTIGACIÓN

**“DESARROLLO DE UN PROTOTIPO PARA LA
IDENTIFICACIÓN AUTOMÁTICA DE PLAGAS Y
ENFERMEDADES EN EL CULTIVO DE PAPA, UTILIZANDO
TÉCNICAS DE INTELIGENCIA ARTIFICIAL EN LA CIUDAD
DE LATACUNGA”**

Proyecto de Investigación presentado previo a la obtención del título de
Ingeniero en Sistemas de Información

AUTORES:

Angel Rodrigo Gavilanez Guanoluisa
Bryan Alexander Saragosin Guamushig

TUTOR:

Ing. Mtr. Karla Susana Cantuña Flores

LATACUNGA – ECUADOR
MARZO-2024



UNIVERSIDAD TÉCNICA DE COTOPAXI – CARRERA DE SISTEMAS DE INFORMACIÓN

DECLARACIÓN DE AUTORÍA

Gavilanez Guanoluisa Angel Rodrigo, con cédula de ciudadanía No. 0550712962, Saragosin Guamushig Bryan Alexander, con cédula de ciudadanía No. 0550054134, declaramos ser autores del presente **PROYECTO DE INVESTIGACIÓN: “DESARROLLO DE UN PROTOTIPO PARA LA IDENTIFICACIÓN AUTOMÁTICA DE PLAGAS Y ENFERMEDADES EN EL CULTIVO DE PAPA, UTILIZANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL, EN LA CIUDAD DE LATACUNGA”**, siendo la Ing. Mtr. Karla Susana Cantuña Flores, Tutora del presente trabajo; y, eximo expresamente a la Universidad Técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

Además, certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Latacunga, marzo 2024

Angel Rodrigo Gavilanez Guanoluisa
C.C: 0550712962

Bryan Alexander Saragosin Guamushig
C.C: 0550054134



AVAL DEL TUTOR DE PROYECTO DE TITULACIÓN

En calidad de Tutora del Proyecto de Investigación sobre el título:

“DESARROLLO DE UN PROTOTIPO PARA LA IDENTIFICACIÓN AUTOMÁTICA DE PLAGAS Y ENFERMEDADES EN EL CULTIVO DE PAPA, UTILIZANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL, EN LA CIUDAD DE LATACUNGA”, de Gavilanez Guanoluisa Angel Rodrigo, Saragosin Guamushig Bryan Alexander, de la carrera de Ingeniería en Sistemas de Información, considero que dicho Informe Investigativo es merecedor del aval de aprobación al cumplir con las normativas técnicas, traducción y formatos previstos, así como también ha incorporado las observaciones y recomendaciones propuestas en la pre-defensa.

Latacunga, marzo 2024

Ing. Mtr. Karla Susana Cantuña Flores
C.C: 0502305113
TUTORA



UNIVERSIDAD TÉCNICA DE COTOPAXI – CARRERA DE SISTEMAS DE INFORMACIÓN

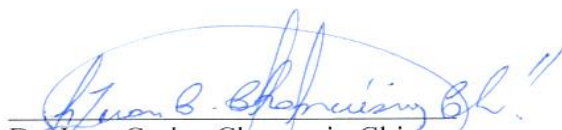
AVAL DE APROBACIÓN DEL TRIBUNAL DE TITULACIÓN


En calidad de Tribunal de Lectores, aprueban el presente Informe de Investigación de acuerdo a las disposiciones reglamentarias emitidas por la Universidad Técnica de Cotopaxi, y por la FACULTAD de Ciencias de la Ingeniería y Aplicadas; por cuanto, los postulantes: Gavilanez Guanoluisa Angel Rodrigo y Saragosin Guamushig Bryan Alexander, con el título del Proyecto de Investigación: **“DESARROLLO DE UN PROTOTIPO PARA LA IDENTIFICACIÓN AUTOMÁTICA DE PLAGAS Y ENFERMEDADES EN EL CULTIVO DE PAPA, UTILIZANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL, EN LA CIUDAD DE LATACUNGA”**, ha considerado las recomendaciones emitidas oportunamente y reúne los méritos suficientes para ser sometido al acto de sustentación del trabajo de titulación.


Por lo antes expuesto, se autoriza grabar los archivos correspondientes en un CD, según la normativa institucional.

Latacunga, marzo 2024

Para constancia firman:


Dr. Juan Carlos Chancusig Chisag
C.C: 0502275779
LECTOR 1 (PRESIDENTE)


Mg. Manuel William Villa Quishpe
C.C: 1803386950
LECTOR 2 (MIEMBRO)


Mg. Edison Patricio Bedón Salazar
C.C: 0502253271
LECTOR 3 (MIEMBRO)

AGRADECIMIENTO

En primer lugar, agradezco a Dios por darme las fuerzas y valentía para seguir adelante.

Agradezco a María Edelmira Guanoluisa, mi fuente inagotable de amor y apoyo. Gracias por ser mi guía, mi inspiración y por enseñarme el valor incalculable del esfuerzo y la perseverancia. Este logro es también suyo.

A mis hermanos, compañeros de vida cuyo apoyo constante y aliento inquebrantable han sido mi fuerza en los momentos desafiantes. Compartir este viaje con ustedes ha hecho cada paso más significativo. Gracias por ser pilares fundamentales en mi camino, por creer en mí y por ser parte esencial de esta etapa inolvidable.

Angel Gavilanez



AGRADECIMIENTO

En primer lugar, agradezco a Dios por darme las fuerzas y valentía para seguir adelante.

Agradezco eternamente a mis padres por su apoyo incondicional y por estar a mi lado en cada momento, por enseñarme a perseverar ante cualquier adversidad y brindarme valiosos consejos, inculcándome la creencia de que con dedicación se pueden alcanzar todas las metas propuestas.

A los docentes de la Universidad Técnica de Cotopaxi por brindarme sus conocimientos y experiencia durante la formación académica.

Bryan Saragosin



DEDICATORIA

Quiero expresar mi profunda gratitud a María Edelmira Guanoluisa, mi madre, cuyo amor y sacrificio han sido la fuente inagotable de mi motivación. A Diego Gavilanez, mi hermano, por su apoyo constante y aliento inquebrantable, que han sido mi roca en los momentos más desafiantes.

A mis hermanos, cuya presencia ha iluminado mi camino y ha enriquecido cada logro. Su respaldo incondicional ha sido una bendición que agradezco sinceramente.

Angel Gavilanes



DEDICATORIA

A mis padres por ser el pilar fundamental para lograr este objetivo, por apoyarme en cada momento y brindarme los consejos para animarme en los momentos difíciles, a mis hermanas por estar al pendiente y mostrarme el apoyo incondicional en cada momento.

Bryan Saragosin



UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

TITULO: “DESARROLLO DE UN PROTOTIPO PARA LA IDENTIFICACIÓN AUTOMÁTICA DE PLAGAS Y ENFERMEDADES EN EL CULTIVO DE PAPA, UTILIZANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL, EN LA CIUDAD DE LATACUNGA”

Autores:

Gavilanez Guanoluisa Angel Rodrigo
Saragosin Guamushig Bryan Alexander

RESUMEN

El presente proyecto de investigación aborda la identificación de plagas y enfermedades en el cultivo de papa en la ciudad de Latacunga, mediante técnicas de Inteligencia Artificial, por medio del juicio de expertos y de la revisión de la literatura científica, considerando para este proyecto la visión artificial mediante el entrenamiento con redes neuronales convolucionales (CNN), cuyos resultados se presentan en una interfaz móvil, la cual fue diseñada para los agricultores del cultivo de papa. Para el desarrollo y entrenamiento del modelo se empleó la metodología KDD, y se usaron herramientas de aprendizaje automático. Además, se empleó Google Colab como plataforma de desarrollo, Python como lenguaje de programación, Keras para la creación del modelo con su motor de ejecución, TensorFlow y Android Studio para crear la interfaz móvil. En el proceso de entrenamiento y prueba del modelo se utilizó 2501 imágenes de plagas y enfermedades basadas en fotografías las cuales fueron divididas en 4 clases como: hojas sanas, hojas con la plaga pulguilla de la papa, hojas con tizón temprano y tardío. Con el objetivo de seleccionar el mejor modelo propuesto, se compararon cinco arquitecturas basadas en transfer learning como: ResNet50, VGG19, InceptionV3, EfficientNetB0 y MobileNet. Además, se desarrolló un modelo propio con una precisión del 95,34 %, el mismo que fue integrado en el prototipo de software para la identificación de plagas y enfermedades.

Palabras claves: Inteligencia artificial, redes neuronales convolucionales, transfer learning Google Colab, Keras.



TECHNICAL UNIVERSITY OF COTOPAXI

FALCULTY OF ENGINEERING SCIENCES AND APPLIED

THEME: “DEVELOPMENT OF A PROTOTYPE FOR THE AUTOMATIC IDENTIFICATION OF PESTS AND DISEASES IN POTATO CROPS, USING ARTIFICIAL INTELLIGENCE TECHNIQUES, IN LATACUNGA CITY”

Authors:

Gavilanez Guanoluisa Angel Rodrigo
Saragosin Guamushig Bryan Alexander

ABSTRACT

The present research project addresses the identification of pests and diseases in potato cultivation in Latacunga city, through Artificial Intelligence techniques, as well as the judgment of experts and the review of scientific literature, considering for this project the artificial vision through training with convolutional neural networks (CNN), whose results are presented in a mobile interface, which is designed for potato farmers. For the development and training of the model, the Discovery Knowledge in Databases (KDD) methodology was applied, and machine learning tools were used. In addition, Google Colab was employed as a development platform, Python as a programming language, Keras for the creation of the model with its execution engine, TensorFlow, and Android Studio to create the mobile interface. In the training and testing process of the model, 2,501 images of pests and diseases based on photographs were used, which were divided into 4 classes, such as: healthy leaves, leaves with the potato flea pest, and leaves with early and late blight. With the objective of selecting the best proposed model, five architectures based on transfer learning were compared, such as ResNet50, VGG19, InceptionV3, EfficientNetB0, and MobileNet. Moreover, an own model was developed with an accuracy of 95.34%, which was integrated into the software prototype for the identification of pests and diseases.

Keywords: Artificial intelligence, convolutional neural networks, transfer learning Google Colab, Keras.



INDICE GENERAL

1.	INFORMACIÓN GENERAL	21
2.	INTRODUCCIÓN	23
2.1.	PROBLEMA	24
2.1.1.	Situación Problemática	24
2.1.2.	Formulación del problema	25
2.2.	OBJETO Y CAMPO DE ACCIÓN	25
2.3.	BENEFICIARIOS	25
2.3.1.	Beneficiarios Directos	25
2.3.2.	Beneficiarios Indirectos	26
2.4.	JUSTIFICACIÓN	26
2.5.	HIPÓTESIS	27
2.6.	OBJETIVOS	27
2.6.1.	Objetivo General.....	27
2.6.2.	Objetivos Específicos	27
2.7.	SISTEMA DE TAREAS	28
3.	FUNDAMENTACIÓN TEÓRICA	30
3.1.	Inteligencia artificial	31
3.2.	Computer visión.....	32
3.2.1.	Procesamiento de Imágenes.....	32
3.3.	Machine Learning	33
3.3.1.	Aprendizaje Supervisado	33
3.3.2.	Aprendizaje No Supervisado	33
3.4.	Deep Learning.....	34
3.5.	Redes neuronales convolucionales (CNN)	34
3.5.1.	Hiperparámetros.	34



3.5.2.	Transfer learning.....	36
3.5.3.	Modelo ResNet50	36
3.5.4.	Modelo InceptionV3	37
3.5.5.	Modelo VGG19	37
3.5.6.	Modelo EfficientNet	37
3.5.7.	Modelo MobileNet	37
3.5.8.	Aplicaciones de las Redes Neuronales Convolucionales (CNN)	37
3.6.	Construcción y validación del modelo.....	40
3.6.1.	Google Colab	41
3.6.2.	Entrenamiento y validación del modelo	41
3.6.3.	Epochs	41
3.6.4.	Keras	41
3.6.5.	Tensorflow	42
3.6.6.	Lenguaje de programación Python	42
3.6.7.	Biblioteca TensorFlow Lite	42
3.7.	Herramientas para el desarrollo del Prototipo	42
3.7.1.	Aplicación móvil	42
3.8.	LA PAPA	43
3.8.1.	Variedades de la papa	43
3.9.	PLAGAS EN LA HOJA DE PAPA.....	45
3.9.1.	Gusano blanco de la papa	45
3.9.2.	Polilla de la papa	46
3.9.3.	Pulguilla de la papa	46
3.9.4.	TRIPS	47
3.9.5.	Mosca minadora	48
3.9.6.	Gusano Trozador	48
3.9.7.	Mosca blanca	49



3.9.8.	Pulgones	50
3.10.	ENFERMEDADES EN LA HOJA DE PAPA.....	51
3.10.1.	Roña	51
3.10.2.	Tizón tardío	51
3.10.3.	OÍDIO	52
3.10.4.	Punta morada	53
3.10.5.	Tizón temprano	53
3.10.6.	Roya	54
3.10.7.	Costra negra	54
3.10.8.	Pie negro y pudrición blanda	55
4.	METODOLOGÍA.....	56
4.1.	Investigación Tecnológica.....	56
4.2.	Investigación De Campo.....	57
4.3.	Investigación Aplicada.....	57
4.4.	Métodos de investigación	57
4.4.1.	Método hipotético deductivo	57
4.4.2.	Método inductivo.....	57
4.4.3.	Método deductivo	58
4.5.	Técnica del cálculo de estimación del costo del software	58
4.5.1.	COCOMO.....	58
4.6.	Herramienta para realizar pruebas del aplicativo	58
4.6.1.	Sauce Labs	58
4.7.	Instrumentos de Investigación	58
4.7.1.	Entrevista	58
4.7.2.	Ficha bibliográfica.....	59
4.8.	Metodología usada para los modelos de inteligencia artificial.....	59
4.8.1.	Metodología KDD	59



4.9.	Metodología de desarrollo de software.....	60
4.9.1.	Mobile-D	60
4.10.	Características de hardware y software	61
4.10.1.	Características de las computadoras	61
5.	ANÁLISIS DE RESULTADOS	62
5.1.	Metodología KDD	62
5.1.1.	Etapa de Selección.....	62
5.1.2.	Etapa de preprocesamiento de datos.....	63
5.1.3.	Etapa de Transformación de datos.....	65
5.1.4.	Etapa de Minería de datos	66
5.1.5.	Etapa de Evaluación de patrones	72
5.1.6.	Comparativa De Modelos.....	83
5.2.	Metodología Mobile-D	85
5.2.1.	Fase 1: Exploración	85
5.2.2.	Fase 2: Inicialización	86
5.2.3.	Fase 3: Producción	92
5.2.4.	Fase 4: Estabilización	94
5.2.5.	Fase 5: Pruebas	95
5.3.	Costo del software	96
5.3.1.	Estimación de la cantidad de instrucciones	96
5.3.2.	Estimación del esfuerzo.....	97
5.3.3.	Estimación del tiempo de desarrollo	97
5.3.4.	Estimación del personal necesario.....	97
5.3.5.	Estimación de productividad	97
5.3.6.	Estimación del coste	97
5.4.	Comprobación de la hipótesis.....	98
5.4.1.	Juicio de experto.....	98



6.	CONCLUSIONES Y RECOMENDACIONES	98
6.1.	Conclusiones.....	98
6.2.	Recomendaciones	99
7.	REFERENCIAS	99
8.	ANEXOS.....	105
8.1.	Anexos: Diagramas de caso de uso a detalle	105
8.2.	Anexos: Diagramas de secuencia.....	107
8.3.	Anexos: Casos de Pruebas	108
8.4.	Anexo: Formulación de la Entrevista	111
8.5.	Anexos: Características de hardware y software de los equipos utilizados para el entrenamiento de los modelos	113
8.6.	Anexos: Juicio del experto.....	114
8.7.	Anexo: Ficha Bibliográfica.....	115
8.8.	Anexo: Pruebas APK UTC Papa IA con la plataforma Sauce Labs	116
8.9.	Anexo: Aplicación de entrevista al experto humano	116
8.10.	Anexo: Hoja de vida investigador 1	117
8.11.	Anexo: Hoja de vida investigador 2.....	118



INDICE TABLAS

Tabla 1.	Sistema de Tareas	28
Tabla 2.	Tratamiento plaga polilla de la papa	46
Tabla 3.	Tratamiento pulgilla de la papa	47
Tabla 4.	Tratamiento plaga gusano trozador	49
Tabla 5.	Tratamiento plaga Mosca blanca	49
Tabla 6.	Tratamiento plaga Pulgones	50
Tabla 7.	Tratamiento enfermedad la Roña	51
Tabla 8.	Tratamiento enfermedad Tizón Tardío	52
Tabla 9.	Tratamiento enfermedad OIDIO	53
Tabla 10.	Tratamiento enfermedad punta morada	54
Tabla 11.	Tratamiento enfermedad costra negra	55
Tabla 12.	Tratamiento enfermedad pie negro y pudrición blanda	56
Tabla 13.	Herramientas utilizadas para la aplicación móvil.....	61
Tabla 14.	Herramientas utilizadas para los modelos	61
Tabla 15.	Especificaciones de la computadora N°1	113
Tabla 16.	Especificaciones de la computadora N°2	113
Tabla 17.	Conjuntos de datos para el entrenamiento de los modelos.....	63
Tabla 18.	Equipo de desarrollo.....	85
Tabla 20.	Requisitos Funcionales	86
Tabla 21.	Requisitos No Funcionales	88
Tabla 22.	CU001 Capturar Fotografía.....	90
Tabla 23.	Interfaces de la aplicación móvil.....	92
Tabla 24.	Caso de prueba capturar imagen.....	95
Tabla 25.	Historia de usuario para buscar la imagen en la galería	
Tabla 28.	CU02 Ingresar a la galería.....	105
Tabla 29.	CU03 Visualizar resultados	105



Tabla 30. Caso de prueba Seleccionar imagen 108
Tabla 31. Caso de prueba visualizar los resultados 109

INDICE FIGURAS

Figura 1. Diagrama de flujo del modelo 38
Figura 2. Esctrucura del modelo Complete Concatenated Deep Learning (CCDL) 38
Figura 3. Resumen esquemático del proyecto 39
Figura 4. Arquitectura de la red neuronal convolucional 39
Figura 5. Estructura de la Arquitectura propuesta 40
Figura 6. Esctrucura del modelo pre-entrenado AlexNet 40
Figura 7. Tubérculos de chola 44
Figura 8. Tubérculos leona negra 44
Figura 9. Tubérculos uvilla 44
Figura 10. Tubérculos de bolona 45
Figura 11. Tubérculos de chaucha colorada 45
Figura 12. Plaga gusano blanco 46
Figura 13. Plaga pulguilla de la papa 47
Figura 14. Plaga TRIPS 48
Figura 15. Plaga mosca minadora 48
Figura 16. Plaga mosca blanca 49
Figura 17. Plaga pulgones 50
Figura 18. Enfermedad tizón tardío 51
Figura 19. Enfermedad OIDIO 52
Figura 20. Enfermedad punta morada 53
Figura 21. Enfermedad tizón temprano 54
Figura 22. Enfermedad costra negra 55
Figura 23. Enfermedad pie negro y pudrición blanda 56



Figura 24.	Versión de Android utilizado	61
Figura 25.	Código para redimensionar las imágenes a 224 x 224	64
Figura 26.	Código para particionar el dataset	65
Figura 27.	Importación de la biblioteca del modelo ResNet50.....	66
Figura 28.	Modelo preentrenado ResNet50.	66
Figura 29.	Construcción del modelo.	66
Figura 30.	Entrenar el modelo ResNet50.....	67
Figura 31.	Importación de la biblioteca VGG19.....	67
Figura 32.	Modelo preentrenado VGG19	67
Figura 33.	Construcción del modelo VGG19	68
Figura 34.	Entrenamiento del modelo VGG19	68
Figura 35.	Importación de la biblioteca del modelo InceptionV3	68
Figura 36.	Modelo preentrenado InceptionV3	68
Figura 37.	Construcción del modelo InceptionV3	69
Figura 38.	Entrenamiento del modelo InceptionV3.....	69
Figura 39.	Importación de la biblioteca del modelo EfficientNetB0	69
Figura 40.	Modelo preentrenado EfficientNetB0	69
Figura 41.	Construcción del modelo EfficientNetB0	70
Figura 42.	Entrenamiento del modelo EfficientNetB0	70
Figura 43.	Importación de la biblioteca del modelo MobileNet	70
Figura 44.	Modelo preentrenado MobileNet.....	70
Figura 45.	Construcción del modelo MobileNet.....	71
Figura 46.	Entrenamiento del modelo MobileNet	71
Figura 47.	Importación de las bibliotecas del modelo propuesto	71
Figura 48.	Construcción del modelo propuesto.....	72
Figura 49.	Entrenamiento del modelo propuesto	72
Figura 50.	Gráfica de entrenamiento y validación de precisión modelo ResNet50.....	73



Figura 51.	Gráfica de entrenamiento y validación de pérdida, modelo ResNet50	73
Figura 52.	Matriz de confusión modelo ResNet50	74
Figura 53.	Gráfica de entrenamiento y validación de precisión, modelo VGG19.....	74
Figura 54.	Gráfica de entrenamiento y validación de pérdida, modelo VGG19	75
Figura 55.	Matriz de confusión modelo VGG19	75
Figura 56.	Gráfica en entrenamiento y validación de precisión, modelo InceptionV3	76
Figura 57.	Gráfica de entrenamiento y validación de pérdida, modelo InceptionV3	76
Figura 58.	Matriz de confusión modelo InceptionV3	77
Figura 59.	Gráfica de entrenamiento, validación de precisión del modelo EfficientNetB0 ..	77
Figura 60.	Gráfica de entrenamiento y validación de pérdida, modelo EfficienNetB0.....	78
Figura 61.	Gráfica de entrenamiento y validación de precisión del modelo MobileNet	78
Figura 62.	Matriz de confusión modelo InceptionV3	79
Figura 63.	Gráfica de entrenamiento y validación de pérdida, modelo EfficienNetB0.....	79
Figura 64.	Matriz de confusión modelo MobileNet.....	80
Figura 65.	Gráfica de entrenamiento de precisión y pérdida modelo propuesto	80
Figura 66.	Matriz de confusión modelo propuesto	81
Figura 67.	Cuadro estadístico de la tabulación	83
Figura 68.	Diagrama de caso de uso general	89
Figura 69.	Diagrama de la aplicación	91
Figura 70.	Diagrama de clases del aplicativo	91
Figura 71.	Diagrama de base de datos	91
Figura 72.	Diagrama de secuencia de captura de imagen	92
Figura 73.	Código fuente visualizar resultados, clasificar imagen	94
	Codigo presentacion de resultados	94
Figura 74.	Código fuente tomar fotografía, seleccionar imagen.....	95
Figura 75.	Caso de prueba seleccionar imagen.....	96
Figura 76.	Diagrama de secuencia ingresar a la galería.....	107



Figura 77.	Diagrama de secuencia visualizar resultados	108
Figura 78.	Caso de prueba seleccionar imagen.....	108
Figura 79.	Caso de prueba visualizar resultados.....	109
Figura 80.	Caso de prueba visualizar resultados.....	111
Figura 81.	Juicio del experto.....	114
Figura 82.	Juicio del experto.....	115
Figura 83.	Ficha Bibliográfica	115
Figura 84.	Pruebas APK UTC Papa IA con la plataforma Sauce Labs.....	116
Figura 85.	Entrevista experto humano	116
Figura 86.	Hoja de vida investigador 1	117
Figura 87.	Hoja de vida investigador 2	118

1. INFORMACIÓN GENERAL

Título: Desarrollo de un prototipo para la identificación automática de plagas y enfermedades en el cultivo de papa, utilizando técnicas de Inteligencia Artificial en la ciudad de Latacunga.

Tipo de Proyecto: Proyecto de Investigación

Carrera: Ingeniería en Sistemas de Información

Proyecto de investigación vinculado: Identificación de las principales plagas y enfermedades del cultivo de papa andina y chocho en la sierra ecuatoriana aplicando técnicas de inteligencia artificial.

Equipo de Trabajo:

Tutor de titulación:

Nombre: Ing. Mtr. Karla Susana Cantuña Flores

Cédula de Identidad: 0502305113

Correo electrónico: karla.cantuna@utc.edu.ec

Dirección: Av. Simón Rodríguez y Calle Paraguay

Estudiantes:

Nombre: Angel Rodrigo Gavilanez Guanoluisa

Cédula de Identidad: 0550712962

Correo electrónico: angel.gavilanez2962@utc.edu.ec

Dirección: San Felipe, Calle Paraguay y 10 de agosto, Cantón Latacunga, Provincia de Cotopaxi

Nombre: Bryan Alexander Saragosin Guamushig

Cédula de ciudadanía: 0550054134

Correo electrónico: bryan.saragosin4134@utc.edu.ec

Dirección: Barrio Alsacia, Calle San Isidro Labrador y Juan Abel Echeverría, Cantón Latacunga, Provincia de Cotopaxi.

Área de Conocimiento: 06 Información y Comunicación (TIC) / 061 Información y Comunicación (TIC) / 0613 Software y desarrollo y análisis de aplicativos.



Línea de investigación: Tecnologías de la información y comunicación (TICS).

Sublíneas de investigación de la Carrera: Inteligencia Artificial e Inteligencia de Negocios



2. INTRODUCCIÓN

En Ecuador, al igual que en muchos países, la agricultura representa un pilar fundamental en la economía y la seguridad alimentaria, es por ello que pocos proyectos se orientan hacia el aporte a la agricultura inteligente.

La detección de plagas y enfermedades de plantas a través de la observación visual de los síntomas por medio de las hojas implica un grado de complejidad, incluso agrónomos experimentados no logran diagnosticar con éxito las enfermedades, en consecuencia, se tiende a conclusiones y tratamientos erróneos.

En este contexto, existe un proyecto que radica en el desarrollo de un modelo de detección de la enfermedad tizón tardío en los cultivos de papa aprovechando la fuerza de la Inteligencia Artificial y el aprendizaje profundo, modelo al cual también se lo entrenó para detectar el tizón temprano ya que comparten formas, colores, y texturas parecidas [1].

En este sentido el presente proyecto de investigación se realiza en el cantón Latacunga, provincia de Cotopaxi, con un enfoque específico en la problemática que afecta a los cultivos de papa, por tal motivo el objetivo de la investigación es desarrollar un prototipo de un sistema para la identificación automática de plagas y enfermedades en el cultivo de papa, utilizando técnicas de inteligencia artificial. Para lograr este propósito, se emplearán metodologías de investigación de campo, tecnológica y aplicada, que permitirán una comprensión de la problemática existente y la identificación de las principales necesidades.

Para el desarrollo del proyecto se utilizan las herramientas de software como Google Colab, Tensorflow, para llevar a cabo el proceso de entrenamiento del modelo de inteligencia artificial, además, se empleará Android Studio para modelar tanto la interfaz y la lógica del sistema, con el fin de presentar de manera efectiva los resultados obtenidos durante el entrenamiento del modelo.

Con el fin de ayudar al agricultor en la identificación de plagas o enfermedades existentes en el cultivo de papa, se llevará a cabo un análisis de modelos previos investigados, estos modelos preexistentes se seleccionarán y ejecutarán con el fin de evaluar su eficacia, el resultado del análisis servirá para la elección de un modelo idóneo, que permitirá usarlo como base para la creación de un modelo personalizado para la identificación de las plagas o enfermedades en la hoja de papa. La presentación de los resultados se realizará a través de una interfaz móvil intuitiva, facilitando la interpretación de los datos por parte de los agricultores.



El presente documento se estructura en varias secciones para representar de manera clara y organizada la investigación realizada, la primera sección contiene la información general, mientras que la segunda sección aborda la introducción en la que se abarca los niveles micro, meso y macro. En la tercera sección, se presenta la fundamentación teórica que explora temas relacionados a la variable dependiente y la variable independiente. Posteriormente, la sección cuatro, detalla las metodologías empleadas tanto para la creación y entrenamiento del modelo de inteligencia artificial como para el desarrollo del prototipo de la aplicación móvil. Después, la quinta sección abarca el análisis de resultados obtenidos y la validación de la hipótesis. Luego, en la sexta sección se representan las conclusiones y recomendaciones derivadas a la investigación, Finalmente, la sección siete contiene las referencias de la investigación, mientras que en la octava se incluyen los anexos, como tablas, fotos e imágenes pertinentes a la investigación.

2.1. PROBLEMA

2.1.1. Situación Problemática

En la industria agrícola la producción de cultivos se enfrenta a una lucha constante contra la presencia de factores bióticos y abióticos que afectan a la salud de las plantas y la productividad de los cultivos a nivel mundial. Entre estas amenazas los factores bióticos como las plagas y enfermedades representan desafíos críticos que impactan negativamente a la agricultura.

Una de las principales enfermedades que afecta la papa en el mundo es el tizón tardío, provocado por el oomicete *Phytophthora infestans*, que puede destruir los cultivos en pocos días si no se toman acciones y que provoca grandes pérdidas económicas que llegan a 3,5 billones de dólares anuales en países en vías de desarrollo y más de 5 billones en todo el mundo[2].

La papa (*Solanum tuberosum* L.) es uno de los cultivos más importante para la seguridad alimentaria de muchas familias en el Ecuador. Uno de los factores bióticos más importantes que afectan a este cultivo en la enfermedad conocida como tizón tardío o lancha (TTP), que aparece en condiciones de alta humedad relativa y temperaturas entre de 15° a 21°C, llegando a producir pérdidas hasta del 100% si no se realiza un manejo oportuno [3].

Según el Centro Internacional de la Papa (CIP), el cultivo de papa en la región andina se encuentra en alerta ante la posibilidad de diseminación de tres plagas que dañan seriamente al



cultivo: la punta morada, el manchado interno de la papa, también conocido como “zebra chip”, y el psílido de la papa [4].

Hernán López, coordinador distrital del Ministerio de Agricultura y Ganadería (MAG), comentó que se ha detectado la presencia de la plaga en Pastocalle y Salache, para esto se activa un plan emergente para dar posibles soluciones. Destacó que se mantienen recorridos en todos los sectores productivos de Cotopaxi para poder determinar si la bacteria está presente [5].

En el año 2020, según Darwin Tapia Coordinador Técnico de la Empresa INTEROC, la misma que se dedica al expendio de Agroquímicos, señala que la punta morada y la lancha son las principales enfermedades que asechan a los cultivos de papa en el cantón Latacunga [6]. Del mismo modo aparecen las plagas como el gusano blanco, la polilla que dañan el tubérculo y los gusanos cortadores que perjudican a las hojas.

A pesar de que existen aplicaciones para la identificación de plagas y enfermedades en la agricultura como en el estudio Identification of Leaf Diseases in Potato Crop Using Deep Convolutional Neural Networks [7], la ausencia de una aplicación agrava esta problemática específicamente en la ciudad de Latacunga. De esta manera la falta de aplicaciones específicas para este propósito dificulta la toma de decisiones asertivas para el control de medidas efectivas en los cultivos de la zona contra los problemas fitosanitarios.

2.1.2. Formulación del problema

¿El desarrollo de una aplicación móvil basada en el uso de Inteligencia Artificial, ayudará a la detección de plagas y enfermedades en el cultivo de la papa andina?

2.2. OBJETO Y CAMPO DE ACCIÓN

Objeto de Investigación: Identificación de plagas y enfermedades en el cultivo de papa en la ciudad de Latacunga.

Campo de Acción: 1203.04 Inteligencia Artificial.

2.3. BENEFICIARIOS

2.3.1. Beneficiarios Directos

Los agricultores de Latacunga se convertirán en beneficiarios esenciales de la aplicación móvil desarrollada a través de esta investigación. Esta herramienta les permitirá identificar plagas y



enfermedades en sus cultivos de papa, lo que resultará en una toma de decisiones más efectiva y un mejor control de plagas.

Los estudiantes y profesores de la Universidad Técnica de Cotopaxi que pertenecen a la Carrera de Agronomía desempeñarán un papel directo como beneficiarios a medida que la investigación avance. Su contribución al proporcionar datos e imágenes de cultivos de papa no solo enriquecerá el proyecto, sino que también les brindará la oportunidad de utilizar la aplicación como una valiosa herramienta de aprendizaje y análisis en su campo de estudio.

2.3.2. Beneficiarios Indirectos

Los consumidores finales de papas en la ciudad de Latacunga experimentarán beneficios indirectos derivados de esta investigación. Puesto que, la detección de enfermedades diagnosticadas por la aplicación y además el control eficiente de plagas y enfermedades que realicen los agricultores en los cultivos, contribuirán para ofrecer un producto saludable.

Además, investigadores y académicos interesados en los campos de la agricultura, la inteligencia artificial y la detección de plagas y enfermedades en cultivos podrán aprovechar esta investigación como una referencia valiosa para futuros proyectos.

2.4. JUSTIFICACIÓN

La investigación propuesta sobre el "Desarrollo de un prototipo para la identificación automática de plagas y enfermedades en el cultivo de papa, utilizando técnicas de Inteligencia Artificial, en la ciudad de Latacunga" será de gran importancia y aportará valiosos beneficios en diversos aspectos la productividad de la papa.

En primer lugar, la detección temprana de plagas y enfermedades en las plantas es de vital importancia para la seguridad alimentaria y la protección del medio ambiente. Las plagas y enfermedades pueden causar graves daños a los cultivos, lo que provoca gran pérdida económica y afecta la disponibilidad de alimentos para la población. Entonces al utilizar técnicas de inteligencia artificial, la precisión y velocidad en la detección pueden aumentar considerablemente, permitiendo una respuesta más rápida y eficiente para controlar y mitigar las afectaciones a los cultivos.

Además, la aplicación de técnicas de inteligencia artificial en este ámbito puede representar un avance trascendental en el conocimiento científico y tecnológico. La combinación de la investigación en la agricultura y la inteligencia artificial puede abrir nuevas perspectivas para



abordar problemas complejos y mejorar la toma de decisiones en el manejo preventivo de plagas y enfermedades.

Desde el punto de vista teórico, esta investigación podría demostrar el uso novedoso de técnicas de inteligencia artificial para resolver problemas específicos en el campo agrícola. Además, se espera que los resultados obtenidos proporcionen información valiosa para futuras investigaciones relacionadas con la optimización y mejora de las técnicas empleadas.

En cuanto al enfoque práctico, el desarrollo de un sistema de detección de plagas y enfermedades basado en inteligencia artificial tendría un impacto directo en el sector agrícola, ya que facilitaría la identificación temprana de problemas y permitiría la implementación oportuna de medidas preventivas y correctivas. Esto contribuiría a la reducción del uso de agroquímicos y a una agricultura más sostenible y considerada con el medio ambiente.

2.5. HIPÓTESIS

La implementación de un prototipo de software basado en Inteligencia Artificial permite la identificación automática de plagas y enfermedades en el cultivo de papa.

Variable dependiente

Identificación automática de plagas y enfermedades del cultivo de papa.

Variable Independiente

Desarrollo de un prototipo para la identificación automática de plagas y enfermedades en el cultivo de papa.

2.6. OBJETIVOS

2.6.1. Objetivo General

Desarrollar un prototipo de software mediante técnicas de inteligencia artificial para la identificación automática de plagas y enfermedades en los cultivos de papa en la ciudad de Latacunga.

2.6.2. Objetivos Específicos

- Realizar una revisión de la literatura científica sobre los modelos de visión artificial aplicados a la identificación de plagas y enfermedades en los cultivos de papa y aplicaciones móviles para la redacción de la fundamentación teórica.



- Aplicar la metodología KDD de manera sistemática en el desarrollo del prototipo para la extracción del conocimiento del conjunto de imágenes digitales acerca de plagas y enfermedades de la papa.
- Desarrollar un prototipo de aplicación móvil a partir de un modelo de Inteligencia Artificial para la predicción de plagas y enfermedades en la papa.

2.7. SISTEMA DE TAREAS

Tabla 1. Sistema de Tareas

Objetivos específicos	Actividades (tareas)	Resultados esperados	Técnicas, Medios e Instrumentos
Realizar una revisión de la literatura científica sobre los modelos de visión artificial aplicados a la identificación de plagas y enfermedades en los cultivos de papa y aplicaciones móviles para la redacción de la fundamentación teórica.	Búsqueda de información en revistas científicas.	Fundamentación teórica	Técnica: Análisis bibliográfico. Medios: Mendeley Instrumentos: Ficha bibliográfica
	Seleccionar los modelos de visión artificial más eficientes.	Técnica de Inteligencia Artificial	Técnica: Análisis bibliográfico Medios: Mendeley Instrumentos: Ficha bibliográfica
Aplicar la metodología KDD de manera sistemática en el desarrollo del prototipo para la extracción del conocimiento del conjunto de imágenes digitales acerca de plagas	Aplicación de la metodología KDD.	Dos modelos CNN entrenados	Técnica: Aplicar algoritmos de inteligencia artificial para identificar patrones y relaciones en el conjunto de datos de imágenes. Medios e Instrumentos: Herramientas de inteligencia artificial como Google Colab



y enfermedades de la papa.	Aplicación de la metodología KDD.	Tres Modelos CNN entrenados	Técnica: Aplicar algoritmos de inteligencia artificial para identificar patrones y relaciones en el conjunto de datos de imágenes. Medios e Instrumentos: Herramientas de inteligencia artificial como Google Colab
Desarrollar una aplicación móvil que despliegue el modelo de inteligencia artificial entrenado para la identificación de plagas y enfermedades de la papa.	Desarrollo de la Interfaz Gráfica de Usuario (GUI) mediante aplicaciones web.	Interfaz Gráfica de Usuario (GUI)	Técnica: diseño Front-end del prototipo. Medios e Instrumentos: Uso de Figma para el diseño de prototipos.
	Desarrollo del código fuente a partir de herramientas de software libre.	Aplicación móvil	Técnica: prototipo de la aplicación Medios e Instrumentos: Uso de Android Studio y kotlin como lenguaje de desarrollo, además de la biblioteca Tensorflowlite.
	Creación de un plan de pruebas	Plan de pruebas	Técnica: Diseñar casos de prueba. Medios e Instrumentos: Herramientas de gestión de pruebas, hojas de cálculo.
	Despliegue del prototipo en una tienda de aplicaciones.	Despliegue de la aplicación móvil en la Play Store	Técnica: Compilación y optimización de la aplicación para la plataforma de distribución. Medios e Instrumentos: Uso de herramientas integradas en Android Studio para generar archivos de instalación APK.

* Elaborado por: Los investigadores



3. FUNDAMENTACIÓN TEÓRICA

Una de las principales limitaciones de esta revisión de la literatura reside en la escasa disponibilidad de información proveniente de revistas científicas que aborden proyectos de magnitud similar en el contexto específico de Ecuador, más concretamente en la región de Cotopaxi y la ciudad de Latacunga. Esta limitación dificulta la recopilación de datos y referencias directamente relacionadas con el ámbito geográfico y temático de la presente investigación. A pesar de esta restricción, se ha llevado a cabo una exhaustiva búsqueda y análisis de la literatura científica más reciente.

A continuación, se presentarán algunos de los artículos seleccionados que destacan las partes más relevantes que contribuyen al conocimiento y desarrollo de soluciones en la agricultura inteligente.

En el artículo titulado "Detección de hojas de plantas y reconocimiento de enfermedades utilizando Deep Learning", se logra una precisión del 96,5% [8]. El sistema se ha diseñado con el propósito de identificar y clasificar varias variedades de plantas, incluyendo manzana, maíz, uva, papa, caña de azúcar y tomate. Además, el sistema es capaz de identificar múltiples enfermedades que afectan a estas plantas. Mediante el uso de un conjunto de datos compuesto por 35,000 imágenes de hojas de plantas, tanto sanas como infectadas con enfermedades, los investigadores lograron entrenar modelos de aprendizaje profundo para detectar y reconocer enfermedades en las plantas, así como determinar la ausencia de estas enfermedades.

En otro estudio denominado "Detección automatizada de enfermedades foliares en diferentes especies de cultivos mediante análisis de características de imágenes y clasificadores de una clase," se logra una destacada precisión del 95% [9]. La metodología propuesta se basa en el uso de clasificadores de una sola clase específica para cada condición de salud de la planta, entre ellas están la presencia de mildiu vellosa, oídio y podredumbre negra. De esta manera, los algoritmos previamente entrenados en hojas de vid se han sometido a pruebas en una variedad de cultivos, demostrando una notable capacidad al ser aplicados en otros tipos de plantas. De esta manera, el algoritmo de este estudio resuelve posibles conflictos entre los clasificadores de una sola clase, garantizando así la correcta identificación de hojas enfermas en distintas condiciones.

De acuerdo con este enfoque, el artículo presenta un modelo de redes neuronales, el cual logra con éxito clasificar las hojas de papa en tres categorías: hojas sanas, hojas con tizón temprano



y hojas con tizón tardío. De este modo, se destacan la efectividad de la visión por computadora y el aprendizaje profundo en la identificación de enfermedades foliares en los cultivos de papas.

En otro estudio, como el "Reconocimiento automático de lesiones de tizón tardío y cuantificación de la severidad a partir de imágenes de campo de diversos genotipos de papa mediante Deep learning," se logra una precisión sobresaliente del 99.6%. Para llevar a cabo este estudio, se recopilaron alrededor de 500 imágenes RGB de campo de diversos genotipos de patata con diferentes niveles de gravedad de la enfermedad [10].

Dentro de este contexto, se usaron 2100 imágenes recortadas a partir de las 500 imágenes iniciales, pues 1600 para entrenar y 250 para validación. Además, se estableció una fuerte relación lineal ($R^2=0.655$) entre las puntuaciones manuales del tizón tardío y las lesiones detectadas por el aprendizaje profundo, resaltando su potencial en la detección precisa de tizón tardío en diversas variedades de la papa.

En otro estudio, Potato diseases detection and classification using deep learning methods En este trabajo, utilizamos los métodos de redes neuronales de convolución (CNN), además, examinamos 5 clases de enfermedades de la patata con los nombres: Sana, Sarna Negra, Sarna Común, Pata Negra, Pudrición Rosada. Se utilizó una base de datos de 5000 imágenes de patatas. Comparamos los resultados de nuestros métodos de clasificación de defectos de la patata con otros métodos como Alexnet, Googlenet, VGG, R-CNN, Transfer Learning [11].

Esta investigación alcanzó entre una precisión del 100% al 99% en algunas clases utilizando redes neuronales convolucionales para el análisis de 5 enfermedades de la papa como: la sarna negra, sarna común, pata negra y pudrición rosada. Usaron modelos basados en CNN lo cual presentó una precisión del alto nivel en comparación a las otras técnicas usando Transfer Learning.

3.1. Inteligencia artificial

Según [12] la inteligencia artificial (IA) se define como un ámbito en la ciencia y la ingeniería que se centra en la comprensión, desde una perspectiva informática, de lo que comúnmente se conoce como comportamiento inteligente, además, se involucra en la creación de artefactos capaces de exhibir dicho comportamiento.

De acuerdo, la inteligencia artificial permite adaptar el aprendizaje experimental a las necesidades individuales.



3.2. Computer visión

La visión computacional constituye un campo interdisciplinario, engloba métodos para capturar, procesar, analizar, comprender imágenes y videos del entorno real con el propósito de emular la capacidad visual del humano, la visión artificial examina y procesa la profundidad en imágenes infrarrojas y se encarga de la extracción de información de las imágenes y videos [13].

3.2.1. Procesamiento de Imágenes

Hace referencia a la automatización, manipulación, análisis e interpretación de imágenes mediante algoritmos y códigos en un entorno computacional, este proceso está adaptado en diversas disciplinas como: campos científicos y tecnológicos, un ejemplo claro, son las redes sociales, que diariamente se comparten numerosas imágenes y que con el uso de estos algoritmos las procesan para analizar qué clase de imagen se sube a la plataforma [13].

3.2.1.1. Adquisición y almacenamiento

Sensores utilizados para la captura de imágenes como: cámara digital, aparato de rayos X, aparatos de ultra sonido, escáner, etc., la imagen debe ser almacenada en algún dispositivo como: disco duro como un archivo y con su respectiva extensión, por lo general un archivo JPEG, el archivo debe ser serializado antes de ejecutar el algoritmo [13].

3.2.1.2. Preprocesamiento

Para una mejor manipulación, se requiere ejecutar algunos algoritmos de preprocesamiento como: la conversión de la imagen a una escala de grises, mejorar la calidad de imagen con un filtrado de desenfoque y degradación del ruido, modificar a un tamaño estándar de la imagen [13].

3.2.1.3. Segmentación

Es necesario realizar la segmentación de la imagen en una o varias imágenes con el fin de simplificar la extracción de características del objeto que se busca tratar [13].

3.2.1.4. Extracción de características

Es fundamental representar la imagen de manera alternativa, permitiendo así la extracción automática de características, tales como los valores de pesos y sesgos aprendidos en las capas ocultas de una red con aprendizaje profundo [13].



3.2.1.5. Reconocimiento de patrones

La comprensión e interpretación de la imagen implica la creación de una representación que se utilizará para una mayor compresión visual, además la representación tiene aplicaciones específicas, como la clasificación de imágenes y la delimitación de la ubicación de objetos [13].

3.2.1.6. Resultado

Se efectúa el procesamiento de nivel superior, en el que se identifica el objeto segmentado, a través de sus características distintivas y de este modo poder clasificarlo en una categoría específica [13].

3.3. Machine Learning

También llamado aprendizaje automático o aprendizaje de máquina, conforma una subdivisión de la inteligencia artificial (IA) que ha experimentado un crecimiento notorio en los últimos años, ha ofrecido diversas técnicas y algoritmos destinados a anticipar situaciones basándose en extensas cantidades de información, con un adecuado procesamiento y filtrado de datos, es una herramienta con la capacidad de generar predicciones altamente efectivas [14].

3.3.1. Aprendizaje Supervisado

El aprendizaje es supervisado implica la utilización de conjuntos de datos etiquetados para instruir a los algoritmos, permitiéndoles clasificar datos o prever resultados con precisión, estos algoritmos absorben información de los datos con el propósito de identificar patrones o tendencias, facilitando la comprensión de los datos y la construcción de un modelo para predecir y clasificar los elementos [15].

Estoy de acuerdo puesto que, durante el proceso, el algoritmo realiza predicciones que se validan hasta lograr un alto nivel de precisión y rendimiento, entre los métodos más comunes en el aprendizaje supervisado se encuentran las redes neuronales, Naive Bayes, regresión lineal, random forest, entre otros.

3.3.2. Aprendizaje No Supervisado

El aprendizaje no supervisado, su enfoque se encuentra en que los datos de entrenamiento no están etiquetados, y es el algoritmo el que se encargará de clasificar la información por sí mismo, su objetivo es buscar patrones subyacentes y agrupamientos naturales, los dos enfoques



comunes en el aprendizaje supervisado son el: agrupamiento o clustering y la reducción de la dimensionalidad [15].

3.4. Deep Learning

Dentro de la rama del machine learning, el deep learning es empleado para reconocer objetos en imágenes, seleccionar resultados de búsqueda, vincular productos o noticias con los intereses de los usuarios, esta rama es altamente eficaz para descubrir estructuras complejas en conjuntos de datos de alta dimensión, siendo aplicable en diversos campos como: la ciencia, negocios, etc. [16].

3.5. Redes neuronales convolucionales (CNN)

Es una arquitectura de red neuronal artificial que incorpora múltiples capas para el cálculo de resultados a partir de un conjunto de datos, las neuronas artificiales en este tipo de estructura se asemejan al funcionamiento de las neuronas de un cerebro biológico, este diseño en particular se lo utiliza para el procesamiento de imágenes, en lo que respecta a la clasificación y segmentación [17].

3.5.1. Hiperparámetros.

En una red neuronal, los parámetros e hiperparámetros son los valores encargados de regir el funcionamiento de la red. Estos valores serán ajustados a lo largo del desarrollo de una red neuronal recurrente, y determinarán los comportamientos de la red en todos sus niveles [18].

3.5.1.1. Número de capas

Se refiere a la cantidad de capas de neuronas que conforman la red. Una red con más capas puede aprender funciones más complejas, pero también es más difícil de entrenar [19].

3.5.1.2. Tipo de algoritmo de entrenamiento

El algoritmo de entrenamiento es el algoritmo que se utiliza para ajustar los pesos de la red. El algoritmo de entrenamiento afecta a la velocidad y la eficacia con la que la red aprende. Algunos algoritmos de entrenamiento son más adecuados para ciertas tareas que otros [19].



3.5.1.3. Función de pérdida

Es una herramienta matemática empleada para cuantificar el error de la red neuronal. La función de pérdida afecta a la forma en que la red aprende. Algunas funciones de pérdida son más adecuadas para ciertas tareas que otras [19]

3.5.1.4. Tasa de Aprendizaje

El sobreajuste sucede cuando la red neuronal se sobre-entrena con ciertos datos, a tal nivel que un dato con una pequeña variación diferente a la del conjunto de entrenamiento, pero de la misma clase, arrojará un mayor error y puede clasificarse en la clase incorrecta [20].

3.5.1.5. Sobreajuste (Overfitting)

El sobreajuste ocurre cuando la red neuronal se adapta excesivamente a los datos de entrenamiento y pierde su capacidad para generalizar correctamente a nuevos datos. Esto puede ocurrir si la red es demasiado grande o si la tasa de aprendizaje es demasiado alta [19].

3.5.1.6. Bajo ajuste (Underfitting)

Se refiere cuando un modelo tiene un rendimiento deficiente con los datos de entrenamiento, y además, no es capaz de generalizar correctamente los datos nuevos, ya que tendrá un bajo nivel de rendimiento en el conjunto de datos de entrenamiento, esto puede ser detectado mediante métricas de rendimiento del modelo [21] .

3.5.1.7. Optimizador

Los optimizadores en las redes neuronales facilitan el ajuste de los pesos, que son el número de neuronas que agregamos en las capas, para minimizar la función de error y de esta manera obtener una mayor facilidad en el entrenamiento y por ende una mejor precisión [22].

3.5.1.8. Tasa de aprendizaje

Es un parámetro que regula la rapidez con la que se modifican los pesos del modelo en cada iteración del entrenamiento, fundamentándose en la gradiente de la función de pérdida, este valor se puede modificar y adaptar acorde al modelo que se requiera entrenar, de esto dependerá el tiempo y eficacia de entrenamiento [22].



3.5.1.9. Funciones de activación

Las funciones de activación son componentes importantes en el aprendizaje profundo, puesto que influyen en el resultado de precisión y eficiencia computacional de entrenamiento de un modelo de aprendizaje profundo, además, afectan en la capacidad de convergencia de la red neuronal y la velocidad de aprendizaje, estas funciones que son ecuaciones matemáticas determinan la salida de una red neuronal, su principal función es normalizar la salida de cada neurona dentro de un rango específico, como 0 y 1 ó -1 y 1 [21].

3.5.1.10. Tamaño de lote

Es la cantidad de imágenes de entrenamiento que se procesan simultáneamente en cada iteración durante el proceso de entrenamiento de aprendizaje automático, el tamaño del lote es un valor que se puede modificar acorde a las necesidades del entrenamiento del modelo [23].

3.5.1.11. Etiquetado de imágenes

El etiquetado de imágenes indica el nombre que se coloca a la carpeta dentro de nuestro dataset, esto dependerá del contexto en el que se esté trabajando y por lo general se lo coloca con separaciones de guiones bajos entre las palabras, esto con el fin de no tener inconvenientes al momento de predecir o entrenar el modelo con nuestro dataset.

3.5.2. Transfer learning

El aprendizaje por transferencia implica mejorar el proceso de aprendizaje al aplicar conocimientos adquiridos previamente en tareas relacionadas, se enfoca en desarrollar métodos que permitan aprovechar la experiencia acumulada con los conocimientos aprendidos en una o varias tareas con el fin de mejorar el aprendizaje de una tarea destino [24].

Haciendo una comparativa con el ser humano, también se maneja este tipo de transferencia de conocimientos entre tareas, donde la adquisición del conocimiento se aplica de manera pertinente a nuevas tareas, basándose en experiencias de aprendizaje anteriores.

3.5.3. Modelo ResNet50

Es una estructura de redes neuronales convolucionales, creada por Microsoft Research en el año 2015, la arquitectura ha intervenido notablemente en el ámbito del aprendizaje profundo



con resultados sobresalientes en diversas funciones de visión por computadora como: la clasificación de imágenes, detección de objetos y segmentación semántica [25].

3.5.4. Modelo InceptionV3

Inception v3 es un modelo de reconocimiento de imágenes que puede alcanzar una exactitud superior al 78.1% en el conjunto de datos de ImageNet. El modelo representa la culminación de muchas ideas que desarrollaron varios investigadores durante años [26].

3.5.5. Modelo VGG19

Presentado por el equipo de investigación Visual Geometry Group (VGG) de la Universidad de Oxford en el año 2014, la red neuronal está compuesta por 19 capas que incorporan tanto capas convolucionales, como de agrupación, se destaca por su rendimiento en la clasificación de imágenes, su principal característica radica en la habilidad de obtener representaciones profundas y significativas [25].

3.5.6. Modelo EfficientNet

En el año 2019, Google IA introdujo un modelo que se desarrolló mediante un enfoque sistemático y más eficiente, reduciendo su tamaño a la mitad. Este logro se basó en la aplicación del método de escalado de modelos, que implica la adaptación de un modelo estándar al ajustar hacia abajo no solo el tamaño de las imágenes de entrada, sino también la profundidad y la amplitud de la red [15].

3.5.7. Modelo MobileNet

Es una arquitectura que goza de popularidad gracias a su rapidez y eficiencia en la detección de objetos en tiempo real, emplea capas convolucionales separables en profundidad para disminuir la cantidad de parámetros y mejorar la eficiencia computacional [27].

3.5.8. Aplicaciones de las Redes Neuronales Convolucionales (CNN)

Muchas de las aplicaciones de las redes neuronales convolucionales se destacó los artículos más relevantes que contribuyen al conocimiento y desarrollo en soluciones en la agricultura inteligente.

De acuerdo con los autores en el artículo “Recognition of plant leaf diseases based on computer vision”, adquirieron un total de 1000 imágenes de hojas de enfermedades como el arroz y el

pepino y la papa entre ella el tizón temprano y tizón tardío formato jpg el modelo ResNet-V2 logra una precisión del 99,87% [28]

Utilizaron un enfoque de segmentación y clasificación combinado para identificar de manera precisa las enfermedades gracias a un algoritmo de segmentación híbrida basado en tono, saturación e intensidad, luego aquellas imágenes segmentadas se sometieron a una red neuronal convolucional para su respectiva clasificación.

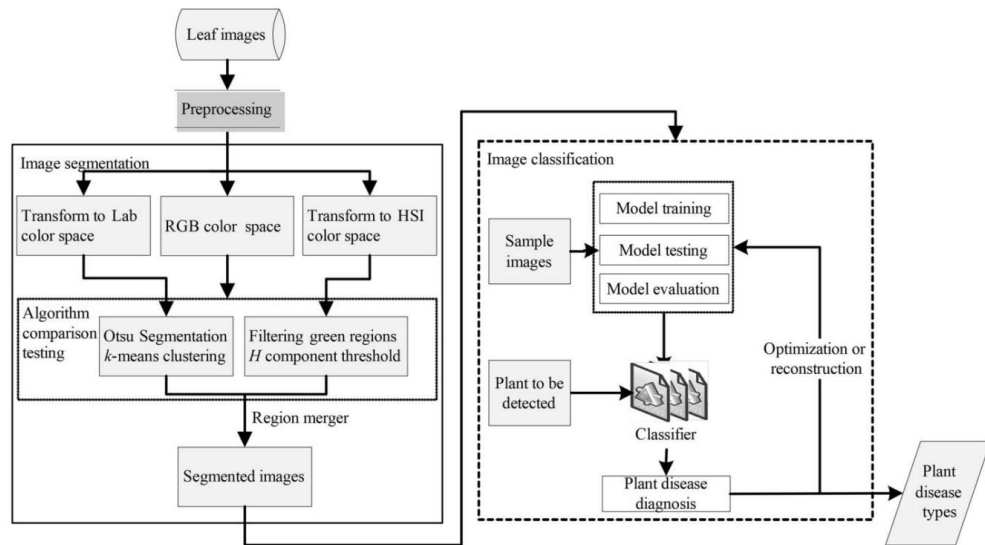


Figura 1. Diagrama de flujo del modelo [28]

Según los autores del artículo “Effective multi-crop disease detection using pruned complete concatenated deep learning model” mencionan que el conjunto de datos lo obtuvieron de Plant Village donde abarcar imágenes de hojas tanto sanas como enfermas de plantas como manzanas, maíz, tomate y papa. Entre las enfermedades que tratan son podredumbre negra, mancha gris de la hoja, moho de la hoja, tizón temprano y tardío. Proponen la arquitectura Complete Concatenated Deep (CCDL), la cual logra una precisión del 98,14% [29].

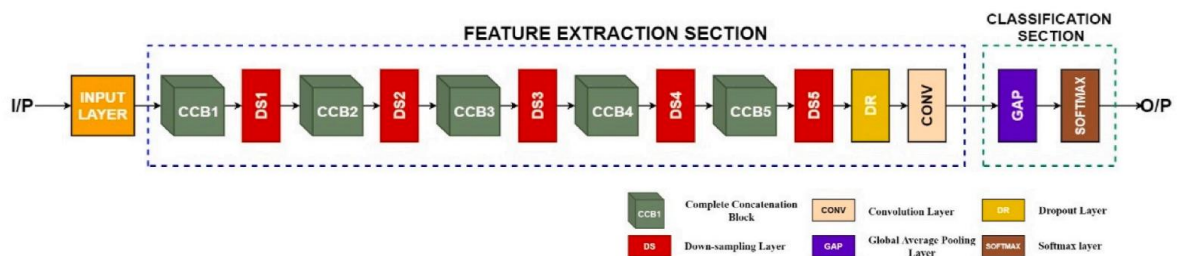


Figura 2. Estructura del modelo Complete Concatenated Deep Learning (CCDL) [29]

Otro autores en el artículo “Automatic late blight lesion recognition and severity quantification based on field imagery of diverse potato genotypes by deep learning” mencionan que obtuvieron 500 imágenes propias que luego

mediante técnicas de aumento de datos se obtuvo 2100 imágenes papa tizón tardío con modelo SegNet con una precisión del 99,60% [30].

Con un algoritmo de aprendizaje profundo basado en una arquitectura codificador-decodificador segmentación semántica de lesiones de la enfermedad tizón tardío.

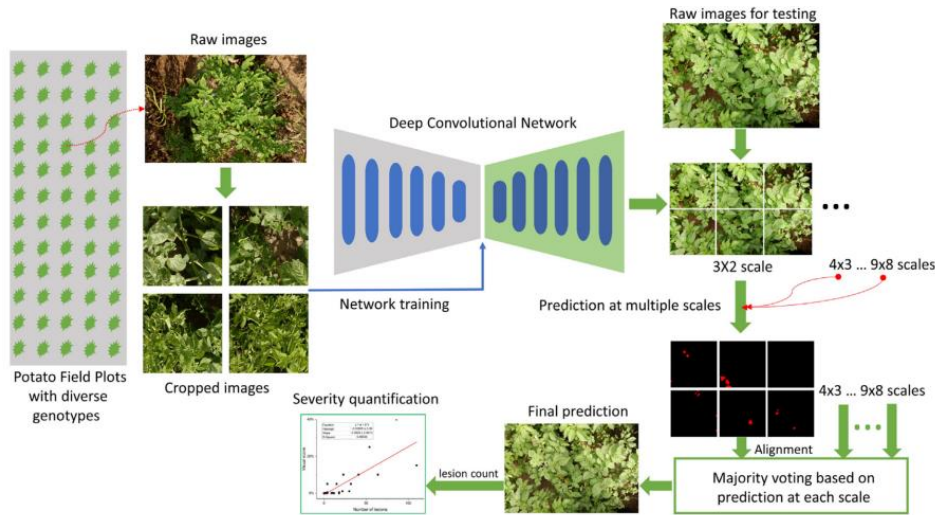


Figura 3. Resumen esquemático del proyecto [30]

Según los autores del artículo “Potato diseases detection and classification using deep learning methods”, utilizaron 5000 imágenes las enfermedades son sarna común, costra negra, putrefacción rosada con una arquitectura propia logran una precisión del 99,00%. Además se compararon con otros modelos como AlexNet, GoogleNet, VGG, R-CNN [31].

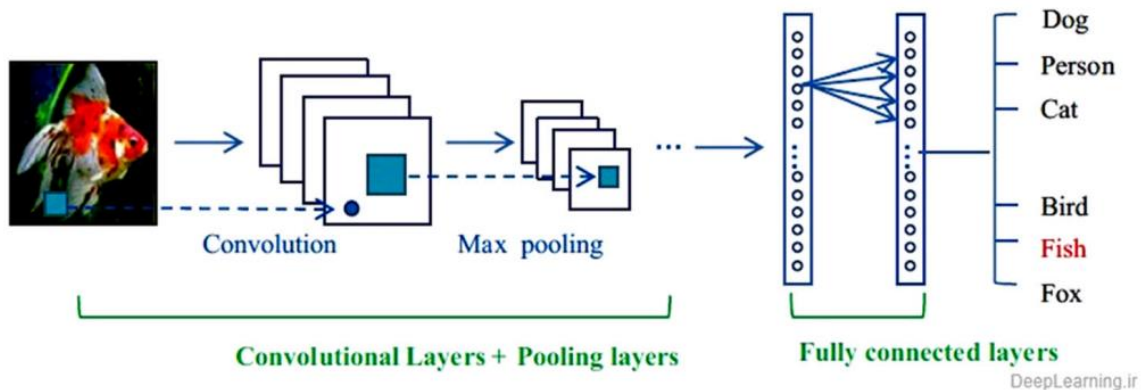


Figura 4. Arquitectura de la red neuronal convolucional [31]

Según los autores del artículo “RDA-CNN: Enhanced Super Resolution Method for Rice Plant Disease Classification”, obtienen imágenes de arroz usa capas de super-resolución para transformar las imágenes de baja resolución las enfermedades que tratan son: machas foliares, podredumbre negra y lesiones foliares en diferentes partes de las plantas de arroz el método RDA-CNN propuesto logra una precisión del 93,26% [32].

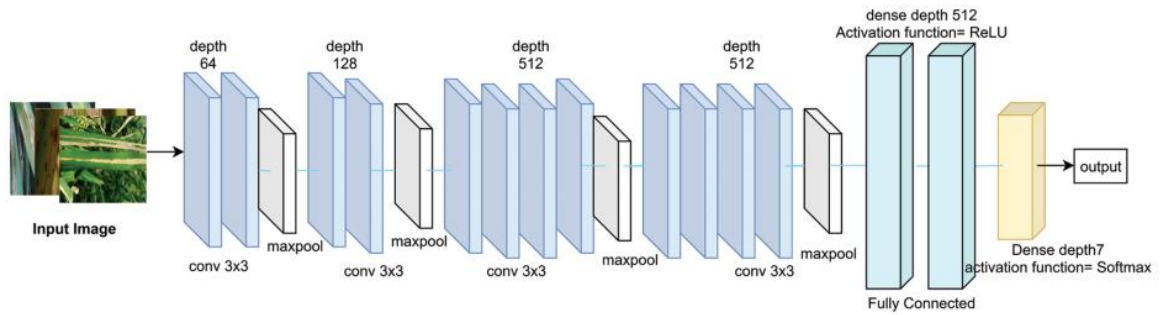


Figura 5. Estructura de la Arquitectura propuesta [32]

Según los autores en el artículo “CNN Based Detection of Healthy and Unhealthy Wheat Crop”, proponen un nuevo método para la identificación de plantas de trigo donde se utiliza una red neuronal convolucional profunda (DCNN) y aprendizaje por transferencia para el entranamiento del modelo la cual alcanza una precisión del 98% [33].

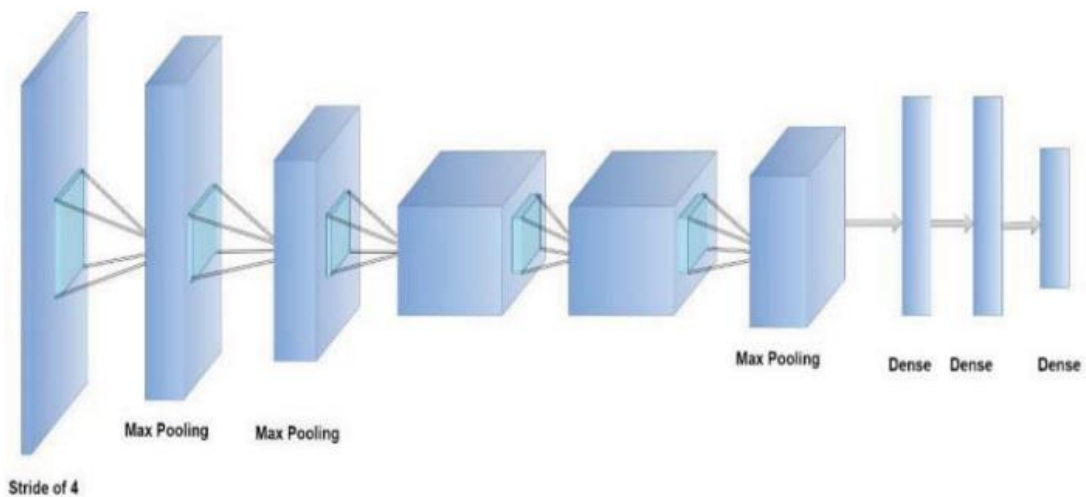


Figura 6. Estructura del modelo pre-entrenado AlexNet [33]

Entrenaron varios modelos CNN para comparar resultados entre ellos: VGG16, VGG19, AlexNet, ResNet-34, ResNet-101, ResNet-50 y ResNet18 la técnica es beneficiosa para los agricultores que puedan identificar la zona estropeada del cultivo.

3.6. Construcción y validación del modelo

Con los datos estructurados y procesados podemos modelar el aprendizaje, se construye el clasificador con el grupo de datos cargados en el dataloaders y la arquitectura definida, la calidad y rapidez de clasificación de los datos, dependerá directamente de la arquitectura electa [15].



De acuerdo con lo citado es muy importante tener en consideración la arquitectura o modelo que se va implementar, identificar los requerimientos acordes a nuestro tema a tratar, será de vital importancia para tener un resultado de calidad sin mucho margen de error, por ello es importante trabajar con varios métodos, basándonos en los resultados obtenidos mediante la prueba y error de la construcción del clasificador.

3.6.1. Google Colab

Conocido también como Colaboratory, este servicio actúa como un servidor de Jupyter con una interfaz diseñada para integrarse con Google Drive y GitHub que es en donde se distribuyen los notebooks originales, además, facilita la ejecución de notebooks tanto en la nube como de forma local, los mismos que se guardan directamente en la cuenta de Google Drive, permitiendo su fácil compartición con distintos usuarios, la ventaja es que no se requiere realizar ningún tipo de mantenimiento en el servidor, ya que no almacena ningún archivo localmente[34].

3.6.2. Entrenamiento y validación del modelo

Con el modelo de aprendizaje construido acorde a las características de la arquitectura y los datos cargados, se procede a entrenar el modelo, en esta fase el modelo se entrena para que aprenda a clasificar las imágenes y se evalúa si las clasificaciones son correctas o no [15].

3.6.3. Epochs

Un epoch representa un ciclo completo a lo largo de todo el conjunto de datos de entrenamiento, al establecer el número de epochs para entrenar nuestra red neuronal, estamos determinando cuántas iteraciones completas el algoritmo de aprendizaje automático realizará en nuestro conjunto de datos durante el proceso de entrenamiento, el número de epochs se determina considerando el tiempo disponible para el entrenamiento y los resultados obtenidos durante la ejecución, hay que tener en cuenta que existe un punto en el cual la red deja de aprender, y en cambio, comienza a mostrar un deterioro en los resultados del conjunto de validación [15].

3.6.4. Keras

Se trata de una biblioteca especializada en aprendizaje profundo, constituyendo una interfaz de programación de aplicaciones (API) diseñada para redes neuronales, la misma que es versátil, permitiendo su ejecución en diversas herramientas de aprendizaje automática, sus características más destacadas incluyen su facilidad de uso, la disponibilidad de nuevos módulos y su compatibilidad con el lenguaje de programación Python [35].



3.6.5. Tensorflow

Es un sistema de aprendizaje automático diseñado para operar en entornos extensos, es una biblioteca de código abierto especializada en los cálculos matemáticos y utiliza la programación basada en grafos de flujo de datos, en el año 2017, Google lanzó una versión con un rendimiento optimizado con el uso de GPUs, TensorFlow optimiza el uso de numerosos servidores que cuentan con la capacidad de aprovechar el rendimiento de las GPUs, agilizando el proceso de entrenamiento con una ejecución eficiente en un extenso conjunto de servidores [35].

3.6.6. Lenguaje de programación Python

Ampliamente reconocido como uno de los lenguajes de programación más utilizados en el ámbito de la ciencia de datos, Python, creado por Guido Van Rossum en 1989, es un lenguaje de código abierto, interpretado y multiplataforma de alto nivel, su popularidad consiste no solo en su sintaxis clara y legible, sino también en su extensa variedad de librerías complementarias que amplían su funcionalidad y versatilidad [36].

3.6.7. Biblioteca TensorFlow Lite

Es un conjunto de herramientas creado para facilitar la incorporación y aplicación de técnicas de aprendizaje automático en dispositivos móviles, posibilita la conversión de modelos previamente entrenados en TensorFlow o Keras a un formato específico identificado por la extensión. tflite, a su vez, permite llevar a cabo el proceso de interpretación del modelo en una amplia variedad de dispositivos como: Android o iOS [37].

3.7. Herramientas para el desarrollo del Prototipo

3.7.1. Aplicación móvil

Es un software específico para los dispositivos móviles, su principal objetivo es la movilidad, ya que permite tener la accesibilidad en cualquier tipo de escenario, entre los sistemas operativos están: el sistema Android y IOS.

3.7.1.1. Sistema Operativo Android

Android creado por Google, es un sistema operativo móvil, su desarrollo tuvo como objetivo principal la adaptación a dispositivos móviles con pantalla táctil como: relojes inteligentes, smartphones, tabletas, televisores y Android auto, Android se posiciona como uno de los principales competidores en el mercado de los dispositivos móviles [38].



3.7.1.2. Android Studio

Fue presentado en el año 2013 y funciona como el entorno de desarrollo integrado (IDE) oficial de Android, este IDE permite desarrollar aplicaciones para una variedad de dispositivos Android y se caracteriza por ser una herramienta completa y funcional, ofrece la flexibilidad de utilizar tanto Java como Kotlin como lenguajes de programación [39].

3.7.1.3. SKD de Android Studio

Kit de desarrollo de software (SDK) es un conjunto de herramientas y bibliotecas necesarias para la creación de aplicaciones orientadas al desarrollo Android, cada ocasión que Google sube una nueva versión o actualización, se publica simultáneamente un SDK actualizado, que es importante para el manejo de la programación de los desarrolladores, puesto que ofrece novedades y mejoras [40].

3.7.1.4. Lenguaje de programación Java

Java es un lenguaje de programación orientado a objetos de alto nivel es decir es entendible para el ser humano, es un lenguaje compilado, es decir que si detecta un error el programa no compilará correctamente, java es fuertemente tipado porque los tipos de datos son necesariamente que ser declarados.

3.8. LA PAPA

La papa con su nombre científico (Solanum tuberosum) ocupa el cuarto lugar entre los alimentos más cruciales del mundo, su cultivo abarca una extensión de 17 millones de hectáreas a nivel global, con una producción mundial que alcanza los 370 millones de toneladas, beneficiando a más de 1500 millones de personas. En el Ecuador, en el año 2020 se registró una producción nacional de 408,313 toneladas de papa, la siembra de papa se concentra en la región sierra, siendo las provincias de Carchi, Chimborazo y Tungurahua las que lideran la cosecha con un 56.7%, además la papa enfrenta desafíos de factores bióticos y abióticos que impactan en la calidad del producto final [41].

3.8.1. Variedades de la papa

3.8.1.1. Papa Chola

La papa chola es una variedad destinada al consumo fresco, sus tubérculos son de tamaño mediano y forma oval-elíptica ligeramente aplanada en la parte superior e inferior, su piel es

rugosa de tono rosado y un área amarilla alrededor de los ojos que son grandes y se sitúan superficialmente [42].



Figura 7. Tubérculos de chola [42]

3.8.1.2. Papa leona negra

Es apta para el consumo en fresco, cocida, al vapor y para frituras, sus tubérculos son de forma ovalada, cuentan con ojos de profundidad media, la piel es de color rojo-morado, mientras que la pulpa es de tono amarillo con pocas manchas [42].



Figura 8. Tubérculos leona negra [42]

3.8.1.3. Papa uvilla

Adecuada en el consumo de pures, locros o para fritura, sus tubérculos tienen forma redonda y presentan ojos superficiales, la piel de estos tubérculos es de color amarillo con manchas de tono morado, mientras que su pulpa es de color crema [42].



Figura 9. Tubérculos uvilla [42]

3.8.1.4. Papa Bolona

Adecuada para sopas y pures debido a su textura harinosa, presenta tubérculos de forma redonda-ovalada ligeramente aplanada la parte apical, la piel de estos tubérculos es de tono crema-violácea, su pulpa es de color crema con pigmentación en el cilindro vascular [42].



Figura 10. Tubérculos de bolona [42]

3.8.1.5. Papa chaucha colorada

Adecuada para el consumo fresco, ya sea cocida, horneada, purés o al vapor, sus tubérculos tienen una forma elíptica y presentan ojos de profundidad mediana, la piel de estos tubérculos es de color rojo-tenue, mientras que la pulpa es de tono amarillo[42].



Figura 11. Tubérculos de chaucha colorada [42]

3.9. PLAGAS EN LA HOJA DE PAPA

3.9.1. Gusano blanco de la papa

La plaga más perjudicial en las provincias productoras de papa en el Ecuador es el gusano blanco de la papa, sus daños se reflejan en el tubérculo ocasionado por la larva que se alimenta de toda la planta de la papa, se come el borde de las hojas de la planta a forma de media luna, esta plaga puede producir la pérdida total de la cosecha [43].



Figura 12. Plaga gusano blanco [43]

3.9.2. Polilla de la papa

Pertenece a la familia de los Lepidópteros, es una plaga con relevancia en los climas cálidos, las larvas atacan a las hojas de la papa, tallos y el tubérculo, se comen el tejido interno de la hoja que forman ampollas transparentes, los tallos pueden quedar débiles y romperse, las larvas entran al tubérculo excavando túneles angostos y largos, dejando excremento en los puntos de entrada, causando pérdidas significativas a los agricultores de papa en el Ecuador [43].

3.9.2.1. Tratamiento:

Tabla 2. Tratamiento plaga polilla de la papa [44]

Fungicida	Modos de acción	Aplicación	Dosis
BELT 480 SC	Contacto Ingestión	Aspersión, intervalo de seguridad 1 día	2 – 3 kg por hectárea
NEEMIX 4.5	Contacto Ingestión	Aspersión	2 – 2.5 kg por hectárea

3.9.3. Pulguilla de la papa

Conocida como pulga saltona, es un insecto diminuto de color negro, con una longitud de 2 a 3 milímetros, tienen la capacidad de saltar fácilmente hacia las hojas de las plantas para alimentarse, causando pequeños orificios, cuando la planta se ve afectada intensamente con esta plaga, las hojas se secan, reduciendo la superficie útil de la planta, lo que complica el proceso de fotosíntesis y alimentación, en estado adulto ataca desde la germinación hasta la floración y en estado de larva atacan en la etapa de desarrollo del tubérculo [43].



Figura 13. Plaga pulguilla de la papa [43]

3.9.3.1. Tratamiento:

Tabla 3. Tratamiento pulguilla de la papa [44]

Fungicida	Modos de acción	Aplicación	Dosis
KUIK 90 PS	Contacto	Aspersión, intervalo de seguridad 14 días.	2 – 3 kg por hectárea
	Ingestión		
FOLIMART	Contacto	Aspersión	2.5 kg por hectárea
	Ingestión		
	Sistémico		

Prefenopac es un insecticida caracterizado por su efecto translaminar, y tiene actividad contra una amplia variedad de insectos, tanto masticadores como chupadores, combate con los gusanos comedores de hojas [45].

3.9.4. TRIPS

Los TRIPS son insectos difícil de identificarlos a simple vista, su longitud promedio es de 0.5-1.5 mm, su coloración varía entre tonos de amarillo pálido y marrón, estos insectos causan daño en el cultivo de papa en la fase de emergencia de plantas y en los periodos de sequía, realizan raspaduras sobre las hojas, causando deformación o marchitez de la hoja, los TRIPS dañan la planta al perforar las células de los tejidos superficiales y succionar su contenido, causando la muerte del tejido circulante [43].



Figura 14. Plaga TRIPS [43]

3.9.5. Mosca minadora

Su presencia atribuye a un díptero identificado como *Liomyza huisobrensis*, durante su fase larval la mosca minadora se convierte en la principal amenaza para el follaje de los cultivos de papa en diversas áreas de la provincia, su etapa de ataque se atribuye con mayor riesgo en el cultivo que se encuentra en la etapa de emergencia, también puede destruir plantaciones desde la etapa de germinación, las larvas forman minas a lo largo de las nervaduras centrales y laterales, produciendo secamiento y defoliación [43].



Figura 15. Plaga mosca minadora [43]

3.9.6. Gusano Trozador

Organismos identificados cuando las plantas han sufrido daños considerables, su presencia en el campo de cultivo de papa conlleva a significativas pérdidas económicas, la infestación de la plaga puede ocasionar la pérdida total de la cosecha, el insecto ataca desde la germinación hasta el establecimiento del cultivo que son a los 25 días de la siembra, las larvas muerden los tallos y destruyen a las plantas en secciones de surco, consumen las raíces, cortan el cuello de la planta y consumen las hojas tiernas, atacan en especial a las plantas jóvenes [43].

3.9.6.1. Tratamiento:

Tabla 4. Tratamiento plaga gusano trozador [44]

Fungicida	Modos de acción	Aplicación	Dosis
NEEMIX 4.5	Contacto Ingestión	Aspersión	2 – 3 kg por hectárea

3.9.7. Mosca blanca

Miembro de la familia de los aleuródidos, abarcan una diversidad de más de 1500 especies diferentes, su longitud va de 1 a 1.5 mm, con una tonalidad amarillo pálido con un par de hojas blancas, la mosca blanca cuenta con un aparato bucal de tipo picador-chupador, la presencia de esta plaga se manifiesta durante las primeras fases de crecimiento del cultivo, en los primeros 45 días, en este periodo la plaga transmite el virus que resulta en daños significativos en las plantas afectadas [43].



Figura 16. Plaga mosca blanca [43]

3.9.7.1. Tratamiento:

Tabla 5. Tratamiento plaga Mosca blanca [44]

Fungicida	Modos de acción	Aplicación	Dosis
MOVENTO	Sistémico Ingestión	Aspersión, intervalo de seguridad 7 días.	2 – 3 kg por hectárea
ACTARA 25 WG	Contacto	Al fondo del surco.	2.5 kg por hectárea

	Sistémico		
--	-----------	--	--

3.9.8. Pulgones

Son insectos diminutos con cuerpos redondeados, con una tonalidad verde o negra, pueden tener alas o carecer de ellas, estos insectos se alimentan de la savia de la planta mediante la succión, debilitándola en su proceso, su presencia suele estar en la parte inferior de las hojas y en las áreas más tiernas de la planta de la papa, atacan a sus hospederos en cualquier etapa de crecimiento, los pulgones perforan los tejidos vegetales para obtener la savia de las plantas, siendo un daño indirecto en el desarrollo de enfermedades por transmisión de virus [43].



Figura 17. Plaga pulgones [43]

3.9.8.1. Tratamiento:

Tabla 6. Tratamiento plaga Pulgones [44]

Fungicida	Modos de acción	Aplicación	Dosis
APTA	Contacto	Aspersión, realizar dos aplicaciones al follaje a intervalos de 7 días, volumen de aplicación 335 – 435 L de agua / Ha, intervalo de seguridad 1 día.	2 – 3 kg por hectárea
BENEVIA	Contacto Ingestión	Aspersión, intervalo de seguridad 1 día.	2 – 3 kg por hectárea

3.10. ENFERMEDADES EN LA HOJA DE PAPA

3.10.1. Roña

Ocasionado por el microorganismo *Spongospora subterranea*, la sarna polvorienta de la papa constituye una enfermedad significativa en la producción de papa, este patógeno se manifiesta en las etapas iniciales de crecimiento, desde la siembra hasta la floración, enfocándose en la raíz como órgano primario afectado, su impacto se concentra esencialmente en los tubérculos, afectando la calidad estética de los mismos [43].

3.10.1.1. Tratamiento:

Tabla 7. Tratamiento enfermedad la Roña [44]

Fungicida	Modos de acción	Aplicación	Dosis
ANTRACOL	Preventivo Multisitio	Rociar a 40 cm del fondo del surco	2kg – 5kg por hectárea

3.10.2. Tizón tardío

Causado por el hongo *Phytophthora infestans*, es una enfermedad fungosa perjudicial para el cultivo de la papa, la etapa de ataque se da en los primeros estados del desarrollo, los síntomas aparecen en las hojas inferiores, con pequeñas manchas de color verde oscuro con un borde amarillo alrededor de la lesión, si la enfermedad alcanza el tallo lo quebrará con facilidad y en el tubérculo la piel se torna de una coloración rojiza [43].



Figura 18. Enfermedad tizón tardío [43]

Es la enfermedad más importante del cultivo de la papa en el Ecuador por la agresividad del patógeno y las condiciones climáticas favorables para el patógeno, las pérdidas pueden llegar al 100%, el manejo de la enfermedad depende a gran medida del control químico [46].

Según [46] Los fungicidas azoxistrobina, dimetomorf y fosfonato potásico fueron los más eficientes para el control del tizón tardío en Pichincha y Carchi, iniciar el control con dimetomorf permitirá reducir el máximo de las primeras infecciones del patógeno por sus altos niveles de eficacia y por no tener riesgo de resistencia.

Tabla 8. Tratamiento enfermedad Tizón Tardío [44]

Fungicida	Modos de acción	Aplicación	Dosis
ALLIETE	Sistémico	Aspersión por un intervalo de seguridad de 14 días	2.5kg por hectárea
ANTRACOL	Preventivo Multisitio	Aspersión por un intervalo de seguridad de 14 días	2kg – 2.5kg por hectárea

3.10.3. OÍDIO

Es un hongo parásito capaz de penetrar en las células epidérmicas de su huésped, la etapa de ataque se produce en etapas avanzadas de crecimiento de la planta, los síntomas se observan con pequeñas manchas de color blanco, de aspecto pulverulento, a medida que la enfermedad progresa, hojas tienen una tonalidad amarillenta, la afección se extiende a los tallos de la planta, culminando con la muerte de la planta [43].



Figura 19. Enfermedad OÍDIO [43]

3.10.3.1. **Tratamiento:**

Tabla 9. Tratamiento enfermedad OIDIO [44]

Fungicida	Modos de acción	Aplicación	Dosis
TEBUCUR	Sistémico Contacto Preventivo	Aspersión por un intervalo de seguridad de 7 días, volumen de aplicación 678 L de agua/ha	2 – 3 kg por hectárea
Terraguard	Sistemático Curativo Erradicante	Aspersión	2 – 3 kg por hectárea

3.10.4. Punta morada

Conocida como PMP o punta morada de la papa, es uno de los principales desafíos en los cultivos de papa, su etapa de ataque desde el inicio afecta severamente a la planta, causando su muerte, las plantas afectadas muestran un decrecimiento anómalo, enrollamiento de las hojas superiores con un cambio de coloración a tonos amarillos o morados, esta enfermedad es ocasionada por fitoplasmas [43].



Figura 20. Enfermedad punta morada [43]

3.10.5. Tizón temprano

Es una enfermedad que se asocia a los tejidos envejecidos, puede contagiar las hojas en cualquier etapa de desarrollo de la planta, las lesiones son de tamaño pequeño, que pueden pasar desapercibidas a simple vista, esta enfermedad puede infectar tallos y tubérculos, este hongo se distingue por afectar la totalidad de la parte aérea de la planta en cada una de las etapas de

desarrollo del cultivo, la pérdida ocasionada por la enfermedad es más profunda cuando se manifiesta después del periodo de floración o durante la tuberización [43].



Figura 21. Enfermedad tizón temprano [43]

3.10.5.1. Tratamiento

Tabla 10. Tratamiento enfermedad punta morada [44]

Fungicida	Modos de acción	Aplicación	Dosis
CUPRAVIT	Contacto	Aspersión	2kg – 4kg por hectárea

Se pueden utilizar fungicidas a base de azoxistrobina, piraclostrobina, difenoconazol, bocalida, clorotalonil, fenamidona o combinaciones de estos, hay que tener en cuenta las condiciones meteorológicas y el intervalo de precosecha [47].

3.10.6. Roya

Es una especie que requiere la presencia de una planta huésped para llevar a cabo sus fases del ciclo biológico, en los residuos de la cosecha, se generan teliosoros que producen teleutosporas, y que a través de la meiosis dan lugar a las basidiosporas, las mismas que infectan las hojas de la planta huésped, las plantas afectadas pueden permanecer débiles y ser objeto de infección de plagas o enfermedades originados por fitopatógenos [43].

3.10.7. Costra negra

La costra negra se propaga a través del suelo y se desplaza a la siguiente temporada de crecimiento en forma de esclerocios, que se presentan tanto en los tubérculos, como en el suelo, los síntomas se presentan con pequeñas lesiones en los brotes de la papa, cuando las semillas están contaminadas con el hongo rizoctonia, este se desarrolla desde la superficie de la semilla

y la infección inicia en los primeros estados de la raíz, el estolón y las hojas, la mayor parte de las infecciones tienden a ocurrir en la fase del crecimiento de la papa [43].



Figura 22. Enfermedad costra negra [43]

3.10.7.1. Tratamiento:

Tabla 11. Tratamiento enfermedad costra negra [44]

Fungicida	Modos de acción	Aplicación	Dosis
FONTELIS	Contacto Translaminar	Aspersión al fondo del surco, realizar una única aplicación sobre los tubérculos al momento de la siembra, volumen de aplicación 300L/ha.	2 – 3 kg por hectárea
Vigold	Sistémico Curativo	Aspersión al fondo del surco, al momento de la siembra mojando el suelo y tubérculo, posteriormente tapar la semilla, intervalo de seguridad 7 días.	2 – 3 kg por hectárea

3.10.8. Pie negro y pudrición blanda

Se distingue por la descomposición del tejido parenquimatoso, culminando en una pudrición húmeda y granulosa de tonalidad blanca, acompañada de un mal olor, se manifiesta en cualquier fase de crecimiento de la planta, ya sea durante la cosecha o el almacenamiento, la bacteria

tiene la capacidad de ingresar al interior del tubérculo mediante las heridas que se generan durante la cosecha [43].



Figura 23. Enfermedad pie negro y pudrición blanda [43]

3.10.8.1. Tratamiento:

Tabla 12. Tratamiento enfermedad pie negro y pudrición blanda [44]

Fungicida	Modos de acción	Aplicación	Dosis
Kasumin	Sistémico Translaminar	Aspersión, intervalo de seguridad 14 días.	2.5 kg por hectárea
Agry-Gent Plus	Sistémico Curativo Preventivo	Aspersión, intervalo de seguridad 28 días.	2 – 3 kg por hectárea

4. METODOLOGÍA

4.1. Investigación Tecnológica

Técnica que permite al investigador obtener información mediante la implementación de una propuesta ya establecida, con el objetivo de optimizar procesos y adaptarlos, es importante que los instrumentos de investigación sean los mismos y permitan una recolección de datos eficaz.

De esta manera se adaptará y mejorará la precisión de la identificación automática de plagas y enfermedades con el uso de varias arquitecturas preentrenadas que permitan el aprendizaje y clasificación correcta.



4.2. Investigación De Campo

Es un enfoque que conlleva la recolección de datos directamente en el lugar donde los fenómenos de interés se manifiestan, los investigadores se desplazan al entorno natural o lugar específico donde se desarrolla el objeto de estudio para obtener la información, permitiendo una inmersión directa con el contexto de estudio.

Utilizamos este método con el fin de identificar plagas o enfermedades específicamente en las hojas de las plantas de papa, utilizando imágenes para llevar a cabo este proceso de reconocimiento.

4.3. Investigación Aplicada

Tiene como objetivo crear conocimiento que pueda ser directamente utilizado para abordar desafíos que enfrenta la sociedad o el ámbito productivo, busca facilitar la conexión entre la teoría y la creación de productos concretos.

El desarrollo del prototipo móvil vinculará la conexión entre el agricultor con la tecnología, siendo un apoyo en la detección de plagas o enfermedades en los cultivos de papa en la ciudad de Latacunga con técnicas de inteligencia artificial.

4.4. Métodos de investigación

4.4.1. Método hipotético deductivo

Evalúa la veracidad de las afirmaciones hipotéticas durante la implementación del proyecto de investigación, al incorporar el modelo de entrenamiento se determina si las afirmaciones son verdaderas o falsas, mediante la observación de la identificación o no de la enfermedad o plaga.

4.4.2. Método inductivo

Implica comenzar con afirmaciones generales sobre la identificación automática de plagas y enfermedades en el cultivo de papa mediante técnicas de inteligencia artificial, y en base a la implementación de prototipo realizar las observaciones para determinar si las afirmaciones cumplen o no en la práctica.

Por medio del uso del prototipo móvil comprobamos el funcionamiento del modelo implementado, realizando pruebas con fotos que no fueron entrenadas dentro del dataset y con imágenes con diferente nivel de luminosidad y calidad.



4.4.3. Método deductivo

El análisis de los resultados obtenidos durante la implementación de la aplicación para identificar el tipo de enfermedad o plaga y en base a ello analizar la eficacia del modelo entrenado.

En base a los resultados obtenidos con la implementación de nuestro modelo, comparamos los resultados del nivel de precisión con los modelos investigados que ya tiene su entrenamiento propio con transfer learning.

4.5. Técnica del cálculo de estimación del costo del software

4.5.1. COCOMO

Para la estimación del coste y esfuerzo del desarrollo del software, se empleó el modelo de estimación COCOMO, que proporcionó un marco estructurado para estimar el esfuerzo, tiempo y costos relacionados con el desarrollo del proyecto.

4.6. Herramienta para realizar pruebas del aplicativo

4.6.1. Sauce Labs

Con esta plataforma se comprobó el prototipo de software, permitiendo ejecutar pruebas manuales por medio del apk, que evaluó la interfaz gráfica con su funcionamiento, facilitando la detección de errores y garantía de calidad del software, esto se encuentra solventado en el documento que se muestra en el anexo 8.9.

4.7. Instrumentos de Investigación

Se estableció diversos parámetros e instrumentos como parte integral sobre plantaciones de papa, así como identificar y analizar las plagas y enfermedades que afectan el cultivo.

4.7.1. Entrevista

Este instrumento de investigación se aplicó al Ingeniero Agrónomo, Ing. Gil Washington Ochoa Oña, el cual está formado por 9 preguntas, diseñadas para evaluar la viabilidad del prototipo de software destinado a la detección automática de plagas y enfermedades en el cultivo de papa, el instrumento se encuentra en la sección de Anexos.



4.7.2. Ficha bibliográfica

En el desarrollo de la investigación se empleó la ficha bibliográfica como un instrumento para documentar las fuentes indagadas. Este proceso permitió optar por proyectos de investigación similares que guiaron a la elección de la técnica de procesamiento de imágenes utilizada en el entrenamiento de la red neuronal, dicho instrumento se encuentra en la sección de Anexos.

4.8. Metodología usada para los modelos de inteligencia artificial

4.8.1. Metodología KDD

Metodología diseñada para extraer conocimiento de manera automatizada a partir de grandes cantidades de datos, es un proceso iterativo, es decir que puede aplicarse las veces necesarias hasta obtener la información deseada, el conocimiento que se busca debe ser implícito, no trivial, previamente desconocido y poseer utilidad.

4.8.1.1. Fase de Recopilación de datos

Para el entrenamiento del modelo se recopilan los datos provenientes de distintas fuentes, como bases de datos, conjuntos de datos existentes como: PlantVillage, Kaggle, que serán específicos para la resolución del problema a tratar.

4.8.1.2. Fase de Preprocesamiento de datos

Los datos recopilados deben ser tratados, es decir aplicar un proceso de limpieza y preparación de la data, se efectúa este proceso para controlar valores atípicos, tamaño de imágenes, número de épocas, etc.

4.8.1.3. Fase de Transformación de datos

Los datos pre procesados deben tener un formato adecuado para el posterior entrenamiento del modelo, es así que se aplica un proceso de normalización para seleccionar las características importantes.

4.8.1.4. Fase de Minería de datos

En esta fase se aplican algoritmos de aprendizaje automático de minería de datos para descubrir los patrones y relaciones en los datos, implica el proceso de entrenar y ajustar el modelo para que pueda realizar tareas específicas como la clasificación, regresión o agrupamiento.



4.8.1.5. Fase de Evaluación de patrones

Verifica que los supuestos de validación siempre se cumplan y se lo puede hacer son la validación cruzada, hay que tener en cuenta que si la fase de validación no es satisfactoria se deberá replantear el problema e intentar con un algoritmo distinto o su vez cambiar los valores en las fases anteriores e ir comprobando el funcionamiento.

4.9. Metodología de desarrollo de software

4.9.1. Mobile-D

Es una metodología ágil diseñada específicamente para el desarrollo de aplicaciones móviles, se caracteriza por la interacción que se tiene con el cliente en el desarrollo del aplicativo, así como la adaptabilidad a los cambios que se efectúen durante el desarrollo, esta metodología se destaca por su eficacia y eficiencia, especialmente en lo que concierne a la reducción del tiempo de producción[48].

4.9.1.1. Fase de exploración

En esta primera fase se define el alcance del proyecto en conjunto con las funcionalidades que se pretenden alcanzar, el enfoque se basa en la elaboración del plan del proyecto con la comprensión de los conceptos fundamentales.

4.9.1.2. Fase de iniciación

En esta fase se organiza el proyecto mediante la identificación y preparación de los recursos requeridos, durante esta fase se destina un día para la planificación y el tiempo restante se emplea en la ejecución y publicación de las tareas.

4.9.1.3. Fase de producto

Durante esta etapa, se realiza la ejecución completa de la implementación, además se sigue la metodología TDD de desarrollo dirigido por pruebas, que implica que antes de comenzar la implementación de una funcionalidad, se debe crear una prueba que valide el funcionamiento.

4.9.1.4. Fase de estabilización

En esta etapa se realizan las últimas acciones de integración con el fin de garantizar el correcto funcionamiento del sistema en su totalidad.

4.9.1.5. Fase de pruebas

En la fase de pruebas se busca asegurar la estabilidad y el correcto funcionamiento de la aplicación para que los clientes puedan utilizarla según los requisitos establecidos por el cliente, corrigiendo cualquier error identificado durante este proceso.

4.10. Características de hardware y software

Tabla 13. Herramientas utilizadas para la aplicación móvil

Herramientas	Descripción	Versión
Lenguaje de programación Java	El lenguaje de programación Java permitió el diseño y de desarrollo de la aplicación móvil.	Java 8
Android Studio	Es un entorno de desarrollo integrado (IDE), nos permitió desarrollar la aplicación móvil y realizar pruebas de su funcionamiento utilizando un emulador.	Android Studio Giraffe 2022.3.1

*Elaborado por: Los investigadores

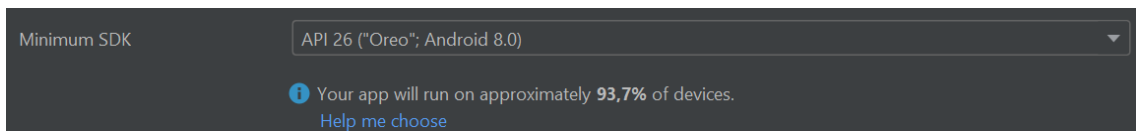


Figura 24. Versión de Android utilizado

Google Colab es un entorno colaborativo de Google que nos proporcionó el acceso a recursos computacionales gratuitos, así como ejecutar código de Python en el navegador.

Tabla 14. Herramientas utilizadas para los modelos

Herramientas	Descripción	Versión
Lenguaje de programación Python	El lenguaje de programación Python facilitó en la creación y entrenamiento del modelo gracias a su variedad de bibliotecas.	3.10.12

*Elaborado por: Los investigadores

4.10.1. Características de las computadoras

La computadora N°1 en donde se destaca la capacidad de la tarjeta gráfica NVIDIA GeForce GTX 1650 para manejar tareas gráficas exigentes debido a que posibilita acelerar el proceso de



entrenamiento de modelos de aprendizaje automático, la tabla de características se encuentra en los Anexos.

La computadora N°2, que a pesar de la ausencia de una tarjeta gráfica dedicada, este ordenador se distingue por su procesador AMD Ryzen 7 de la serie 3700U, el cual ofrece un notable rendimiento sin una tarjeta gráfica independiente.

5. ANÁLISIS DE RESULTADOS

El desarrollo de este proyecto se realizó en Google Colab, una plataforma en línea que proporciona recursos de computación, incluyendo acceso a GPUs y TPUs para desarrollar y ejecutar código de forma colaborativa en notebooks de Jupyter utilizando el lenguaje de programación Python.

5.1. Metodología KDD

5.1.1. Etapa de Selección

Para la recopilación de datos se utilizó el repositorio PlantVillage, una base de datos abierta que proporcionó 2152 imágenes referentes a las hojas de papa en fotografías a color, las cuales se dividen por el tipo de enfermedad, obteniendo así 150 hojas sanas, 1000 hojas con tizón tardío y 1000 hojas con tizón temprano.

Además, se recurrió a la base de datos Kaggle en donde se encontró un dataset que contenía 38 imágenes a color correspondientes a la enfermedad de la pulguilla de la papa.

Por otra parte, se obtuvo un conjunto de datos propio que incluyó hojas de papa en diferentes condiciones de crecimiento. Entre los datos recopilados se encontraron plantas con un mes de edad, fotografiadas en entornos soleados a las 15:00 horas con un nivel de luminosidad alto. Además, se capturaron imágenes de plantas con un mes y dos semanas de edad en condiciones con escasa luminosidad solar a las 17: 00 horas con un nivel bajo de luminosidad.

Se utilizó una combinación de imágenes obtenidas de diferentes fuentes de datos para entrenar nuestro modelo, este conjunto de datos incluye 38 imágenes de hojas con pulguilla del repositorio Kaggle, del repositorio PlantVillage con un total de 2152 imágenes, las mismas que se dividen en : hojas sanas 152 imágenes, hoja con tizón tardío 1000 imágenes, hoja con tizón temprano 1000 imágenes, además incorporamos 311 imágenes de nuestro propio dataset, distribuidas de la siguiente manera: 153 imágenes sanas, 68 imágenes con tizón temprano, 48 imágenes con tizón tardío y 42 imágenes con pulguilla.



Para el entrenamiento de este modelo, hemos empleado las siguientes clases: hoja con pulguilla, hoja con tizón temprano, hoja con tizón tardío, y hoja sana.

La presente tabla describe los conjuntos de datos utilizados para el entrenamiento de los modelos de clasificación de imágenes.

Tabla 15. Conjuntos de datos para el entrenamiento de los modelos

Conjunto de Datos			
Fuente de Datos	Total de imágenes	Clases	Total
PlantVillage	2152	Hoja con tizón temprano	1000
		Hoja con tizón tardío	1000
		Hoja sana	152
Kaggle	38	Hoja con pulguilla	38
Dataset propio	311	Hoja con tizón temprano	68
		Hoja con tizón tardío	48
		Hoja sana	153
		Hoja con pulguilla	42

*Elaborado por: Los investigadores

5.1.2. Etapa de preprocesamiento de datos

Las imágenes del conjunto de datos fueron redimensionadas a un tamaño de 224 x 224 píxeles y se utilizaron tres canales RGB. Este estándar fue esencial para asegurar la compatibilidad de entrada con las distintas arquitecturas convolucionales, tales como ResNet50, VGG, Inception, EfficientNetB0 y MobileNet.

Se implementó un script en Python utilizando la biblioteca PIL (Python Imaging Library) para llevar a cabo este proceso de redimensionamiento. El código recorrió el directorio especificado para las imágenes, seleccionando archivos con extensiones .jpg, .jpeg o .png y redimensionando solo aquellas imágenes que originalmente tenían un tamaño de 256 x 256 píxeles.

El bloque de código empleado para este propósito se presenta a continuación.

```
from PIL import Image
import os

# Directorio donde se encuentran tus imágenes
images_directory = '/content/DataSetInternet'

# Redimensionar solo las imágenes de formatos .jpg, .jpeg o .png y tamaño 256x256 a 224x224
for root, dirs, files in os.walk(images_directory):
    for file in files:
        file_path = os.path.join(root, file)
        if file_path.endswith(".jpg") or file_path.endswith(".jpeg") or file_path.endswith(".png"):
            img = Image.open(file_path)
            if img.size == (256, 256): # Verificar si la imagen es de tamaño 256x256
                img = img.resize((224, 224))
                img.save(file_path)
```

Figura 25. Código para redimensionar las imágenes a 224 x 224

Sin embargo, cada arquitectura tuvo diferentes procesos de preprocesamiento de datos, por ejemplo:

5.1.2.1. ResNet50

En el preprocesamiento, se aplicará la reducción de la resolución de todas las imágenes, proceso necesario para que las imágenes tengan la misma resolución, además, se implementará el aumento del número de imágenes en el conjunto de datos de entrenamiento mediante técnicas como la rotación, desplazamiento de anchura y altura.

5.1.2.2. VGG19

En este modelo se aplicará el redimensionamiento de las imágenes con una resolución de 224x224 píxeles para que coincida con el formato de entrada esperado y se aplicará la normalización de píxeles en un rango entre 0 y 1 como parte del modelo.

5.1.2.3. Inception NetV3

Para esta arquitectura cada imagen del conjunto de datos se aplicará el redimensionamiento de una resolución de 299x299, además se deberá normalizar el valor de cada pixel entre los rangos de 0 y 1, estos dos pasos de pre procesamiento son importantes para su posterior entrenamiento del modelo.

5.1.2.4. EfficientNetB0

Durante la fase de preprocesamiento para esta arquitectura se aplicará una serie de transformaciones como la técnica de aumento de datos que incluirá operaciones horizontales,



verticales y rotaciones a 90°. Estas transformaciones permitirán generar variaciones en las imágenes de entrenamiento, enriqueciendo así el conjunto de datos con distintas muestras.

5.1.2.5. MobileNet

Para esta arquitectura, se llevó a cabo un conjunto de transformaciones como el filtrado gaussiano, destinado a suavizar y reducir el ruido presente en las imágenes, lo que ayudó a mejorar la calidad y la uniformidad de los datos.

Posterior al filtrado gaussiano, se implementó un proceso de normalización para establecer una escala común entre 0 y 1 en los valores pixeles.

5.1.2.6. Modelo propuesto

En la arquitectura propuesta utilizamos el redimensionamiento de la imagen de 256 pixeles, que es un tamaño aceptable para el proceso de entrenamiento.

5.1.3. Etapa de Transformación de datos

En esta etapa se llevó a cabo una partición del dataset para adaptarlo al proceso de entrenamiento, validación y pruebas requerido por todos los modelos empleados. Por lo que se aplicó una división estándar en la que conjunto de datos se dividió en tres subconjuntos: entrenamiento un 80%, para validación, un 10%, y para pruebas otro 10%.

El siguiente código implementa una función en Python que se encarga del particionamiento del dataset. Esta función facilita la división del conjunto de datos en porciones específicas para entrenamiento, validación y pruebas.

```
# Particionamiento del dataset y preprocesamiento de datos
def get_dataset_partitions_tf(ds, train_split=0.8, val_split=0.1, test_split=0.1, shuffle=True, shuffle_size=1000):
    ds_size = len(ds)

    if shuffle:
        ds = ds.shuffle(shuffle_size, seed=12)
        train_size = int(train_split * ds_size)
        val_size = int(val_split * ds_size)

    train_ds = ds.take(train_size)
    val_ds = ds.skip(train_size).take(val_size)
    test_ds = ds.skip(train_size).skip(val_size)
    return train_ds, val_ds, test_ds
```

Figura 26. Código para particionar el dataset

5.1.4. Etapa de Minería de datos

Durante esta etapa, se utilizará modelos preentrenados los mismos que se personalizarán para adaptarlos a la necesidad del caso de estudio. Este proceso incluirá la importación y configuración de bibliotecas correspondientes a cada uno de los distintos modelos utilizados.

A continuación, se detallará el procedimiento que se ejecutará para cada modelo específico.

5.1.4.1. ResNet50

Importación de la biblioteca ResNet50 preentrenada para su posterior personalización.

```
[ ] import tensorflow as tf
    from tensorflow.keras import layers
    from tensorflow.keras.applications import ResNet50
    from tensorflow.keras.layers.experimental import preprocessing
```

Figura 27. Importación de la biblioteca del modelo ResNet50

Se utilizará el modelo de red neuronal de ResNet50 preentrenado utilizando TensorFlow.

```
# Modelo ResNet50 pre-entrenado
base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(IMAGE_SIZE, IMAGE_SIZE, CHANNELS))

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50\_weights\_tf\_dim\_ordering\_tf\_kernels\_notop.h5
94765736/94765736 [=====] - 1s 0us/step
```

Figura 28. Modelo preentrenado ResNet50.

Se construyó el modelo que consta de una serie de capas que incluirá la red ResNet50 preentrenada como base, además se incluirá las capas densas, capa de dropout y la capa final que produce las predicciones.

```
[ ] # Construir el modelo
    model = tf.keras.Sequential([
        resize_and_rescale,
        base_model,
        global_average_layer,
        layers.Dense(256, activation='relu'),
        layers.Dropout(0.5),
        prediction_layer
    ])
```

Figura 29. Construcción del modelo.

Se empleará el modelo para entrenar, con varios parámetros como `train_ds` que se proporcionará al modelo para entrenar los patrones, con el número de épocas en el caso de ResNet50 con 25 epochs, el tamaño del lote, mensajes de progreso y la validación del conjunto de datos que se programará para evaluar el rendimiento del modelo de cada época.

```
[ ] # Entrenar el modelo
    history = model.fit(
        train_ds,
        epochs=EPOCHS,
        batch_size=BATCH_SIZE,
        verbose=1,
        validation_data=val_ds
    )
```

Figura 30. Entrenar el modelo ResNet50

5.1.4.2. VGG19

Se importará la biblioteca de la arquitectura preentrenada VGG19, además se importarán los módulos layers, models y preprocessing de la biblioteca Keras para construir y entrenar el modelo.

```
import tensorflow as tf
from tensorflow.keras.applications import VGG19
from tensorflow.keras import layers, models, preprocessing
```

Figura 31. Importación de la biblioteca VGG19

Se creará un modelo utilizando la arquitectura preentrenada VGG19 de TensorFlowKeras, indicando que se deberá utilizar los pesos preentrenados en el conjunto de datos, además se indicará que no se deben incluir las capas totalmente porque las capas finales deben ir asociadas a nuestra especificación.

```
# Crear el modelo VGG19 preentrenado
base_model = VGG19(weights='imagenet', include_top=False, input_shape=(IMAGE_SIZE, IMAGE_SIZE, 3))

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg19/vgg19\_weights\_tf\_dim\_ordering\_tf\_kernels\_notop.h5
80134624/80134624 [=====] - 0s 0us/step
```

Figura 32. Modelo preentrenado VGG19

Se construirá un modelo secuencial, en base al modelo preentrenado, agregando un promedio global para calcular cada característica, también agregamos una capa con 256 neuronas para que tenga una capacidad adicional en el aprendizaje ya que aprende patrones más específicos.

```
# Construir el modelo
model = tf.keras.Sequential([
    base_model,
    global_average_layer,
    layers.Dense(256, activation='relu'),
    layers.Dropout(0.5),
    prediction_layer
])
```

Figura 33. Construcción del modelo VGG19

El entrenamiento del modelo de la red neuronal se va ir almacenando en el objeto history en donde se evaluará el conjunto de datos de validación después de cada época, la velocidad del entrenamiento dependerá del tipo de computadora que se disponga.

```
# Entrenar el modelo
history = model.fit(
    train_ds,
    epochs=EPOCHS,
    batch_size=BATCH_SIZE,
    verbose=1,
    validation_data=val_ds
)
```

Figura 34. Entrenamiento del modelo VGG19

5.1.4.3. Inception

Se importará la biblioteca del modelo preentrenado InceptionV3 para el uso en la construcción de modelos de aprendizaje profundo.

```
import tensorflow as tf
from tensorflow.keras import layers
from tensorflow.keras.applications import InceptionV3
from tensorflow.keras.layers.experimental import preprocessing
```

Figura 35. Importación de la biblioteca del modelo InceptionV3

Se instanciará un modelo utilizando la arquitectura InceptionV3 de TensorFlowKeras, especificando los pesos preentrenados en el conjunto de datos que deben ser utilizados.

```
#creación de base_model con InceptionV3
base_model = InceptionV3(weights='imagenet', include_top=False, input_shape=(IMAGE_SIZE, IMAGE_SIZE, CHANNELS))
```

Figura 36. Modelo preentrenado InceptionV3

Se construirá el modelo compuesto por diversas capas, entre las cuales se encuentran la red InceptionV3 preentrenada como base, se ha incorporarán capas densas, una capa de dropout y la capa final encargada de generar las predicciones.

```
model = tf.keras.Sequential([
    resize_and_rescale,
    base_model,
    global_average_layer,
    layers.Dense(256, activation='relu'),
    layers.Dropout(0.5),
    prediction_layer
])
```

Figura 37. Contrucción del modelo InceptionV3

En el objeto history se realizará la evaluación del conjunto de datos de validación después de cada época, algo que recalcar en este modelo es el número de épocas, ya que contemplará la mayor cantidad de épocas, obteniendo mayor oportunidad de ajustar sus pesos para aprender de manera más completa, esto conllevará un tiempo de entrenamiento más prolongado debido a la mayor cantidad de iteraciones en el conjunto de datos de entrenamiento.

```
# Entrenar el modelo
history = model.fit(
    train_ds,
    epochs=EPOCHS,
    batch_size=BATCH_SIZE,
    verbose=1,
    validation_data=val_ds
)
```

Figura 38. Entrenamiento del modelo InceptionV3

5.1.4.4. EfficienNetB0

Importación de la biblioteca EfficientNetB0, a través de TensorFlow y Keras permitió la exploración y configuración de sus capas y métodos.

```
import tensorflow as tf
from tensorflow.keras import layers
from tensorflow.keras.applications import EfficientNetB0
from tensorflow.keras.layers.experimental import preprocessing
```

Figura 39. Importación de la biblioteca del modelo EfficientNetB0

Se empleará el modelo EfficientNetB0 especificando los pesos preentrenados en el conjunto de datos que deben ser utilizados.

```
# Modelo EfficientNetB0 pre-entrenado con fine-tuning
base_model = EfficientNetB0(weights='imagenet', include_top=False, input_shape=(IMAGE_SIZE, IMAGE_SIZE, CHANNELS))
```

Figura 40. Modelo preentrenado EfficientNetB0

Este bloque de código define un modelo que combina capas de preprocesamiento, un modelo preentrenado, capas de preprocesamiento de características, capas de densidad y la capa final para generar las predicciones.

```
# Construir el modelo
model = tf.keras.Sequential([
    resize_and_rescale,
    base_model,
    global_average_layer,
    layers.Dense(256, activation='relu'),
    layers.Dropout(0.5),
    prediction_layer
])
```

Figura 41. Contrucción del modelo EfficientNetB0

Este bloque de código realiza el entrenamiento del modelo, ajustando sus pesos y parámetros internos durante múltiples épocas para mejorar su capacidad predictiva y monitoreando su desempeño utilizando un conjunto de validación.

```
# Entrenar el modelo con más épocas
history = model.fit(
    train_ds,
    epochs=EPOCHS,
    batch_size=BATCH_SIZE,
    verbose=1,
    validation_data=val_ds
)
```

Figura 42. Entrenamiento del modelo EfficientNetB0

5.1.4.5. MobileNet

Se importó la biblioteca MobileNet, a través de TensorFlow y Keras permitió la exploración y configuración de sus capas y métodos.

```
import tensorflow as tf
from tensorflow.keras import layers
from tensorflow.keras.applications import MobileNet
from tensorflow.keras.layers.experimental import preprocessing
```

Figura 43. Importación de la biblioteca del modelo MobileNet

Se emplea el modelo MobileNet especificando los pesos preentrenados en el conjunto de datos que deben ser utilizados.

```
# Modelo MobileNetV2 pre-entrenado
base_model = MobileNetV2(weights='imagenet', include_top=False, input_shape=(IMAGE_SIZE, IMAGE_SIZE, CHANNELS))
```

Figura 44. Modelo preentrenado MobileNet

Este bloque de código definirá un modelo que combina capas de preprocesamiento, un modelo preentrenado, capas de preprocesamiento de características, capas de densidad y la capa final para generar las predicciones

```
# Construir el modelo
model = tf.keras.Sequential([
    resize_and_rescale,
    base_model,
    global_average_layer,
    layers.Dense(256, activation='relu'),
    layers.Dropout(0.5),
    prediction_layer
])
```

Figura 45. Construcción del modelo MobileNet

Este bloque de código realizará el entrenamiento del modelo, ajustando sus pesos y parámetros internos durante múltiples épocas para mejorar su capacidad predictiva y monitoreando su desempeño utilizando un conjunto de validación.

```
# Entrenar el modelo
history = model.fit(
    train_ds,
    epochs=EPOCHS,
    batch_size=BATCH_SIZE,
    verbose=1,
    validation_data=val_ds
)
```

Figura 46. Entrenamiento del modelo MobileNet

5.1.4.6. Modelo Propuesto

Se importó la librería de Keras, que nos permitió crear el modelo utilizando su motor de ejecución TensorFlow, también importamos las bibliotecas de Matplotlib para generar gráficos de precisión y pérdida, además, empleamos IPython para la escritura interactiva de python.

```
[ ] import tensorflow as tf
    from tensorflow.keras import models, layers
    import matplotlib.pyplot as plt
    from IPython.display import HTML
```

Figura 47. Importación de las bibliotecas del modelo propuesto

Se configuró el proceso de entrenamiento del modelo utilizando el optimizador Adam, que permití ajustar los pesos del modelo durante el entrenamiento, se especificó la función de pérdida para evaluar las predicciones y etiquetas verdaderas durante el entrenamiento, por último la métrica accuracy para el calculo de la precisión del modelo.

```
[ ] model.compile(  
    optimizer='adam',  
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),  
    metrics=['accuracy']  
)
```

Figura 48. Contrucción del modelo propuesto

Se utilizó el método model.fit para entrenar el modelo en Keras, llamando las variables definidas con los valores establecidos que incluyen los datos de entrenamiento, número de épocas, tamaño del lote, información de progreso y datos de validación.

```
[ ] history = model.fit(  
    train_ds,  
    epochs=EPOCHS,  
    batch_size=BATCH_SIZE,  
    verbose=1,  
    validation_data=val_ds  
)
```

Figura 49. Entrenamiento del modelo propuesto

5.1.5. Etapa de Evaluación de patrones

Para esta etapa se presentan los resultados obtenidos en la evaluación de patrones de las distintas arquitecturas de redes neuronales convolucionales. Estos resultados se basan en el análisis de las métricas de evaluación como las gráficas de entrenamiento vs validación, entrenamiento vs pérdida y la matriz de confusión las cuales presentan el desempeño de los modelos a lo largo del proceso de entrenamiento.

A continuación, se presentan los resultados obtenidos de las arquitecturas de redes neuronales convolucionales.

5.1.5.1. ResNet50

Para evidenciar la evaluación de patrones nos valemos de gráficas.

La gráfica de entrenamiento y validación en la precisión, se puede apreciar que los valores van desde un promedio de 0.94 desde el inicio en la época 0, teniendo un incremento en 1 desde la época 5, manteniéndose en dicho valor hasta la época 25, es importante considerar que los valores de precisión deben oscilar alrededor de 1, por lo que el entrenamiento y validación en la precisión muestra un rendimiento positivo.

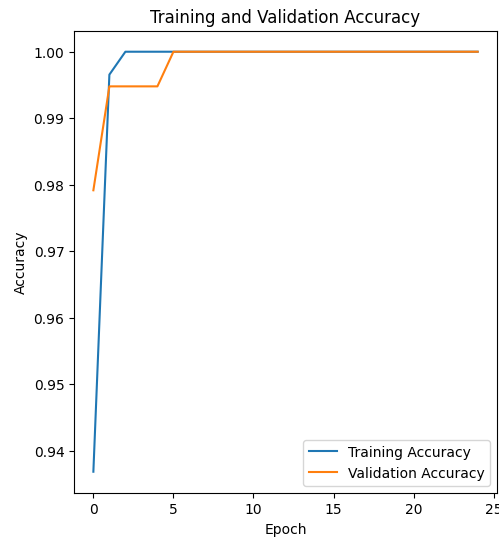


Figura 50. Gráfica de entrenamiento y validación de precisión modelo ResNet50

Gráfica de entrenamiento y validación en la pérdida, se apreciará que los valores están adecuadamente proporcionados, en la época cero, la pérdida inicia con un valor de 0.175 y a partir de la época 3 se mantiene en valores comprendidos entre 0.025 y 0.00, considerando que el valor de pérdida debería tender a 0, se interpreta que el nivel de pérdida es positivo.

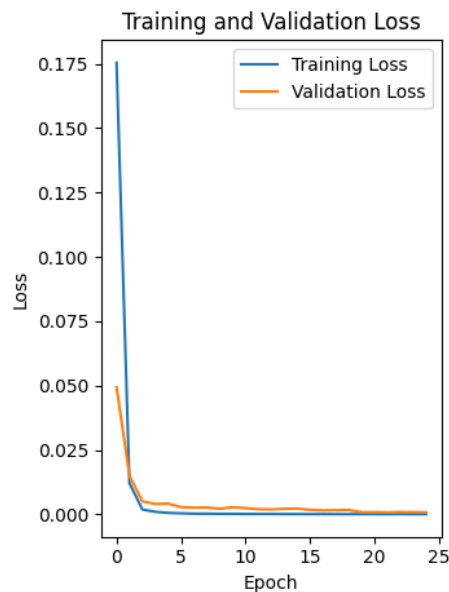


Figura 51. Gráfica de entrenamiento y validación de pérdida, modelo ResNet50

Matriz de confusión

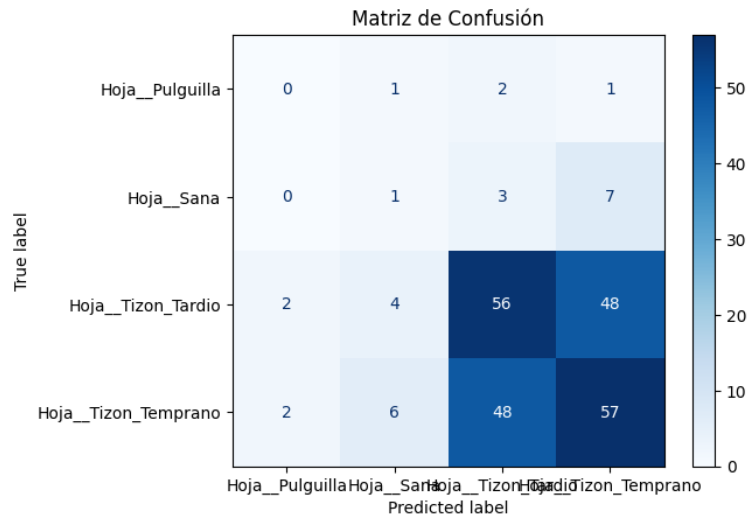


Figura 52. Matriz de confusión modelo ResNet50

5.1.5.2. VGG19

La representación gráfica de la precisión en el entrenamiento y la validación revela una tendencia distinta en cada época, desde la época 0 hasta la 5, los valores oscilan entre 0.3 a 0.5, lo cual está por debajo del rendimiento ideal ya que la precisión debería acercarse a 1, mientras que desde la época 10, se observa una mejora significativa. Con valores ascendentes desde 0.6 a 0.9, el valor más cercano a 0.9 se acerca al nivel deseado de precisión.

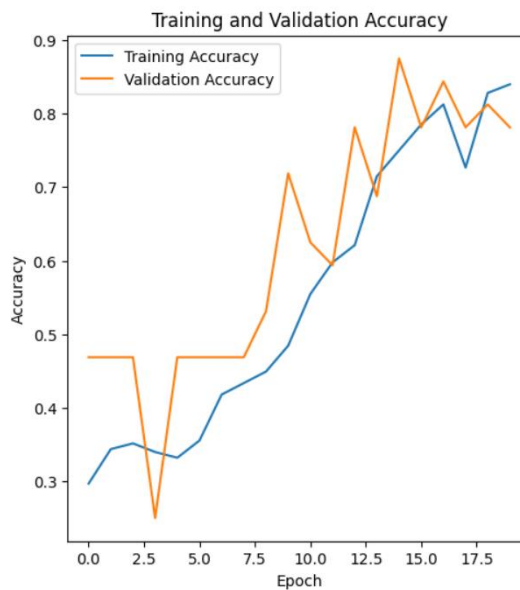


Figura 53. Gráfica de entrenamiento y validación de precisión, modelo VGG19

Gráfica de entrenamiento y validación en la pérdida, se apreciará que los valores son elevados, oscilando entre 2.5 y 1.5 desde el inicio, lo cual es significativamente alto en comparación con

el valor de referencia ideal, que debería aproximarse a 0, a partir de la época 13, se observa que los valores empezarán a bajar a un nivel de 0.5, este valor representa una tasa de pérdida aceptable en el entrenamiento y validación del modelo VGG19.

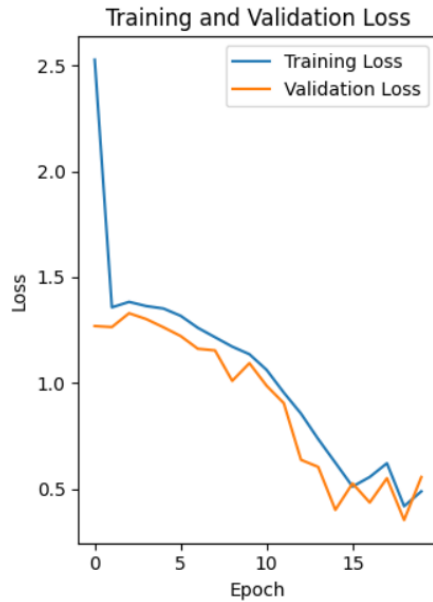


Figura 54. Gráfica de entrenamiento y validación de pérdida, modelo VGG19

Matriz de confusión

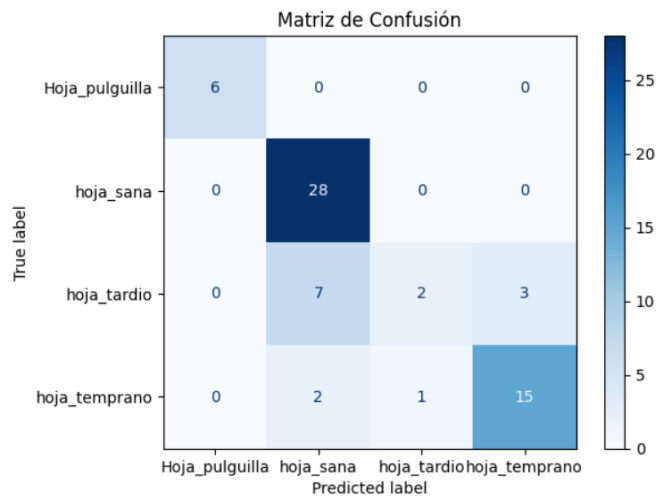


Figura 55. Matriz de confusión modelo VGG19

5.1.5.3. InceptionV3

La gráfica de entrenamiento en la precisión empezará con un valor de 0.4 y se mantiene constante hasta la época 2, de ahí en adelante se mantiene en un valor de 1.0, que es el ideal, en la gráfica de validación de entrenamiento, el valor de 0.4 persiste hasta la época 5, y luego

experimenta un aumento a partir de la época 10, alcanzando un rango entre valor de 0.6 a 0.8 hacia el final, el valor máximo de validación es 0.8, lo que indica un rendimiento aceptable y muy bueno en la tasa de precisión en promedio.

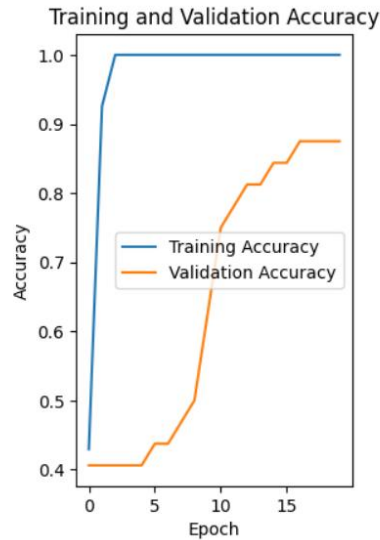


Figura 56. Gráfica en entrenamiento y validación de precisión, modelo InceptionV3

La gráfica de entrenamiento en la pérdida, empieza con un valor de 1.5 y permanecerá constante hasta la época 2, a partir de ahí se mantendrá en un valor de 0.0 hasta el final de la época, que es el ideal, en la gráfica de validación de pérdida, el valor inicial es 2.0 con un incremento a 2.5 hasta la época 5, después comienza a descender, manteniéndose entre 0.5 desde la época 10 hasta el final, estos patrones indican en promedio un rendimiento aceptable en términos de pérdida.

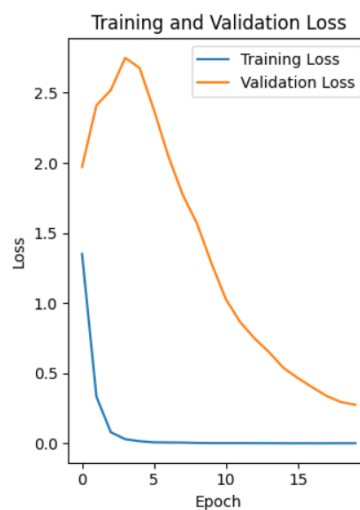


Figura 57. Gráfica de entrenamiento y validación de pérdida, modelo InceptionV3

Matriz de confusión

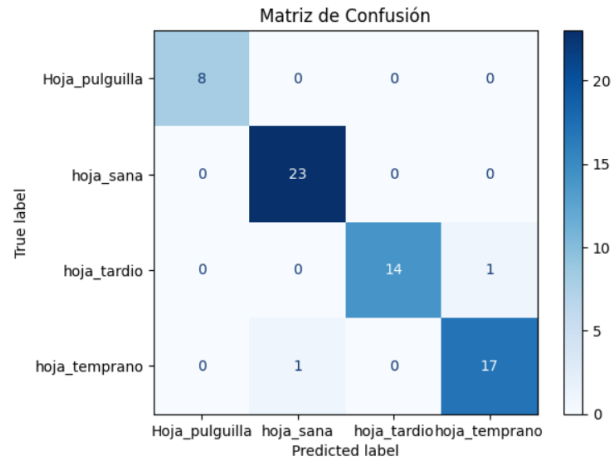


Figura 58. Matriz de confusión modelo InceptionV3

5.1.5.4. EfficienNetB0

Entrenamiento vs Validación de precisión

Se ilustra como la curva de entrenamiento y validación en general van a la par y se mantiene en la precisión cerca o igual a 1.00 lo cual es un buen desempeño, aunque en ocasiones presenta ligeros descensos.

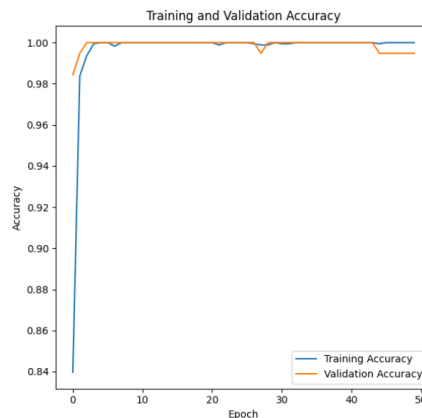


Figura 59. Gráfica de entrenamiento y validación de precisión del modelo EfficientNetB0

Entrenamiento vs validación pérdida

Se ilustra como la curva de entrenamiento de pérdida empieza desde la época cero y de cierto modo hasta la época 3 a descender desde 0.4 al valor inicial de 0.0 en donde alcanza a la

validación de pérdida, luego ambas curvas se mantienen en el valor 0.0 lo que es muy excelente desempeño.

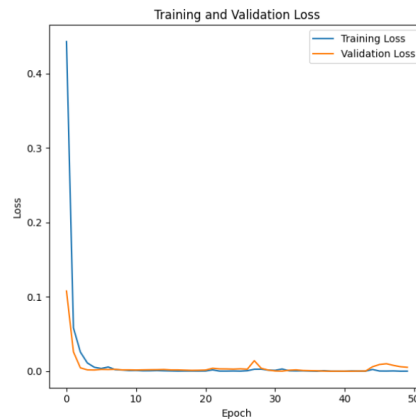


Figura 60. Gráfica de entrenamiento y validación de pérdida, modelo EfficientNetB0

Entrenamiento vs Validación de precisión

En esta gráfica se presenta como la curva de entrenamiento y validación empiezan con una precisión del 0.75 y conforme va avanzando en el entrenamiento van ascendiendo casi a al mismo nivel de precisión hasta alcanzar una puntuación de 0.90 puesto que a partir de la época 30 ambas curvas se van separando.

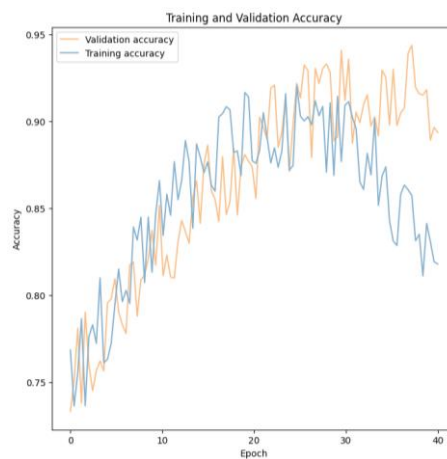


Figura 61. Gráfica de entrenamiento y validación de precisión del modelo MobileNet

Matriz de confusión

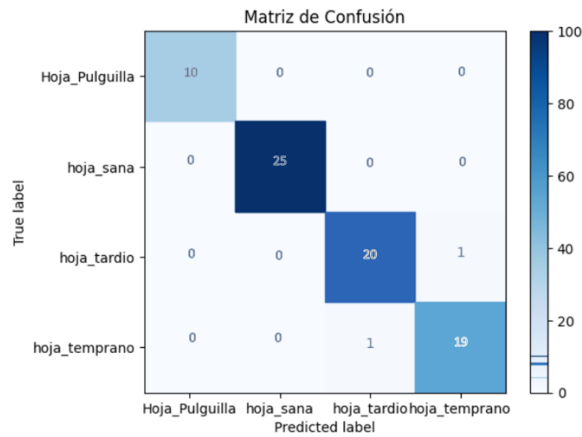


Figura 62. Matriz de confusión modelo InceptionV3

5.1.5.5. MobileNet

Entrenamiento vs validación pérdida

Se presenta el comportamiento dispar que tienen las curvas de entrenamiento y de pérdida debido a que desde la época uno la curva de entrenamiento empieza con 0.25 pero va ascendiendo hasta 0.50 mientras que la curva de validación empieza con 0.35 y durante casi todas las épocas mantiene esa puntuación, aunque en ciertos momentos desciende o asciende levemente.

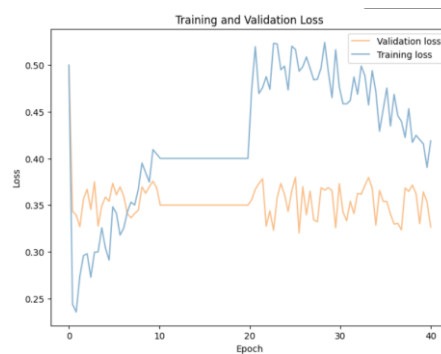


Figura 63. Gráfica de entrenamiento y validación de pérdida, modelo EfficientNetB0

Matriz de confusión

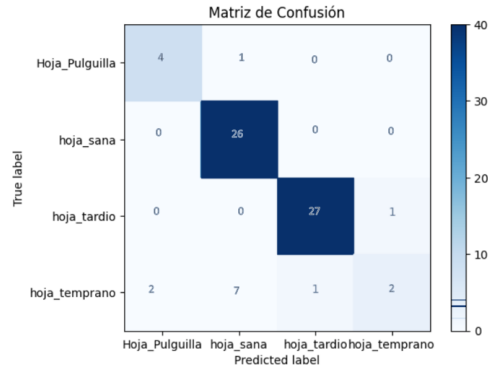


Figura 64. Matriz de confusión modelo MobileNet

5.1.5.6. Modelo propuesto

Validación de precisión vs validación de pérdida

Se observa en la gráfica de precisión que el comportamiento de los valores va desde 0.6 hasta un máximo de 0.9, manteniendo el valor desde la época 20 hasta el final, se considera un valor positivo ya que se aproxima a 1 que es el valor ideal, en la gráfica de pérdida empieza con valores altos con respecto a su valor ideal que es 0, y empieza a tener mejoría desde la época 10 con valores de 0.2, bajando hasta 0.010.

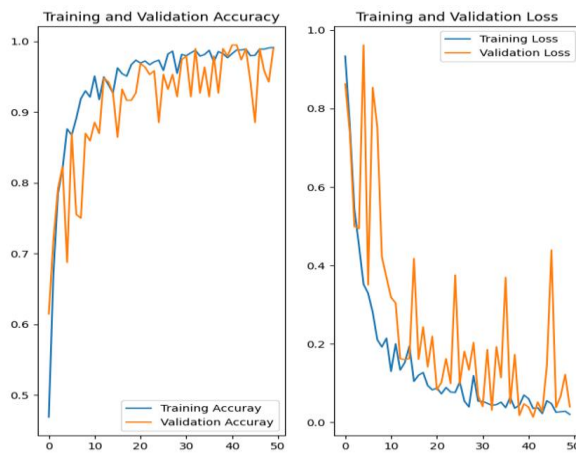


Figura 65. Gráfica de entrenamiento de precisión y pérdida modelo propuesto

Matriz de confusión

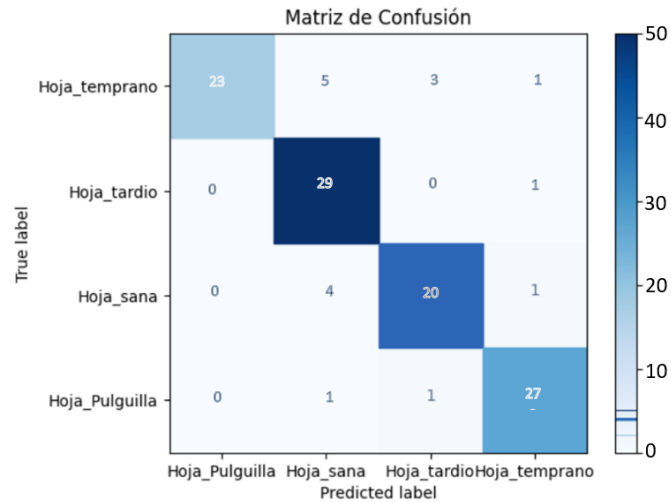


Figura 66. Matriz de confusión modelo propuesto

Para reflejar la efectividad de los resultados del entrenamiento del modelo propuesto se tabuló los datos con imágenes entrenadas en la red neuronal mediante la comparativa con un experto humano, en este caso con el Ingeniero Agrónomo, Ing. Gil Washington Ochoa Oña.

Tabla 16. Tabulación de resultados de experto humano y modelo propuesto

N°	ANÁLISIS DEL EXPERTO	AFECTACION	ESTADO DE LA HOJA	APLICACIÓN MÓVIL CON RED NEURONAL	RESULTADO COMPARATIVO
1	hoja_pulgilla	plaga	plaga	hoja_pulgilla	SI
2	hoja_sana	sano	sano	hoja_sana	SI
3	hoja_tizon_tardio	enfermedad	enfermedad	hoja_tizon_tardio	SI
4	hoja_tizon_temprano	enfermedad	enfermedad	hoja_tizon_temprano	SI
5	hoja_tizon_tardio	enfermedad	enfermedad	hoja_tizon_tardio	SI
6	hoja_tizon_temprano	enfermedad	enfermedad	hoja_tizon_tardio	NO
7	hoja_tizon_temprano	enfermedad	enfermedad	hoja_tizon_temprano	SI
8	hoja_tizon_temprano	enfermedad	enfermedad	hoja_tizon_temprano	SI
9	hoja_pulgilla	plaga	plaga	hoja_pulgilla	SI
10	hoja_tizon_tardio	enfermedad	enfermedad	hoja_tizon_tardio	SI
11	hoja_tizon_temprano	enfermedad	enfermedad	hoja_tizon_temprano	SI
12	hoja_pulgilla	plaga	plaga	hoja_pulgilla	SI
13	hoja_pulgilla	plaga	plaga	hoja_pulgilla	SI
14	hoja_sana	sano	plaga	hoja_pulgilla	NO
15	hoja_tizon_tardio	enfermedad	enfermedad	hoja_tizon_tardio	SI



16	hoja_tizon_tardio	enfermedad	enfermedad	hoja_tizon_tardio	SI
17	hoja_tizon_temprano	enfermedad	enfermedad	hoja_tizon_tardio	NO
18	hoja_tizon_temprano	enfermedad	enfermedad	hoja_tizon_temprano	SI
19	hoja_tizon_tardio	enfermedad	enfermedad	hoja_tizon_tardio	SI
20	hoja_tizon_temprano	enfermedad	enfermedad	hoja_tizon_temprano	SI
21	hoja_tizon_temprano	enfermedad	enfermedad	hoja_tizon_temprano	SI
22	hoja_tizon_temprano	enfermedad	enfermedad	hoja_tizon_temprano	SI
23	hoja_pulgilla	plaga	plaga	hoja_pulgilla	SI
24	hoja_tizon_tardio	enfermedad	enfermedad	hoja_tizon_tardio	SI
25	hoja_tizon_temprano	enfermedad	enfermedad	hoja_tizon_temprano	SI
26	hoja_pulgilla	plaga	plaga	hoja_pulgilla	SI
27	hoja_pulgilla	plaga	plaga	hoja_pulgilla	SI
28	hoja_sana	sano	sano	hoja_sana	SI
29	hoja_tizon_tardio	enfermedad	enfermedad	hoja_tizon_tardio	SI
30	hoja_tizon_tardio	enfermedad	enfermedad	hoja_tizon_tardio	SI
31	hoja_tizon_temprano	enfermedad	enfermedad	hoja_tizon_tardio	NO
32	hoja_pulgilla	plaga	plaga	hoja_pulgilla	SI
33	hoja_pulgilla	plaga	plaga	hoja_pulgilla	SI
34	hoja_sana	sano	sano	hoja_sana	SI
35	hoja_tizon_tardio	enfermedad	enfermedad	hoja_tizon_tardio	SI
36	hoja_pulgilla	plaga	plaga	hoja_pulgilla	SI
37	hoja_pulgilla	plaga	plaga	hoja_pulgilla	SI
38	hoja_sana	sano	sano	hoja_sana	SI
39	hoja_tizon_tardio	enfermedad	enfermedad	hoja_tizon_tardio	SI
40	hoja_pulgilla	plaga	plaga	hoja_pulgilla	SI
41	hoja_pulgilla	plaga	plaga	hoja_pulgilla	SI
42	hoja_sana	sano	sano	hoja_tizon_temprano	NO
43	hoja_sana	sano	sano	hoja_sana	SI
44	hoja_sana	sano	sano	hoja_sana	SI
45	hoja_tizon_temprano	enfermedad	enfermedad	hoja_tizon_temprano	SI
46	hoja_tizon_tardio	enfermedad	enfermedad	hoja_tizon_tardio	SI
47	hoja_tizon_tardio	enfermedad	enfermedad	hoja_tizon_tardio	SI

48	hoja_pulgilla	plaga	plaga	hoja_pulgilla	SI
49	hoja_tizon_tardio	enfermedad	enfermedad	hoja_tizon_tardio	SI
50	hoja_pulgilla	plaga	plaga	hoja_pulgilla	SI

*Elaborado por: Los investigadores

Tabla 17. Frecuencia y porcentaje de resultados de experto humano y modelo propuesto

Predicciones	Frecuencia	Porcentaje
Aciertos	45	90%
Errores	5	10%
Total	50	100%

*Elaborado por: Los investigadores



Figura 67. Cuadro estadístico de la tabulación

5.1.6. Comparativa De Modelos

La siguiente tabla presenta los datos obtenidos del entrenamiento de las 5 arquitecturas y el modelo propio, la comparativa proporciona una visión del rendimiento de cada modelo en la clasificación de imágenes, resaltando la eficacia de las arquitecturas estándar como la exploración de un modelo personalizado.

Tabla 18. Comparativa de las 5 arquitecturas y el modelo propio

Comparativa de las 5 Arquitecturas CNN y el modelo propio							
Arquitecturas	Preprocesamiento de imágenes	Épocas	Número de Capas	Funciones de activación	Algoritmo de Optimización	Tiempo de entrenamiento	Precisión
ResNet50	Resolución de imágenes: 224x224 píxeles	25	50 capas Max Pooling	Softmax	SGD (stochastic gradient descent)	8 horas con 17 min.	99,22%



	Trasformación de imágenes: Rotación, desplazamiento en anchura y altura.		Fine-Tuning				
VGG19	Resolución de imágenes: 224x224 píxeles Escalamiento Min-Max: Normalización (0,1)	20	20 capas Max Pooling	Softmax	ReLU	5 horas con 22 min.	82.81 %
InceptionV3	Resolución de imágenes: 299x299 píxeles Normalización: (0,1)	200	48 capas Average Pooling Max Pooling	Softmax	Adam	7 horas 34 min.	98.04%
EfficientNetB0	Técnica de aumento de datos, rotación a 90°	100	16 capas	ReLU Sigmoide	Adam CosineAnnealing inLR	8 horas con 21 min.	99,48%
MobileNet	Filtrado Gaussiano Escalamiento Min-Max: Normalización (0,1)	40	16 capas	Hardswish ReLU Sigmoid	Adam	3 horas con 47 min.	89,40%
Modelo Propio	Resolución de imágenes: 256x256 píxeles Normalización: (0,1)	50	54 capas	ReLU Softmax	Adam	3 horas con 15 min.	95.34%

*Elaborado por: Los investigadores

EfficientNetB0 ha destacado como la arquitectura con la mayor precisión, alcanzando un impresionante 99.48% con un tiempo de entrenamiento de 8 horas y 21 minutos. Esta eficiencia



resalta la capacidad de EfficientNetB0 para lograr resultados excepcionales con un número relativamente menor de capas.

Por otro lado, el modelo propio ha demostrado un rendimiento competitivo al lograr una precisión del 95.34%. Este resultado sugiere que, a pesar de contar con una mayor complejidad, el modelo desarrollado internamente ha alcanzado una notable eficacia en la tarea de clasificación de imágenes.

5.2. Metodología Mobile-D

En la metodología implementada se mostrará el primer conjunto de Historias de usuario, diagramas de caso de uso a detalle, diagramas de secuencia y casos de prueba, los demás elementos están siendo documentados en el apartado de Anexos.

5.2.1. Fase 1: Exploración

a) Establecimiento de Skateholders

Durante la etapa inicial de la primera fase, se establecieron los involucrados del proyecto junto con sus respectivas responsabilidades.

Tabla 19. Equipo de desarrollo

Equipo de desarrollo	Encargado
Líder del proyecto: Es la persona que tiene la responsabilidad de asegurar el cumplimiento de las tareas asignadas en el proyecto.	Ing. Mtr. Karla Cantuña
Programador: Es la persona encargada del desarrollo de la aplicación móvil.	Angel Gavilanez
Tester: Persona encargada de llevar a cabo pruebas en la aplicación con el fin de comprobar su conformidad de las funcionalidades definidas.	Bryan Saragosin

*Elaborado por: Los investigadores



b) Definición del alcance

El alcance de la aplicación móvil que se desarrolló como prototipo para ejemplificar la investigación realizada en este documento, la aplicación estará diseñada para detectar enfermedades y plagas en las hojas de las plantas de papa a través de la captura de imágenes instantáneas o la selección de imágenes desde la galería del dispositivo, además, la aplicación proporcionará información sobre el nombre de la enfermedad o plaga junto con un porcentaje que indica el nivel de certeza o precisión, presentando visualmente mediante una barra de progreso, también incluirá un texto informativo que sugiere posibles fungicidas específicos según el tipo de enfermedad o plaga detectada, facilitando la aplicación de medidas correctivas.

El procedimiento de identificación de imágenes se fundamenta en la implementación de una arquitectura de red neuronal convolucional (CNN), la cual pertenece al ámbito del aprendizaje automático (machine learning), este proyecto está diseñado para los agricultores de la ciudad de Latacunga donde se desarrollará toda la investigación.

c) Establecimiento del proyecto

Se definirá las herramientas que se emplearán, los entornos de desarrollo, los lenguajes de programación, el sistema de gestión de base de datos, los equipos utilizados y las metodologías aplicadas, los cuales se muestran a continuación.

- Entorno de desarrollo Android Studio Giraffe 2022.3.1
- Tensor Flow lite android studio 0.1.0
- Lenguaje de programación Java 8
- Gestor de base de datos SQLite3
- Navicat Premium 16

5.2.2. Fase 2: Inicialización

a) Especificación de Requisitos

Se determinan los requisitos funcionales:

Tabla 20. Requisitos Funcionales



ID	REQUERIMIENTO	DESCRIPCIÓN DEL REQUERIMIENTO	PRIORIDAD
RF01	Capturar imagen	La aplicación deberá permitir al usuario capturar imágenes utilizando la cámara del dispositivo móvil para el reconocimiento de enfermedades y plagas en las plantas de papa.	Alta
RF02	Seleccionar imagen de la galería	La aplicación deberá permitir al usuario elegir la imagen desde la galería de su dispositivo para el reconocimiento de enfermedades y plagas en las plantas de papa.	Alta
RF03	Visualizar el nombre de la enfermedad o plaga	La aplicación deberá permitir al usuario visualizar el nombre de la enfermedad o plaga de la hoja de la planta de papa.	Alta
RF04	Porcentaje de precisión	La aplicación deberá mostrar al usuario de manera numérica el porcentaje de precisión correspondiente al reconocimiento de la imagen.	Media
RF05	Barra de progreso	La aplicación deberá mostrar al usuario la precisión de reconocimiento de la imagen de manera visual a través de una barra de progreso.	Media
RF06	Crear base de datos	La aplicación deberá incluir la funcionalidad de crear una base	Media



		de datos que contenga los registros y resultados detallados sobre las enfermedades y los tratamientos asociados.	
RF07	Presentar resultados	La aplicación deberá mostrar el usuario el nombre del fungicida de acuerdo a la plaga o enfermedad identificada	Alta
RF08	Guardar resultados	La aplicación deberá permitir al usuario guardar los datos de los resultados obtenidos.	Alta
RF09	Listar resultados	La aplicación deberá permitir al usuario listar los resultados guardar.	Alta
RF10	Eliminar resultados	La aplicación deberá permitir al usuario eliminar el resultado guardado.	Alta
RF11	Seleccionar la ubicación	La aplicación deberá permitir al usuario seleccionar la Parroquia y Barrio donde se fotografió la imagen.	Media
RF12	Calcular la dosis del fungicida	La aplicación deberá permitir al usuario ingresar la dimensión del terreno y calcular la dosis a aplicar del fungicida.	Alta

*Elaborado por: Los investigadores

Se determinan los requisitos no funcionales:

Tabla 21. Requisitos No Funcionales

ID	REQUERIMIENTO	DESCRIPCIÓN DEL REQUERIMIENTO	PRIORIDAD
RNF01	Splash Screen	La aplicación deberá presentar al usuario una pantalla de inicio con el nombre del aplicativo incluido una imagen representativa durante el proceso de carga inicial.	Media
RNF02	Compatibilidad de dispositivo	La aplicación deberá ser compatible para dispositivos móviles con sistema Android.	Alta
RNF03	Pixeles de la cámara	La aplicación deberá ser compatible con cámara de 20px mínimo.	Media
RNF04	Conexión a internet	La aplicación deberá funcionar sin el uso de internet.	Alta

*Elaborado por: Los investigadores

b) Diagrama de casos de uso

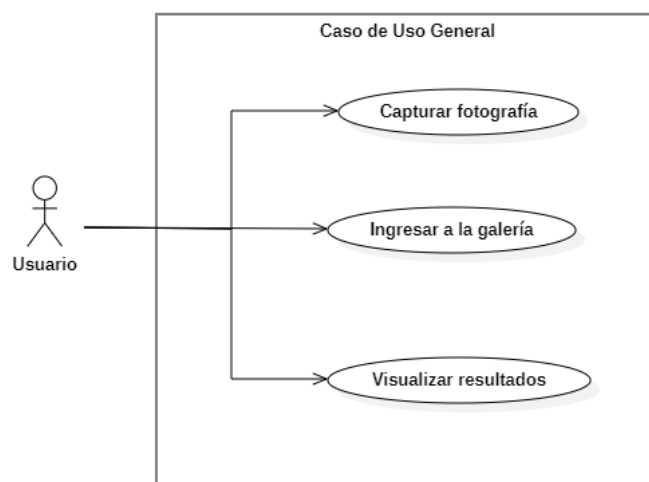


Figura 68. Diagrama de caso de uso general

c) Diagramas de caso de uso a detalle



Tabla 22. CU001 Capturar Fotografía

Núm.:	CU01
Nombre:	Capturar Fotografía
Autores:	Investigadores
Descripción:	El usuario realizará la detección de la imagen desde la cámara.
Actor:	Usuario
Precondiciones:	El usuario debe tener la aplicación.
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario ingresará a la aplicación. 2. El usuario seleccionará el botón “Tomar Foto” 3. La aplicación mostrará un mensaje de permisos del uso de la cámara. 4. La aplicación mostrará la interfaz para tomar la fotografía. 5. La aplicación activará la cámara del dispositivo. 6. El usuario tomará la fotografía. 7. La aplicación detectará la enfermedad o plaga de la hoja de la planta de papa. 8. El aplicativo mostrará el nombre de la plaga o enfermedad y el porcentaje de precisión.
Flujo alternativo 1:	<p>OPCIÓN INCORRECTA:</p> <ol style="list-style-type: none"> 2.1- El usuario no seleccionará el botón “Tomar Foto” 4.1 La aplicación no mostrará la interfaz para tomar la fotografía. 6.1 El usuario tomará una fotografía incorrecta 7.1 La aplicación no detectará la plaga o enfermedad
Post condiciones:	El usuario realizará la detección de la plaga o enfermedad en la hoja de la planta de la papa con éxito.

*Elaborado por: Los investigadores

d) Diagrama de la aplicación



Figura 69. Diagrama de la aplicación

e) Diagrama de clases aplicación – modelo

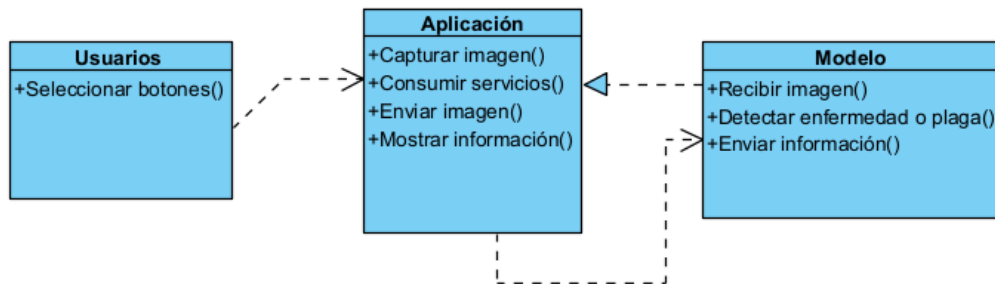


Figura 70. Diagrama de clases del aplicativo

f) Diagrama de base de datos

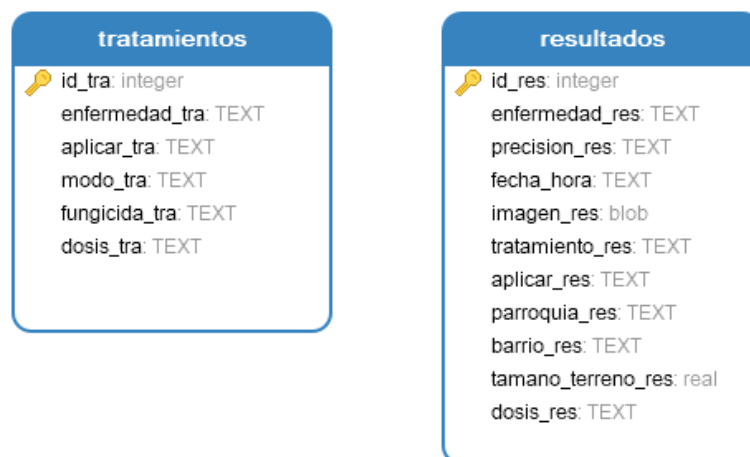


Figura 71. Diagrama de base de datos

g) Diagrama de secuencia

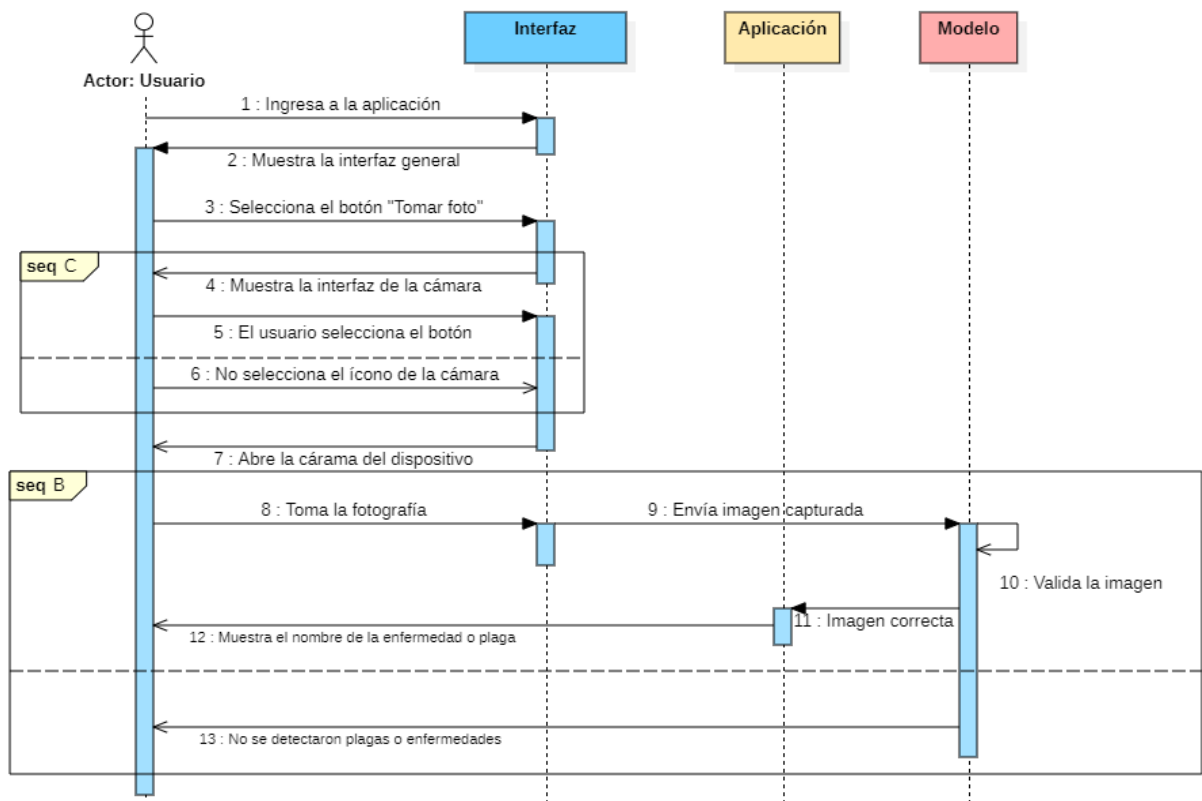






Figura 72. Diagrama de secuencia de captura de imagen

5.2.3. Fase 3: Producción

Luego de llevar a cabo el análisis de requisitos, se procedió a desarrollar las interfaces específicas para la aplicación móvil, las mismas que se integraron para el accionar del usuario.

Tabla 23. Interfaces de la aplicación móvil

Nombre	Imagen
Splash Screen	

<p>Captura de fotografía</p> <p>Selección de galería</p>	
<p>Visualizar resultados</p>	
<p>Lista de resultados guardados</p>	

*Elaborado por: Los investigadores

5.2.4. Fase 4: Estabilización

En esta fase se lleva a cabo la fusión del código fuente con las características propuestas, con el fin de garantizar la calidad durante la ejecución del proyecto.

Código clasificación de imágenes

```
public void classifyImage(Bitmap image) {
    try {
        Model model = Model.newInstance(getApplicationContext());

        // Controlando entradas de imágenes.
        TensorBuffer inputFeature0 = TensorBuffer.createFixedSize(new int[]{1, 256, 256, 3}, DataType.FLOAT32);
        ByteBuffer byteBuffer = ByteBuffer.allocateDirect( capacity: 4 * imageSize * imageSize * 3);
        byteBuffer.order(ByteOrder.nativeOrder());

        int[] intValues = new int[imageSize * imageSize];
        image.getPixels(intValues, offset: 0, image.getWidth(), x: 0, y: 0, image.getWidth(), image.getHeight());
        int pixel = 0;
        // Iterar sobre cada pixel y extraer los valores de R, G y B. Agregar esos valores individualmente al búfer de bytes.
        for (int i = 0; i < imageSize; i++) {
            for (int j = 0; j < imageSize; j++) {
                int val = intValues[pixel++]; // RGB
                byteBuffer.putFloat( value: ((val >> 16) & 0xFF) * (1.f / 1));
                byteBuffer.putFloat( value: ((val >> 8) & 0xFF) * (1.f / 1));
                byteBuffer.putFloat( value: (val & 0xFF) * (1.f / 1));
            }
        }

        inputFeature0.loadBuffer(byteBuffer);
    }
}
```

Figura 73. Código fuente visualizar resultados, clasificar imagen

Código presentación de resultados

```
10 usages
private int imageSize = 256;
8 usages
private DBmanager dbManager;

angee +1
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    // Establecer la barra de estado como transparente
    getWindow().setStatusBarColor(Color.parseColor("#88000000")); // Negro con 50% de opacidad

    // Inicializar DBmanager
    dbManager = new DBmanager( context: this);

    camera = findViewById(R.id.button);
    gallery = findViewById(R.id.button2);

    result = findViewById(R.id.result);
    imageView = findViewById(R.id.imageView);

    angee
    camera.setOnClickListener(new View.OnClickListener() {
        angee
    }
}
```



Figura 74. Código fuente tomar fotografía, seleccionar imagen

5.2.5. Fase 5: Pruebas

En este procedimiento, se describió exhaustivamente todas las pruebas del sistema con el objetivo de asegurar la ausencia de errores y garantizar su correcto funcionamiento.

a) Caso de prueba Captura de imagen

Tabla 24. Caso de prueba capturar imagen

ID/Nombre Caso de Prueba: CP01: Capturar imagen		Autor del caso de prueba: Bryan Saragosin, Angel Gavilanez	
Versión del caso de prueba: 1		Versión: 1.0	
Fecha de creación: 01/05/2024		Fecha de Ejecución: 01/05/2024	
Condición: El usuario deberá capturar la imagen			
Flujo caso de la prueba:			
Nº	Descripción	Resultado esperado	Resultado obtenido
1	La aplicación captura la imagen mediante la cámara del dispositivo móvil para poder ingresar en el sistema y detectar con el objeto de referencia.	La aplicación permita capturar la imagen mediante la cámara del dispositivo móvil para ingresar y que así pueda determinar el tipo de plaga o enfermedad tiene la hoja de la planta de papa.	La aplicación permitió capturar la imagen con la cámara del dispositivo, verificando la imagen y determinando el tipo de plaga o enfermedad que tiene la hoja de la planta.
Imagen			



Decisión de aprobación del caso del prueba: Aprobado: () Fallido: ()
Fecha de aprobación del caso de prueba: 07/02/2024

*Elaborado por: Los investigadores

5.3. Costo del software

El costo del software de la aplicación móvil se determinó utilizando el método de COCOMO, este modelo se emplea en el cálculo de proyectos de software y proporciona una estimación de la duración en meses que se requerirá del programador para llevar a cabo el desarrollo del proyecto.

5.3.1. Estimación de la cantidad de instrucciones

$$L = 1265 * FD \tag{1}$$

$$L = 1265 * 5 = 6325$$

Para la estimación se basó en la siguiente fórmula (1) en donde L significa la cantidad de líneas de código, FD E/S= flujo de entrada y salida que tiene la aplicación móvil.

$$M L = \frac{F}{10000} \tag{1.1}$$

$$M L = \frac{6325}{10000} = 0.632$$

En la ecuación (1.1) ML = miles de código fuente que tendrá la aplicación móvil.



5.3.2. Estimación del esfuerzo

$$ESF = 3 * M L^{1.12} \quad (2)$$

$$ESF = 3 * 0.632^{1.12} = 1.76$$

$$ESF = 2 \text{ persona}$$

En la fórmula (2) el 3 = modo de desarrollo del proyecto, y el ML = miles de línea de código del aplicativo móvil.

5.3.3. Estimación del tiempo de desarrollo

$$TDE = 2.5 * ESF^{0.35} \quad (3)$$

$$TDE = 2.5 * 1.76^{0.35} = 3.05$$

$$TDE = 3 \text{ meses}$$

En la fórmula (3) TDE = tiempo de desarrollo del proyecto, 2.5 = modo semiencajado dependiendo del problema, el grupo puede incluir personal experimentado y no experimentado, 0.35 = el tiempo de desarrollo en el modo semiencajado.

5.3.4. Estimación del personal necesario

$$CP = \frac{ESF}{TDE} \quad (4)$$

$$CP = \frac{1.76}{3.05} = 0.57 = 1$$

En la fórmula (4) ESF = estimación del esfuerzo, TDE = estimación de tiempo de desarrollo.

5.3.5. Estimación de productividad

$$P = \frac{L}{ESF} \quad (5)$$

$$P = \frac{6325}{1.76} = 3593 \frac{\text{instrucciones}}{\text{personas_mes}}$$

En la fórmula (5) detalla ESF = estimación del esfuerzo y TDE = estimación de tiempo de desarrollo.

5.3.6. Estimación del coste

$$PV = ESF * CHM \quad (6)$$

$$PV = 1.76 * 400 = \$704$$



En la fórmula (6) ESF = estimación del esfuerzo, CHM = sueldo del personal proyecto.

Para la realización del presupuesto del prototipo de la aplicación móvil para la detección de enfermedades y plagas en las hojas de la planta de papa, fue de un valor de \$704, la estimación de esfuerzo de 1.76, personal requerido para realizar este proyecto es de 2 personas.

5.4. Comprobación de la hipótesis

5.4.1. Juicio de experto

Para verificar la hipótesis, se llevó a cabo un análisis por parte de un experto en la materia, el Ingeniero Agrónomo Gil Washington Ochoa Oña, a través de un cuestionario específico, el experto confirmó que el proceso de entrenamiento de la red neuronal logró una precisión considerable, respaldando así los objetivos del proyecto que por medio de la tabla de evaluación de resultados, como se muestra en la tabla 17, con un total de 50 pruebas efectuadas, se obtuvieron 45 predicciones correctas y 5 predicciones erróneas, que da como resultado un porcentaje de acierto del 90 % de acierto y un porcentaje de error del 10 % según el cuadro estadístico de tabulación. Se alcanzó una exactitud considerable al proyecto de estudio para los procesos de clasificación de imágenes acorde a las plagas y enfermedades, por lo cual supo manifestar que la investigación desarrollada por parte del equipo de trabajo servirá de gran ayuda en la detección de plagas y enfermedades en el cultivo de papa, esto se encuentra solventado en el documento que se muestra en el anexo 8.7.

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

Además, la aplicación móvil desarrollada no solo facilita la toma de decisiones en tiempo real para los agricultores, sino que también ofrece un potencial escalable para su implementación en otras regiones agrícolas. Este enfoque tiene el poder de transformar la manera en que se abordan las problemáticas fitosanitarias en la agricultura, promoviendo una gestión más eficiente y sostenible.

La implementación de la metodología KDD ha sido fundamental para una extracción eficiente en la recopilación de las distintas fuentes de bases de datos relacionadas con las plagas y enfermedades de la papa, la metodología ha proporcionado un marco estructurado para la identificación, procesamiento y análisis de los datos, facilitando el desarrollo del modelo entrenado.



Con el desarrollo del prototipo de aplicación móvil se ha comprobado que el modelo creado tiene la capacidad de predecir plagas y enfermedades en el cultivo de papa, gracias al aprendizaje automático por medio del entrenamiento con el conjunto de datos de imágenes.

6.2. Recomendaciones

Es fundamental aumentar el dataset de imágenes con más enfermedades y plagas para conseguir un modelo mucho más robusto.

Fomentar la colaboración de expertos en la agricultura para enriquecer la investigación y entender las necesidades reales de los agricultores.

Con el fin de desarrollar nuevos modelos se propone la investigación continua de nuevas arquitecturas y la variación de parámetros, para lograr configuraciones que mejoren la eficacia del modelo en la detección de plagas y enfermedades en los cultivos de papas.

7. REFERENCIAS

- [1] M. V. Andrade Basantes, “Detección de cultivos infectados con tizón tardío en la papa (*Solanum tuberosum*) basado en imágenes”, Universidad De Las Fuerzas Armadas ESPE, 2023. [En línea]. Disponible en:
<https://repositorio.espe.edu.ec/bitstream/21000/36785/3/T-ESPE-058133.pdf>
- [2] W. Vásquez-Castillo y X. C. Ruales, Andrea Sevilla Rivadeneira Jorge Rivadeneira, “Resistencia genética como estrategia para el control de *Phytophthora infestans* en papa (*Solanum tuberosum*)”, *Ciencia y Tecnología Agropecuaria*, p. 15, 2022.
- [3] D. P. José, Fausto Yumisaca, Victoria López y J. C. C. T. Tello, “ALERTA TEMPRANA PARA EL MANEJO DEL TIZÓN TARDIO DE LA PAPA. ATN/RF 16678 RG”. [En línea]. Disponible en:
https://www.fontagro.org/new/uploads/productos/16678_-_Producto_7_.pdf
- [4] “Plagas de papa en Ecuador podrían diseminarse a la región Andina”. [En línea]. Disponible en: <https://cipotato.org/es/blog-es/plagas-papa-ecuador-podria-diseminarse/>
- [5] R. la Hora, “Plaga afecta a las papas de Cotopaxi”. [En línea]. Disponible en: <https://www.lahora.com.ec/noticias/plaga-afecta-a-las-papas-de-cotopaxi/>
- [6] R. Corape, “COTOPAXI: En Cotopaxi cultivos de papas son afectados por diferentes enfermedades y plagas”. Consultado: el 3 de diciembre de 2023. [En línea].



Disponible en: <https://radio.corape.org.ec/noticia/item/cotopaxien-cotopaxi-cultivos-de-papas-son-afectados-por-diferentes-enfermedades-y-plagas>

- [7] Z. Saeed, M. U. Khan, A. Raza, N. Sajjad, S. Naz, y A. Salal, “Identification of Leaf Diseases in Potato Crop Using Deep Convolutional Neural Networks (DCNNs)”, *ICET 2021 - 16th International Conference on Emerging Technologies 2021, Proceedings*, 2021, doi: 10.1109/ICET54505.2021.9689807.
- [8] S. V. Militante, B. D. Gerardo, y N. V. Dionisio, “Plant Leaf Detection and Disease Recognition using Deep Learning”, *2019 IEEE Eurasia Conference on IOT, Communication and Engineering, ECICE 2019*, pp. 579–582, 2019, doi: 10.1109/ECICE47484.2019.8942686.
- [9] X. E. Pantazi, D. Moshou, y A. A. Tamouridou, “Automated leaf disease detection in different crop species through image features analysis and One Class Classifiers”, *Comput Electron Agric*, vol. 156, núm. November 2018, pp. 96–104, 2019, doi: 10.1016/j.compag.2018.11.005.
- [10] J. Gao, J. C. Westergaard, E. H. R. Sundmark, M. Bagge, E. Liljeroth, y E. Alexandersson, “Automatic late blight lesion recognition and severity quantification based on field imagery of diverse potato genotypes by deep learning”, *Knowl Based Syst*, vol. 214, p. 106723, 2021, doi: 10.1016/j.knosys.2020.106723.
- [11] A. Arshaghi, M. Ashourian, y L. Ghabeli, “Potato diseases detection and classification using deep learning methods”, *Multimed Tools Appl*, vol. 82, núm. 4, pp. 5725–5742, 2023, doi: 10.1007/s11042-022-13390-1.
- [12] D. A. del Puerto y P. G. Esteban, “La Inteligencia Artificial como recurso educativo durante la formación inicial del profesorado”, *RIED-Revista Iberoamericana de Educación a Distancia*, vol. 25, núm. 2, pp. 347–358, 2022.
- [13] Valenzuela S, “Detección y Clasificación de Enfermedades en el Tomate Mediante Deep Learning y Computer Vision”, Universidad Nacional de La Plata, La Plata, 2021.
- [14] Forero W y Negre F, “Técnicas y aplicaciones del Machine Learning e Inteligencia Artificial en educación: una revisión sistemática. ”, *RIED-Revista Iberoamericana De Educación a Distancia*, vol. 27, núm. 1, pp. 209–253, 2023.



- [15] Romon I, “Clasificación de imágenes espaciales mediante visión artificial”, Universidad Politécnica de Madrid, Madrid, 2022.
- [16] Guaña J, Salgado N, y Escobar Ch, “La informática y la ciencia de datos utilizando Deep learning”, *Revista Ibérica de Sistemas e Tecnologías de Informação*, vol. 49, pp. 116–127, 2022.
- [17] Casado E, “Detección de anomalías mediante redes neuronales convolucionales (CNN) en radio espectrómetros solares”, Universidad de Alcalá, 2022.
- [18] V. García Muñoz, “PARAMETRIZACIÓN Y OPTIMIZACIÓN DE UNA RED NEURONAL RECURRENTE APLICADA A LA GENERACIÓN DE TEXTO”, Universidad Pontificia Comillas, Madrid, 2020. Consultado: el 3 de febrero de 2024. [En línea]. Disponible en: https://repositorio.comillas.edu/xmlui/bitstream/handle/11531/46468/TFG_GarciaMunoz_Venancio.pdf?sequence=1
- [19] Andrade Basantes Marco Vinicio, “Detección de cultivos infectados con tizón tardío en la papa (*solanum tuberosum*) basado en imágenes”, Universidad de Las Fuerzas Armadas, Sangolquí, 2023. Consultado: el 6 de febrero de 2024. [En línea]. Disponible en: file:///C:/Users/ANGEL/Desktop/8vo_CICLO/1_Tesis/tesis_ejemplos/T-ESPE-058133.pdf
- [20] Manuela Viviana Chacón Chamorro, “Estudio de la reducción del sobreajuste en arquitecturas de redes neuronales residuales ResNet en un escenario de clasificación de patrones”, Universidad Nacional de Colombia, Colombia, 2023. Consultado: el 6 de febrero de 2024. [En línea]. Disponible en: <https://repositorio.unal.edu.co/bitstream/handle/unal/84211/1085325637.2023.pdf?sequence=2&isAllowed=y>
- [21] B. Romero, “Una Introducción A Los Modelos De Machine Learning”, Benemérita Universidad Autónoma de Puebla, Puebla, 2020.
- [22] D. Martínez y S. Ramírez, “Sistema de Inventario y proceso de abastecimiento en la empresa TextiNort”, Universidad Politécnica Estatal del Carchi, Tulcán, 2023. Consultado: el 5 de febrero de 2024. [En línea]. Disponible en: <http://repositorio.upec.edu.ec/bitstream/123456789/1763/1/077-%20CASTILLO%20DENYSSE%20-%20RAMIREZ%20SOFIA.pdf>



- [23] F. Gutierrez, A. Cota, M. Sierra, y C. Ortega, “Reconocimiento de la denominación de billetes a través de una aplicación móvil con reconocimiento de imagen”, *Revista electrónica de Computación, Informática, Biomédica y Electrónica*, vol. 9, núm. 1, pp. 1–16, may 2020, Consultado: el 5 de febrero de 2024. [En línea]. Disponible en: <https://www.redalyc.org/journal/5122/512267930001/512267930001.pdf>
- [24] J. Aviñó, “Estudio y desarrollo de transfer learning aplicado a la segmentación automática de estructuras morfológicas para el apoyo, el análisis y el diagnóstico médico.”, Trabajo Fin de Grado, Escuela Politécnica de Valencia, Valencia, 2022. Consultado: el 16 de enero de 2024. [En línea]. Disponible en: <https://riunet.upv.es/bitstream/handle/10251/187654/Avino%20-%20Estudio%20y%20desarrollo%20de%20transfer%20learning%20aplicado%20a%20la%20segmentacion%20automatica%20de%20estruc....pdf?sequence=1&isAllowed=y>
- [25] Hernández D, “Galaxy Zoo: Modelo para clasificación de galaxias”, Universidad de Antioquia, Medellín, 2023.
- [26] “Guía avanzada de Inception v3”, Google Cloud. Consultado: el 6 de febrero de 2024. [En línea]. Disponible en: <https://cloud.google.com/tpu/docs/inception-v3-advanced?hl=es-419#:~:text=Inception%20v3%20es%20un%20modelo,conjunto%20de%20datos%20de%20ImageNet.>
- [27] Salazar D, Flores J, y Barba L, “Tecnicas de visión por computadora para la estimación de la intensidad, la velocidad y la densidad en el flujo vehicular”. 2021.
- [28] Y. A. ; Z. D. C. J. T. Y. A.-N. N. Nanehkaran, “Recognition of plant leaf diseases based on computer vision”, 2020, pp. 1–18, 2020. Consultado: el 23 de febrero de 2024. [En línea]. Disponible en: <https://access.clarivate.com/login?app=wos&alternative=true&shibShireURL=https%3A%2F%2Fwww.webofknowledge.com%2F%3Fauth%3DShibboleth&shibReturnURL=https%3A%2F%2Fwww.webofknowledge.com%2F%3FDestApp%3DUA%26action%3Dtransfer%26mode%3DNextgen%26path%3D%252Fwos%252Fwoscc%252Ffull-record%252FWOS%253A000566045300001&referrer=mode%3DNextgen%26path%3D%252Fwos%252Fwoscc%252Ffull->



record%252FWOS%253A000566045300001%26DestApp%3DUA%26action%3Dtransfer&roaming=true

- [29] R. Arumuga Arun y S. Umamaheswari, “Effective multi-crop disease detection using pruned complete concatenated deep learning model”, *Expert Syst Appl*, vol. 213, mar. 2023, doi: 10.1016/j.eswa.2022.118905.
- [30] J. Gao, J. C. Westergaard, E. H. R. Sundmark, M. Bagge, E. Liljeroth, y E. Alexandersson, “Automatic late blight lesion recognition and severity quantification based on field imagery of diverse potato genotypes by deep learning”, *Knowl Based Syst*, vol. 214, feb. 2021, doi: 10.1016/j.knosys.2020.106723.
- [31] A. Arshaghi, M. Ashourian, y L. Ghabeli, “Potato diseases detection and classification using deep learning methods”, *Multimed Tools Appl*, vol. 82, núm. 4, pp. 5725–5742, feb. 2023, doi: 10.1007/s11042-022-13390-1.
- [32] K. Sathya y M. Rajalakshmi, “RDA-CNN: Enhanced Super Resolution Method for Rice Plant Disease Classification”, *Computer Systems Science and Engineering*, vol. 42, núm. 1, pp. 33–47, 2022, doi: 10.32604/CSSE.2022.022206.
- [33] Kongunadu College of Engineering & Technology y Institute of Electrical and Electronics Engineers, *Proceedings, International Conference on Smart Electronics and Communication (ICOSEC 2020) : 10-12, September 2020*.
- [34] Cervera E, Marín R, y Marín J, “Más allá de Jupyter: usando Google Colab para la programación de robots”. pp. 662–669, 2022.
- [35] Zambrano J, “Comparación y análisis de métodos de clasificación con las bibliotecas scikit-learn y TensorFlow en Python”, Universidad de Málaga, Málaga, 2019.
- [36] Raschka A y Mirjalili V, *Python Machine Learning*, Segunda. 2019.
- [37] A. Fragua, “Ejecución de redes neuronales en móviles Android con aceleración hardware mediante Keras y Tensorflow Lite”, Universidad Autónoma de Madrid, Madrid, 2021. Consultado: el 19 de enero de 2024. [En línea]. Disponible en: <https://repositorio.uam.es/handle/10486/698211>
- [38] S. Rodríguez, “SISTEMA OPERATIVO ANDROID DEL FUTURO”, Moquegua, 2022. Consultado: el 19 de enero de 2024. [En línea]. Disponible en: <https://repositorio.unam.edu.pe/server/api/core/bitstreams/a28a3878-23b7-4554-8c26-d815c2c61a90/content>



- [39] Santos S, “Desarrollo de Aplicación Móvil de Baloncesto en Android Studio: BasketScore”, Universidad Politécnica de Madrid, Madrid, 2023.
- [40] E. Vaati, “Qué es Android SDK y cómo empezar a usarlo”. Consultado: el 19 de enero de 2024. [En línea]. Disponible en: <https://code.tutsplus.com/es/the-android-sdk-tutorial--cms-34623t>
- [41] X. Cuesta y J. Rivadeneira, “Estado actual de la investigación de la papa en el Ecuador”, en *Artículos del Noveno Congreso Ecuatoriano de la Papa*, INIAP-CIP., M. Racines, X. Cuesta, J. Rivadeneira, y J. L. Pantoja, Eds., Latacunga, 2021, pp. 15–17.
- [42] T. Lucía, C. Xavier, M. Cecilia, y R. Jorge, “INVENTARIO DE TECNOLOGÍAS E INFORMACIÓN PARA EL CULTIVO DE PAPA EN ECUADOR”. Consultado: el 12 de enero de 2024. [En línea]. Disponible en: <https://cipotato.org/papaenecuador/variedades-de-papa/#:~:text=De%20las%20350%20variedades%20que,colorada%2C%20Santa%20Rosa%20y%20Carrizo.>
- [43] H. Saragosin, “Revisión bibliográfica de las principales plagas y enfermedades del cultivo de papa (*Solanum tuberosum*) en la provincia de Cotopaxi en el año 2023”, Universidad Técnica de Cotopaxi, Laacunga, 2023.
- [44] AgriSolver, “Comunidad para Soluciones en Agricultura”. Consultado: el 19 de enero de 2024. [En línea]. Disponible en: <https://www.agrisolver.com/>
- [45] Solagro, “Ficha Técnica PROFENOPAC”. Consultado: el 12 de enero de 2024. [En línea]. Disponible en: https://agripac.com.ec/wp-content/uploads/2021/02/PROFENOPAC_FT.pdf
- [46] C. Tello *et al.*, “Principios de manejo de control químico del tizón tardío de la papa en Ecuador”, pp. 91–93, jul. 2015.
- [47] Plantix, “Tizón temprano”. Consultado: el 12 de enero de 2024. [En línea]. Disponible en: <https://plantix.net/es/library/plant-diseases/100321/early-blight/>
- [48] G. Victor *et al.*, “Aplicación Móvil con Realidad Aumentada, utilizando la metodología Mobile-D, para el marketing relacional en el museo histórico del Mobiliario”, *Iberian Journal of Information Systems and Technologies*, vol. 4, núm. E57, pp. 1–13, 2023.



8. ANEXOS

8.1. Anexos: Diagramas de caso de uso a detalle

Tabla 25. CU02 Ingresar a la galería

Núm.: CU02	
Nombre:	Ingresar a la galería
Autores:	Investigadores
Descripción:	El usuario realizará la detección de la imagen desde la una imagen guardada en la galería del dispositivo móvil.
Actor:	Usuario
Precondiciones:	El usuario deberá tener la aplicación y tener una imagen de la hoja de la planta de papa en la galería.
Flujo Normal:	<ol style="list-style-type: none">1. El usuario ingresará a la aplicación.2. El usuario seleccionará el botón “Abrir Galería”3. La aplicación mostrará la interfaz para seleccionar la imagen de la galería.4. El usuario seleccionará la imagen.5. La aplicación activará la cámara del dispositivo.6. La aplicación detectará la enfermedad o plaga de la hoja de la planta de papa.7. El aplicativo mostrará el nombre de la plaga o enfermedad y el porcentaje de precisión.
Flujo alternativo 1:	<p>OPCION INCORRECTA:</p> <ol style="list-style-type: none">2.1- El usuario no seleccionará el botón “Abrir Galería”2.1 El usuario regresará al paso 1.4.1 El usuario seleccionará una imagen incorrecta6.1 La aplicación no detectará la plaga o enfermedad
Post condiciones:	El usuario realizará la detección de la plaga o enfermedad en la hoja de la planta de la papa con éxito.

Tabla 26. CU03 Visualizar resultados



Núm.:	CU03
Nombre:	Visualizar resultados
Autores:	Investigadores
Descripción:	La aplicación muestra al usuario los resultados de la plaga o enfermedad detectada, con una barra de progreso y el porcentaje de precisión, además, muestra el fungicida para el tratamiento.
Actor:	Usuario
Precondiciones:	El usuario deberá tener la aplicación.
Flujo Normal:	<ol style="list-style-type: none">1. El usuario ingresará a la aplicación.2. La aplicación mostrará los resultados obtenidos en la pantalla principal.3. La aplicación mostrará el nombre de la plaga o enfermedad y el porcentaje de precisión.4. La aplicación mostrará la barra de progreso5. La aplicación mostrará los fungicidas para el tratamiento de acuerdo a la plaga o enfermedad.
Flujo alternativo 1:	OPCION INCORRECTA: 2.1- La aplicación no mostrará resultados de ninguna enfermedad o plaga. 2.2 La aplicación mostrará un mensaje “Esto no parece un cultivo”
Post condiciones:	El usuario visualizará la detección de la plaga o enfermedad en la hoja de la planta de la papa con éxito.

*Elaborado por: Los investigadores

8.2. Anexos: Diagramas de secuencia

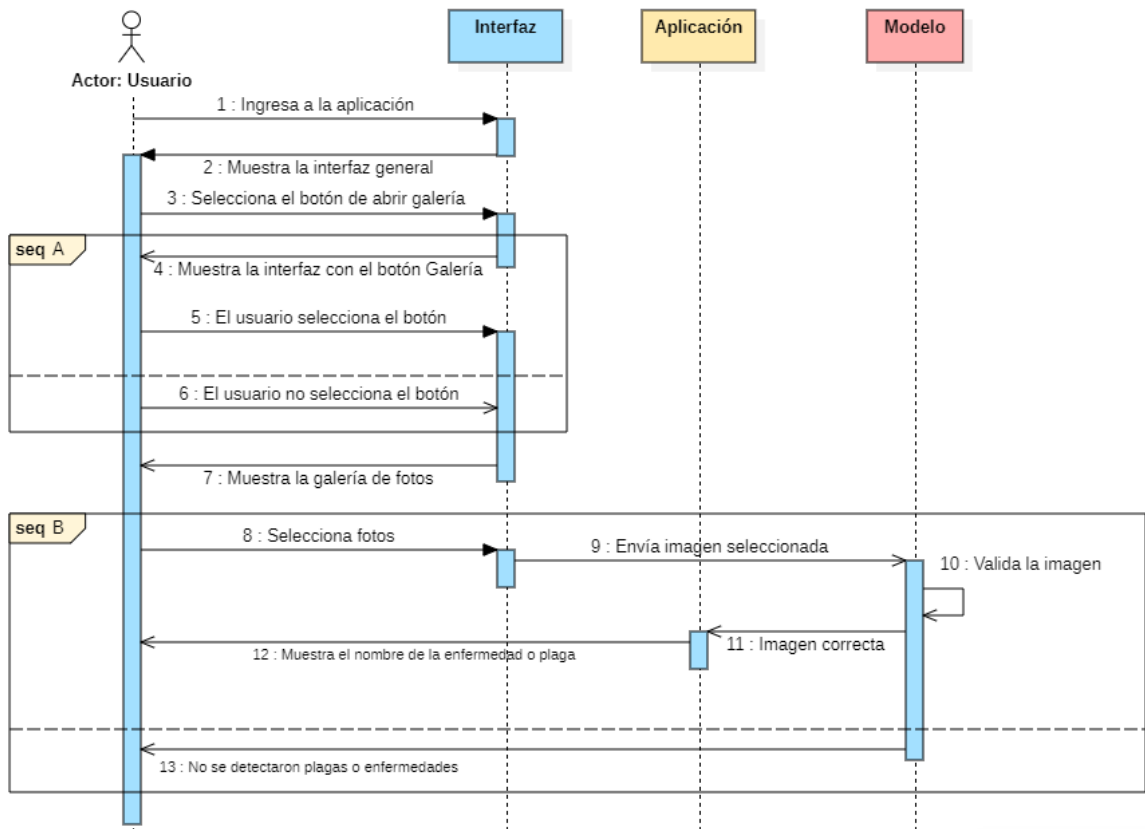


Figura 76. Diagrama de secuencia ingresar a la galería

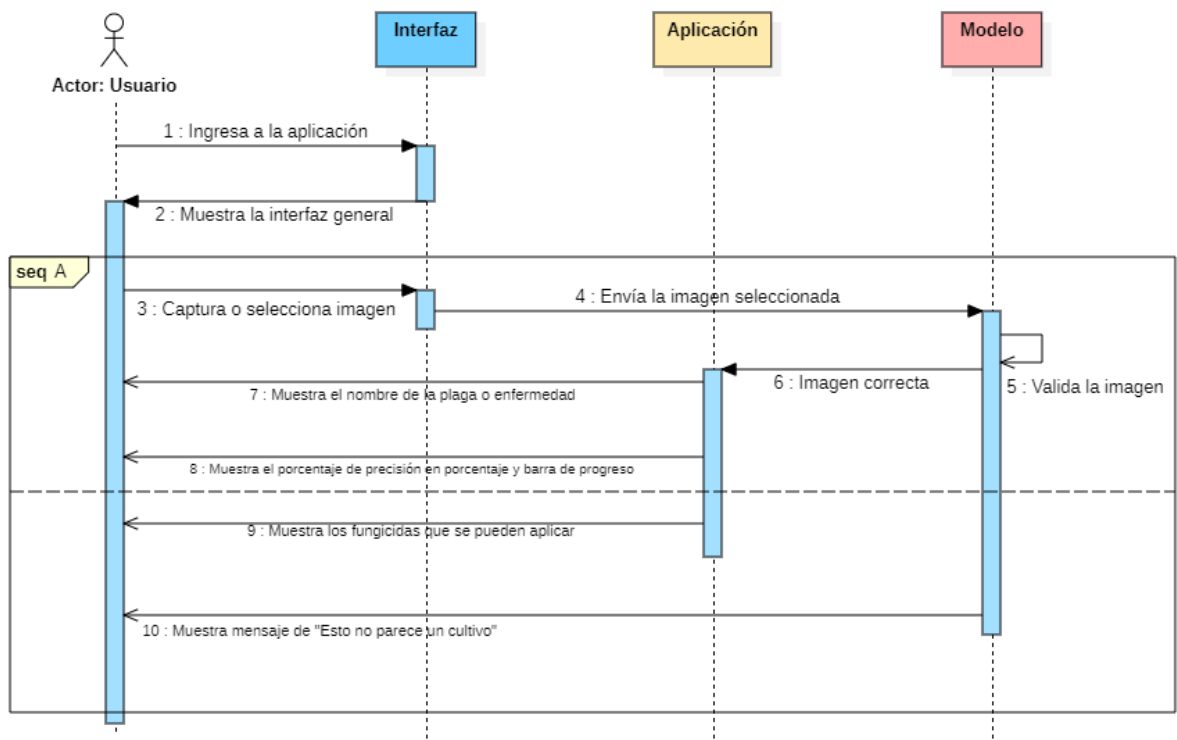


Figura 77. Diagrama de secuencia visualizar resultados

8.3. **Anexos:** Casos de Pruebas

Caso de prueba seleccionar imagen

Tabla 27. Caso de prueba Seleccionar imagen

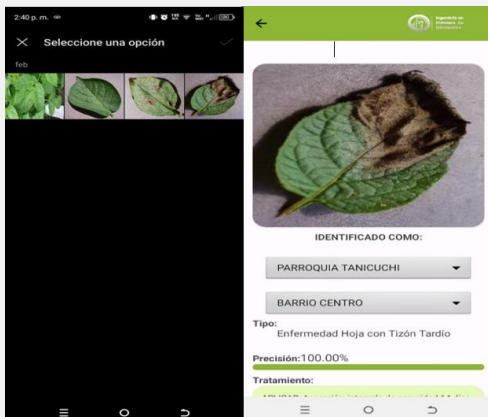
ID/Nombre Caso de Prueba: CP02: Seleccionar imagen		Autor del caso de prueba: Bryan Saragosin, Angel Gavilanez	
Versión del caso de prueba: 1		Versión: 1.0	
Fecha de creación: 01/05/2024		Fecha de Ejecución: 01/05/2024	
Condición: El usuario deberá seleccionar la imagen			
Flujo caso de la prueba:			
Nº	Descripción	Resultado esperado	Resultado obtenido
1	La aplicación permitirá seleccionar la imagen por medio de la galería.	La aplicación permita seleccionar la imagen de la y así pueda determinar el tipo de plaga o enfermedad tiene la hoja de la planta de papa.	La aplicación permitió seleccionar la imagen con la cámara del dispositivo, verificando la imagen y determinando el tipo de plaga o enfermedad que tiene la hoja de la planta.
Imagen			
			
Decisión de aprobación del caso del prueba: Aprobado: (X) Fallido: ()			

Figura 78. Caso de prueba seleccionar imagen

Fecha de aprobación del caso de prueba: 07/02/2024

*Elaborado por: Los investigadores

Visualizar los resultados

Tabla 28. Caso de prueba visualizar los resultados

ID/Nombre Caso de Prueba: CP03: Visualizar los resultados.		Autor del caso de prueba: Bryan Saragosin, Angel Gavilanez	
Versión del caso de prueba: 1		Versión: 1.0	
Fecha de creación: 01/05/2024		Fecha de Ejecución: 01/05/2024	
Condición: El usuario deberá visualizar los resultados.			
Flujo caso de la prueba:			
Nº	Descripción	Resultado esperado	Resultado obtenido
1	La aplicación mostrará los resultados de la identificación de la imagen.	La aplicación mostrará los resultados como: el nombre de la enfermedad o plaga, barra de progreso y porcentaje de precisión, los fungicidas que se podrían aplicar para el tratamiento de la plaga o enfermedad.	La aplicación mostró como resultados el nombre de la enfermedad, la barra de progreso del nivel de precisión, porcentaje de precisión, los fungicidas que se pueden aplicar para el tratamiento de la enfermedad.

Imagen

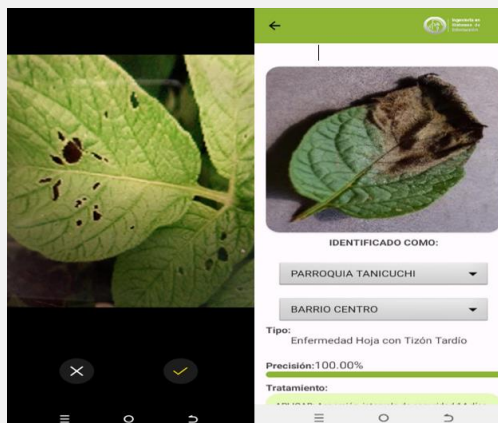


Figura 79. Caso de prueba visualizar resultados



Decisión de aprobación del caso del prueba: Aprobado: (X) Fallido: ()

Fecha de aprobación del caso de prueba: 07/02/2024

*Elaborado por: Los investigadores

Guardar los resultados

Tabla 29. Caso de prueba guardar los resultados

ID/Nombre Caso de Prueba: CP03: Guardar los resultados.		Autor del caso de prueba: Bryan Saragosin, Angel Gavilanez	
Versión del caso de prueba: 1		Versión: 1.0	
Fecha de creación: 01/05/2024		Fecha de Ejecución: 01/05/2024	
Condición: El usuario deberá guardar los resultados.			
Flujo caso de la prueba:			
Nº	Descripción	Resultado esperado	Resultado obtenido
1	la 30. La aplicación guardará los resultados de la identificación de la imagen.	La aplicación guardará los resultados como: el nombre de la enfermedad o plaga, la precisión, foto tomada o seleccionada, nombre del fungicida, la dosis, el modo de aplicar, parroquia, barrio, la fecha y hora	La aplicación guardó correctamente los resultados como: el nombre de la enfermedad o plaga, la precisión, foto tomada o seleccionada, nombre del fungicida, la dosis, el modo de aplicar, parroquia, barrio, la fecha y hora.
Imagen			




Figura 80. Caso de prueba visualizar resultados

Decisión de aprobación del caso del prueba: Aprobado: (X) Fallido: ()	
Fecha de aprobación del caso de prueba: 07/02/2024	

*Elaborado por: Los investigadores

8.4. Anexo: Formulación de la Entrevista

UNIVERSIDAD TÉCNICA DE COTOPAXI FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS CARRERA DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN

ENTREVISTA

¿Cuáles son los principales desafíos que enfrenta actualmente el cultivo de papa en la ciudad de Latacunga?

Los principales desafíos se basan en las enfermedades que se presentan en los cultivos de la papa, porque daña totalmente la producción y por ende la cosecha tiende a ser baja.

Además del costo, ¿cómo evalúa la eficacia de los fertilizantes utilizados?

Son los que comúnmente se aplica ante este tipo de enfermedades, se los encuentra con facilidad en el mercado y en relación con el precio son accesibles.

¿Cómo elige los fungicidas específicos para abordar plagas y enfermedades en el cultivo de papa?



En base a conocer que tipo de enfermedad o plaga está presente en la papa, se puede elegir el respectivo fungicida.

¿Cuáles son las señales tempranas de infestación por plagas o enfermedades en los cultivos de papa?

Por lo general la visualización de la hojas, en lo que respecta a su pigmentación, la aparición de manchas, el estado mismo de la hoja es el principal factor para tener una noción de la enfermedad en la papa, y en lo que respecta a las plagas es ver si en la hoja, tallo y ramas hay insectos puestos que los mismos se comen el cultivo, dejando orificios visibles.

¿Existe algún programa de capacitación específico relacionado con la identificación y gestión de plagas y enfermedades en el cultivo de papa?

Claro, el INIAP es el encargado de estas capacitaciones.

¿Cómo visualiza la integración de una aplicación móvil en las prácticas agrícolas existentes?

Le veo muy interesante porque en la actualidad con el tema de la inteligencia artificial, se ve que ayuda a obtener mejores resultados.

¿Qué tipo de información considera crucial para incluir en la aplicación para una detección efectiva de plagas y enfermedades?

Agregar más enfermedades y plagas que no son comunes, pero que pueden aparecer.

¿Cuál cree que podría ser el impacto económico de implementar una solución basada en Inteligencia Artificial para la detección de plagas y enfermedades?

En lo que respecta al tema económico, podría ayudar a disminuir las pérdidas de cultivo, porque con el aplicativo ya se sabría la posible enfermedad que tiene la papa, y de esta manera actuar con rapidez.

¿Ve posibles colaboraciones entre agricultores locales y desarrolladores de tecnología para abordar los desafíos del cultivo de papa?

Sí, porque sería un beneficio tanto para los agricultores como para los investigadores, un trabajo en conjunto que sería muy positivo.



8.5. Anexos: Características de hardware y software de los equipos utilizados para el entrenamiento de los modelos

Tabla 31. Especificaciones de la computadora N°1

Características computadora N°1			
Sistema Operativo:	Windows 10 con 64 bits		
Procesador:	Intel(R) Core (TM) i5-10300H CPU	Frecuencia:	2.50GHz
Memoria RAM:	16 GB (2 ranuras de 8 GB)		
Memoria ROM:	SSD 240 (Sistema Operativo) HDD 1TB (Almacenamiento)		
Tarjeta gráfica:	NVIDIA GeForce GTX 1650 con 4GB GDDR5 (Dedicada)		

*Elaborado por: Los investigadores

Tabla 32. Especificaciones de la computadora N°2

Características computadora N°2			
Sistema Operativo:	Windows 10 con 64 bits		
Procesador:	AMD Ryzen 7 de la serie 3700U	Frecuencia:	2.30GHz
Memoria RAM:	8 GB (2 ranuras de 4 GB)		
Memoria ROM:	SSD 500 GB		
Tarjeta gráfica:	Radeon Vega Mobile Graphics (Integrada)		

*Elaborado por: Los investigadores

8.6. Anexos: Juicio del experto

UNIVERSIDAD TÉCNICA DE COTOPAXI FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS CARRERA DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN

La información proporcionada a continuación, referente a los tratamientos recomendados para el control de plagas y enfermedades en el cultivo de papa, ha sido certificada y aprobada por el Ingeniero Agrónomo, Ing. Gil Washington Ochoa Oña, con número de cédula 0502528664.

En particular, los tratamientos más efectivos, claramente identificados y subrayados en cada tabla, han sido destacados por el respaldo del Ingeniero Agrónomo.

A continuación, se detallan los tratamientos recomendados, sus modos de acción, métodos de aplicación y dosis correspondientes.

Tratamientos recomendados para plagas y enfermedades

Plagas

Pulguilla de la Papa

Fungicida	Modos de acción	Aplicación	Dosis
KUIK 90 PS	Contacto Ingestión	Aspersión, intervalo de seguridad 14 días.	2 – 3 kg por hectárea

Enfermedades

Tizón tardío

Fungicida	Modos de acción	Aplicación	Dosis
ALLIETE	Sistémico	Aspersión por un intervalo de seguridad de 14 días	2 – 3 kg por hectárea

Figura 81. Juicio del experto

Tizón temprano

Fungicida	Modos de acción	Aplicación	Dosis
CUPRAVIT	Contacto	Aspersión	2kg – 4kg por hectárea



Ing. Gil Washington Ochoa Oña.
C.I. 0502528664
Ingeniero Agrónomo

Figura 82. Juicio del experto

8.7. Anexo: Ficha Bibliográfica

FICHA BIBLIOGRÁFICA						
N°	AÑO	TÍTULO	AUTORES	ÁREA DE INTELIGENCIA ARTIFICIAL	PRECISIÓN Y MODELO CNN	OBSERVACIONES
WEB OF SCIENCE						
2	2020	Recognition of plant leaf diseases based on computer vision	Nanehkarán, Y. A.; Zhang, Defu; Chen, Junde; Tian, yuan; Al-Nabhan, Najla	Procesamiento de imágenes	99,78% (ResNet-V2)	Utilizaron un enfoque de segmentación y clasificación combinado para identificar de manera precisa las enfermedades gracias a un algoritmo de segmentación híbrida basado en tono, saturación e intensidad.
8	2023	Effective multi-crop disease detection using pruned complete concatenated deep learning model	Arun, R. Arumuga; Umamaheswari, S.	Deep Learning	98,14% (Arquitectura Propia)	Los autores propusieron una arquitectura denominada Complete Concatenated Deep (CCDL) que consiguió una precisión del 98,14%
9	2021	Automatic late blight lesion recognition and severity quantification based on field imagery of diverse potato genotypes by deep learning	Gao, Junfeng; Westergaard, Jesper Cairo; Sundmark, Ea Hoegh Riis; Bagge, Merethe; Liljeroth, Erland; Alexandersson, Erik	Deep Learning y procesamiento de imágenes	99,60% (SegNet)	Emplearon el modelo SegNet, basado en una arquitectura codificador-decodificador, la cual consiste en la segmentación semántica de lesiones foliares dando un resultado del 99,60% de precisión.

Figura 83. Ficha Bibliográfica

8.8. Anexo: Pruebas APK UTC Papa IA con la plataforma Sauce Labs

Resultados de la prueba / Detalles de la prueba

UTC Papa IA (Versión: 1.0; Compilación: 1)
18 de febrero de 2024 a las 13:09 por @oauth-baecentrumplay-59342

10 samsung galaxy s9 Equipo Editor 1m 37s

Interacciones (21)	Video	Registros (1)	Partes Vitales	Metadatos
Desplazado com.utcpapa.miapp.id/recyclerViewResultados 01:02:30	ID de tuner asignado : nulo			Configuración Básica : nulo
Desplazado com.utcpapa.miapp.id/recyclerViewResultados 01:02:94	Grabar Capturas De Pantalla : verdadero			Grabar Video : verdadero
aprovechado com.utcpapa.miapp.id/btnEliminarResultado 01:04:71	Datos Personalizados : nulo			Público : equipo
aprovechado ic_volver_encabezado 01:09:20	Manual : verdadero			
Desplazado 01:39:37				
Botón Atrás presionado 01:41:20				

Figura 84. Pruebas APK UTC Papa IA con la plataforma Sauce Labs

8.9. Anexo: Aplicación de entrevista al experto humano



Figura 85. Entrevista experto humano



8.10. Anexo: Hoja de vida investigador 1

<p style="text-align: center;"><u>HOJA DE VIDA</u> <u>INVESTIGADOR 1</u></p> <p style="text-align: center;"></p> <p><u>DATOS PERSONALES</u></p> <p>Apellidos y Nombres: Gavilanez Guanoluisa Angel Rodrigo</p> <p>Cédula de ciudadanía: 0550712962</p> <p>Fecha de nacimiento: 19-02-2001</p> <p>Estado civil: Soltero</p> <p>Email institucional: angel.gavilanez2962@utc.edu.ec</p> <p>Teléfono: 0992829599</p> <p><u>FORMACIÓN ACADÉMICA:</u></p> <p>Instrucción secundaria: Unidad Educativa San José La Salle</p> <p>Título obtenido: Bachiller en Ciencias</p> <p>Instrucción de tercer nivel: Universidad Técnica de Cotopaxi</p> <p>Carrera: Ingeniería en Sistemas de Información</p>

Figura 86. Hoja de vida investigador 1



8.11. Anexo: Hoja de vida investigador 2

<p style="text-align: center;"><u>HOJA DE VIDA</u> <u>INVESTIGADOR 2</u></p> <p style="text-align: center;"></p> <p><u>DATOS PERSONALES</u></p> <p>Apellidos y Nombres: Saragosin Guamushig Bryan Alexander</p> <p>Cédula de ciudadanía: 0550054134</p> <p>Fecha de nacimiento: 05-08-1998</p> <p>Estado civil: Soltero</p> <p>Email institucional: alexsaragosin5@gmail.com</p> <p>Teléfono: 0986568067</p> <p><u>FORMACIÓN ACADÉMICA:</u></p> <p>Instrucción secundaria: Unidad Educativa “Ramón Barba Naranjo”</p> <p>Título obtenido: Bachiller en Electrónica de Consumo</p> <p>Instrucción de tercer nivel: Universidad Técnica de Cotopaxi</p> <p>Carrera: Ingeniería en Sistemas de Información</p>
--

Figura 87. Hoja de vida investigador 2