



UNIVERSIDAD LAICA “ELOY ALFARO” DE MANABÍ
EXTENSIÓN EN EL CARMEN
CARRERA DE INGENIERÍA EN INGENIERÍA DE SOFTWARE
Creada Ley No. 10 – Registro Oficial 313 de noviembre 13 de 1985

PROYECTO INTEGRADOR
PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN INGENIERÍA DE
SOFTWARE

Sistema Informático con DANN para el diagnóstico de enfermedades
fitosanitarias en cultivos de plátano del cantón El Carmen

AUTOR
ZAMBRANO QUIROZ JEAN FREDD
TUTOR
ING. CARLOS VINICIO LOPEZ RODRIGUEZ

EL CARMEN, 2026

CERTIFICACIÓN DEL TUTOR

CERTIFICACIÓN

En calidad de docente tutor de la Extensión El Carmen de la Universidad Laica "Eloy Alfaro" de Manabí, CERTIFICO:

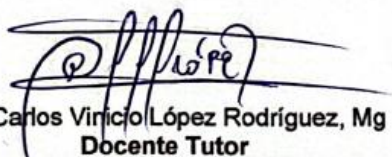
Haber dirigido, revisado y aprobado preliminarmente el Trabajo de Integración Curricular bajo la autoría del estudiante **JEAN FREDD ZAMBRANO QUIROZ**, legalmente matriculado/a en la carrera de Tecnologías de la Información período académico 2026-1, cumpliendo el total de 288 horas, cuyo tema del proyecto o núcleo problémico es **"SISTEMA INFORMÁTICO CON DANN PARA EL DIAGNÓSTICO DE ENFERMEDADES FITOSANITARIAS EN CULTIVOS DE PLÁTANO DEL CANTÓN EL CARMEN"**.

La presente investigación ha sido desarrollada en apego al cumplimiento de los requisitos académicos exigidos por el Reglamento de Régimen Académico y en concordancia con los lineamientos internos de la opción de titulación en mención, reuniendo y cumpliendo con los méritos académicos, científicos y formales, y la originalidad del mismo, requisitos suficientes para ser sometida a la evaluación del tribunal de titulación que designe la autoridad competente.

Particular que certifico para los fines consiguientes, salvo disposición de Ley en contrario.

El Carmen, 31 de julio de 2026.

Lo certifico,



Ing. Carlos Vinicio López Rodríguez, Mg
Docente Tutor
Área: Tecnologías de la Información

TRIBUNAL DE SUSTENTACIÓN

TRIBUNAL DE SUSTENTACIÓN

**Título del Trabajo de Titulación:**

Sistema informático con DANN para la evaluación automatizada del estado general de salud vegetal en cultivos de plátano del cantón El Carmen.

Modalidad:

Proyecto
Integrador

Autor:

Zambrano Quiroz Jean Fredd

Tutor:

Ing. López Rodríguez Carlos Vinicio.

Tribunal de Sustentación:

- Presidente: Ing. Mora Marcillo Alex Bladimir, Mg.
- Miembro: Ing. Mendoza Villamar Rocío Alexandra, Mg.
- Miembro: Ing. Arévalo Hermida Rómulo Danilo, Mg.

Fecha de Sustentación:

26 de agosto del 2026.

UNIVERSIDAD LAICA “ELOY ALFARO” DE MANABÍ
EXTENSIÓN EN EL CARMEN



DECLARACIÓN DE AUTORÍA

La responsabilidad del contenido de este Trabajo de titulación, cuyo tema es: Domain Adaptation (DANN) para mejorar el diagnóstico de enfermedades fitosanitarias en cultivos de plátano en la Granja Experimental ULEAM, extensión El Carmen, corresponde exclusivamente a: Zambrano Quiroz Jean Fredd con CI. 2300711260, y los derechos patrimoniales de la misma corresponden a la Universidad Laica “Eloy Alfaro” de Manabí.



Zambrano Quiroz Jean Fredd
C.I. 2300711260

DEDICATORIA

Dedico este trabajo de titulación, en primer lugar, a Dios, por haberme dado la vida, la salud, la sabiduría y la fortaleza necesarias para llegar hasta este momento tan importante de mi formación profesional.

Dedico este logro, de manera muy especial, a mi madre, quien ha sido padre y madre para mí, mi mayor ejemplo de amor, esfuerzo, sacrificio y valentía. Gracias por luchar incansablemente por mí, por darme todo sin esperar nada a cambio y por enseñarme con su ejemplo que los sueños se alcanzan con humildad, responsabilidad y perseverancia. Cada logro que hoy alcanzo lleva su nombre, porque detrás de este trabajo están sus desvelos, sus consejos, sus palabras de aliento y su amor incondicional.

A mi abuelita, por su cariño, su ternura y sus consejos, por estar presente en mi vida con ese amor sincero que solo una abuela puede brindar, y por ser también parte importante de mi crecimiento personal.

A mi novia, por su amor, paciencia, comprensión y apoyo durante este proceso; gracias por acompañarme en los días de cansancio, por motivarme cuando sentía que las fuerzas disminuían y por creer en mí incluso en los momentos en que yo dudaba.

A Brenda, mi amiga a quien considero como una hermana, por su cariño, apoyo sincero y por estar presente en esta etapa tan significativa.

A mi hermano Johan, a mi hermano de sangre Jhon Gilces y a mi prima Gabriela Muñoz, por formar parte de mi vida y de este camino, por su compañía, apoyo y por ser una motivación para seguir adelante.

Zambrano Quiroz Jean Fredd

AGRADECIMIENTO

Expreso mi más sincero agradecimiento a Dios, por permitirme culminar esta etapa académica y por darme la fortaleza necesaria para superar cada dificultad presentada durante el desarrollo de este trabajo de titulación.

A mi tutor designado, Ing. López Rodríguez Carlos Vinicio, por su orientación, tiempo, paciencia y apoyo académico durante el desarrollo de este proyecto. Sus conocimientos, observaciones y recomendaciones fueron importantes para mejorar este trabajo y culminarlo de la mejor manera.

A la Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen, y a la carrera de Ingeniería de Software, por brindarme la oportunidad de formarme profesionalmente y adquirir los conocimientos necesarios para mi desarrollo académico y personal.

A los docentes que fueron parte de mi formación, por compartir sus conocimientos y contribuir a mi crecimiento como futuro profesional.

Finalmente, agradezco a todas las personas que, de una u otra manera, me brindaron su apoyo, motivación y confianza durante este proceso. Cada palabra de aliento y cada gesto de ayuda fueron valiosos para alcanzar esta meta.

Zambrano Quiroz Jean Fredd

ÍNDICE GENERAL

CERTIFICACIÓN DEL TUTOR.....	I
TRIBUNAL DE SUSTENTACIÓN.....	II
DECLARACIÓN DE AUTORÍA.....	III
DEDICATORIA	IV
AGRADECIMIENTO	V
ÍNDICE GENERAL	VI
RESUMEN	XV
ABSTRACT.....	XVI
CAPÍTULO I	1
1 INTRODUCCIÓN	1
1.1 Introducción	1
1.2 Presentación del tema.....	1
1.3 Ubicación y contextualización de la problemática.....	2
1.4 Planteamiento del problema	3
1.4.1 Problematicación.....	3
1.4.2 Génesis del problema.....	4
1.4.3 Estado actual del problema	4
1.5 Diagrama causa – efecto del problema	5
1.6 Objetivos	5
1.6.1 Objetivo general.....	5
1.6.2 Objetivos específicos	6
1.7 Justificación.....	6
1.8 Impactos esperados	7
1.8.1 Impacto tecnológico.....	7
1.8.2 Impacto social	8
1.8.3 Impacto ecológico.....	8
CAPÍTULO II	9
2 MARCO TEÓRICO.....	9
2.1 Antecedentes históricos.....	9
2.2 Antecedentes investigativos	10
2.3 Definiciones conceptuales.....	12
2.3.1 Modelo de inteligencia artificial basado en ResNetVD34 y DANN	12
2.3.1.1 Aprendizaje profundo y redes neuronales convolucionales.....	12

2.3.1.2	Arquitectura ResNetVD34 y aprendizaje por transferencia	12
2.3.1.3	Adaptación de dominio DANN y capa de inversión de gradiente	13
2.3.2	Evaluación automatizada del estado general de salud vegetal.....	14
2.3.2.1	Características visuales de las hojas de plátano	14
2.3.2.2	Categorías de clasificación y umbral de confianza.....	14
2.3.2.3	Métricas de evaluación del modelo.....	15
2.3.3	Metodología de Desarrollo	15
2.3.3.1	Metodología Ágil Scrum para la Gestión del Software	15
2.4	Conclusiones del marco teórico	15
CAPÍTULO III.....		17
3	MARCO INVESTIGATIVO	17
3.1	Introducción	17
3.2	Tipos de investigación.....	17
3.2.1	Investigación aplicada.....	17
3.2.2	Investigación explicativa	17
3.2.3	Investigación de campo.....	18
3.3	Métodos de investigación.....	19
3.3.1	Método Analítico	19
3.3.2	Método deductivo	19
3.4	Fuentes de información de datos	19
3.4.1	Fuentes primarias	19
3.4.1.1	Encuesta	20
3.4.1.2	Entrevista	20
3.4.2	Fuentes secundarias	20
3.4.3	Población.....	21
3.4.4	Muestra censal	21
3.4.5	Análisis de las herramientas de recolección de datos a utilizar	22
3.4.5.1	Encuesta	22
3.4.5.2	Entrevista	22
3.4.5.3	Estructura de los instrumentos de recolección de datos aplicados	22
3.4.6	Plan de Recolección de datos.....	23
3.5	Análisis y presentación de resultados.....	23
3.5.1	Presentación y descripción de los resultados obtenidos	23

3.5.1.1	Análisis de encuesta realizada a estudiantes de la carrera Agropecuaria de la Uleam extensión El Carmen	23
3.5.1.2	Análisis de entrevista realizada al Docente de la Granja Experimental Uleam extensión El Carmen	28
3.5.2	Informe final del análisis de datos	30
CAPÍTULO IV		32
4	MARCO PROPOSITIVO	32
4.1	Introducción	32
4.2	Descripción de la propuesta	32
4.3	Determinación de recursos	33
4.3.1	Recursos Humanos.....	34
4.3.2	Recursos Tecnológicos	35
4.3.3	Recursos Económicos	37
4.4	Desarrollo del sistema bajo la metodología SCRUM	37
4.4.1	Descripción del producto	38
4.4.1.1	Propósito del producto	38
4.4.1.2	Funcionalidades claves	38
4.4.1.3	Usuarios objetivos.....	38
4.4.1.4	Condiciones de Éxito del Producto.....	39
4.4.2	Requerimientos del sistema	40
4.4.2.1	Historias de Usuarios	40
4.4.2.2	Requerimientos funcionales.....	44
4.4.2.3	Requerimientos no funcionales.....	45
4.4.3	Diseño del sistema	46
4.4.3.1	Casos de uso.....	46
4.4.3.2	Diagramas de secuencias	47
4.4.3.3	Diagramas de Estado.....	49
4.4.4	Arquitectura del sistema	52
4.4.4.1	Arquitectura general.....	52
4.4.4.2	Tecnologías Utilizadas.....	53
4.4.5	Roles Scrum	54
4.4.6	Planificación del Sprint.....	54
4.4.6.1	Sprint 0: Preparación del entorno y preprocesamiento del conjunto de imágenes.	54

4.4.6.2 Sprint 1: Desarrollo y entrenamiento del modelo	56
4.4.6.3 Sprint 2: Evaluación y conversión del modelo	57
4.4.6.4 Sprint 3: Integración móvil y pruebas finales	58
4.4.7 Product Backlog.....	59
4.4.7.1 Backlog inicial	59
4.4.7.2 Sprint Backlog	61
4.4.8 Desarrollo de los sprints.....	63
4.4.8.1 Desarrollo del Sprint 0: Preparación del entorno y dataset.....	64
4.4.8.2 Desarrollo del Sprint 1: Entrenamiento inicial del modelo	67
4.4.8.3 Desarrollo del Sprint 2: Verificación y perfeccionamiento del modelo	70
4.4.8.4 Desarrollo del Sprint 3: Integración y validación del modelo	72
4.4.9 Resultado técnico del modelo entrenado	76
4.4.10 Integración del modelo TensorFlow Lite en Flutter	78
4.4.11 Pruebas funcionales del módulo móvil	79
CAPÍTULO V	81
5 EVALUACIÓN DE RESULTADOS	81
5.1 Introducción	81
5.2 Presentación de los resultados.....	81
5.2.1 Resultados del entrenamiento y evaluación del modelo	82
5.2.2 Interpretación objetiva	83
5.2.3 Resultados de las pruebas funcionales	84
5.3 Análisis e interpretación de los resultados	86
5.4 Cumplimiento de los objetivos de investigación	87
5.5 Limitaciones identificadas.....	88
5.6 Síntesis del capítulo.....	90
CAPÍTULO VI.....	91
6 CONCLUSIONES Y RECOMENDACIONES	91
6.1 Conclusiones	91
6.2 Recomendaciones.....	91
7 REFERENCIAS.....	93
8 ANEXOS	96
8.1 Anexo A: Evidencias técnicas del desarrollo del módulo.....	96
8.2 Anexo B: Evidencias de la entrevista y la encuesta	99
8.3 Anexo C: Asignación de tutor	102

8.4	Anexo D: Reporte de similitud.....	103
9	GLOSARIO	105

ÍNDICE DE TABLAS

Tabla 1 Plan de Recolección de datos.....	23
Tabla 2 Análisis y presentación de resultados	24
Tabla 3 Análisis de entrevista	28
Tabla 4 Recursos Humanos	34
Tabla 5 Recursos Tecnológicos	35
Tabla 6 Recursos Económicos	37
Tabla 7 Usuarios objetivos.....	38
Tabla 8 Historia de Usuario 1: Carga y organización del conjunto de imágenes.....	40
Tabla 9 Historia de Usuario 2: Procesamiento del conjunto de imágenes	41
Tabla 10 Historia de Usuario 3: Entrenamiento del modelo ResNetVD34 con DANN.....	41
Tabla 11 Historia de Usuario 4: Evaluación del modelo entrenado	42
Tabla 12 Historia de Usuario 5: Conversión y exportación del modelo.....	42
Tabla 13 Historia de Usuario 6: Integración del modelo en la aplicación móvil	43
Tabla 14 Historia de Usuario 7: Presentación del resultado e historial local	44
Tabla 15 Tecnologías Utilizadas.....	53
Tabla 16 Roles Scrum.....	54
Tabla 17 Sprint 0: Preparación del entorno y preprocesamiento del conjunto de imágenes. ..	54
Tabla 18 Sprint 1: Desarrollo y entrenamiento del modelo	56
Tabla 19 Sprint 2: Evaluación y conversión del modelo	57
Tabla 20 Sprint 3 Integración móvil y pruebas finales	58
Tabla 21 Backlog Inicial	59
Tabla 22 Sprint Backlog: Preparación del entorno y del conjunto de imágenes	61
Tabla 23 Sprint Backlog: Desarrollo y entrenamiento del modelo.....	62
Tabla 24 Sprint Backlog: Evaluación y conversión del modelo.....	62
Tabla 25 Sprint Backlog: Integración móvil y pruebas finales.....	63
Tabla 26 Métricas obtenidas en la evaluación final del modelo	76
Tabla 27 Pruebas funcionales del módulo móvil.....	79
Tabla 28 Resultados de las pruebas funcionales.....	85

ÍNDICE DE GRÁFICOS E ILUSTRACIONES

Ilustración 1 diagrama causa - efecto del problema.....	5
Ilustración 2 Preparación del conjunto de imágenes.....	46
Ilustración 3 Entrenamiento y evaluación del modelo.....	46
Ilustración 4 Análisis de una imagen y presentación de resultados.....	47
Ilustración 5 Diagrama de secuencia Preparación del conjunto de imágenes	47
Ilustración 6 Diagrama de secuencia: Entrenamiento, evaluación y conversión del modelo ..	48
Ilustración 7 Diagrama de secuencia: Análisis de imagen en la aplicación móvil	48
Ilustración 8 diagrama de estado: Preparación del conjunto de imágenes.....	49
Ilustración 9 Diagrama de estado: Entrenamiento y evaluación del modelo.....	50
Ilustración 10 Diagrama de estado: Análisis de imagen y presentación de resultados.....	51
Ilustración 11 Arquitectura general del módulo de Inteligencia Artificial	52
Ilustración 12 Configuración del entorno virtual y verificación de las dependencias instaladas	64
Ilustración 13 Estructura final de carpetas de los conjuntos de imágenes.....	65
Ilustración 14 Muestra de las imágenes del dominio fuente utilizadas en la clasificación.....	65
Ilustración 15 Verificación de la distribución de los conjuntos de imágenes.....	66
Ilustración 16 Implementación del extractor de características ResNet34	68
Ilustración 17 Implementación del modelo ResNet34 con adaptación de dominio DANN	68
Ilustración 18 Implementación de la capa de inversión de gradiente	69
Ilustración 19 Registro de ejecución del entrenamiento inicial de ResNetVD34 con DANN	70
Ilustración 20 captura de las métricas	71
Ilustración 21 matriz de confusión.....	71
Ilustración 22 Resultados de la evaluación y conversión del modelo a TensorFlow Lite	72
Ilustración 23 Configuración del modelo TensorFlow Lite en el proyecto Flutter	73
Ilustración 24 Presentación del resultado del análisis y del historial de la aplicación móvil ..	74
Ilustración 25 Ejecución de la aplicación móvil en modo sin conexión en el dispositivo de prueba.....	75
Ilustración 26 Publicación del código fuente y del modelo en GitHub	75
Ilustración 27 Generación del APK de la aplicación.....	79
Ilustración 29 Funcionalidad de la app móvil.....	80
Ilustración 30 Organización de las imágenes en saludable y salud comprometida.	96
Ilustración 31 Entrenamiento de ResNetVD34 con DANN.	96
Ilustración 32 Pantalla principal de la aplicación Flutter.....	97

Ilustración 33 Resultado de una hoja saludable aplicación Flutter	97
Ilustración 34 Resultado de una hoja con salud comprometida aplicación Flutter.....	98
Ilustración 35 Resultado de la aplicación funcional en un dispositivo Android.....	98
Ilustración 36 Evidencia de la entrevista semiestructurada aplicada al docente especialista de la Granja Experimental ULEAM El Carmen.	99
Ilustración 37 Evidencia de la encuesta aplicada a los estudiantes de agropecuaria de la Granja Experimental ULEAM El Carmen.....	99
Ilustración 38 Estructura de la encuesta realizada a los estudiantes de agropecuaria de la Granja Experimental ULEAM El Carmen.....	100
Ilustración 39 Estructura de la entrevista semiestructurada aplicada al docente especialista de la Granja Experimental ULEAM El Carmen	102
Ilustración 40 Asignación de tutor	102
Ilustración 41 Reporte de similitud.....	103

ÍNDICE DE ANEXOS

8.1	Anexo A: Evidencias técnicas del desarrollo del módulo	96
8.2	Anexo B: Evidencias de la entrevista y la encuesta.....	99
8.3	Anexo C: Asignación de tutor.....	102
8.4	Anexo D: Reporte de similitud	103

RESUMEN

El presente proyecto tuvo como objetivo desarrollar un módulo móvil basado en inteligencia artificial para apoyar la evaluación del estado general de salud vegetal en hojas de plátano, bajo condiciones reales de campo en la Granja Experimental ULEAM, Extensión El Carmen. La problemática identificada se relacionó con la variabilidad de la evaluación visual realizada en campo y con las diferencias existentes entre las imágenes de conjuntos de datos públicos y las fotografías capturadas directamente en el entorno agrícola, debido a factores como iluminación, fondo, sombras, distancia, calidad de la cámara y condiciones ambientales. Esta situación puede afectar el desempeño de los modelos de visión artificial y limitar la confiabilidad de los resultados.

La investigación consideró como población a 25 estudiantes de la carrera de Agropecuaria vinculados con actividades prácticas en la Granja Experimental ULEAM, Extensión El Carmen. Debido al tamaño reducido y accesible de la población, se trabajó con una muestra censal que incluyó a la totalidad de los 25 estudiantes. Metodológicamente, se aplicó una investigación de campo, documental y aplicada, con enfoque cuantitativo y apoyo descriptivo. Para la recolección de información se utilizaron encuestas dirigidas a los estudiantes y una entrevista semiestructurada aplicada a un docente del área agropecuaria. El desarrollo del módulo se organizó mediante la metodología ágil Scrum, permitiendo planificar las actividades de preparación de datos, entrenamiento del modelo, evaluación e integración móvil.

Para el modelo se utilizó la arquitectura ResNetVD34, preentrenada con ImageNet, junto con una Red Neuronal Adversaria para Adaptación de Dominio, conocida como DANN, con el fin de reducir la brecha entre el dominio fuente y el dominio objetivo. El conjunto de datos público fue depurado, balanceado y organizado en dos clases: hojas saludables y hojas con salud comprometida. Además, se emplearon 64 fotografías reales tomadas en campo como dominio objetivo. La evaluación final se realizó con 12 imágenes balanceadas, obteniendo una exactitud, precisión, recall y medida F1 del 83,33 %, con una confianza promedio del 86,91 %. Posteriormente, el modelo fue convertido a TensorFlow Lite mediante optimización float16 e integrado en una aplicación móvil desarrollada en Flutter, donde se verificó el uso de cámara y galería, la inferencia sin conexión, la visualización de probabilidades, el umbral de confianza del 65 % y el almacenamiento local del historial.

Palabras clave: inteligencia artificial, adaptación de dominio, ResNetVD34, salud vegetal, TensorFlow Lite.

ABSTRACT

This project aimed to develop a mobile module based on artificial intelligence to support the assessment of the general plant health status of plantain leaves under real field conditions at the ULEAM Experimental Farm, El Carmen Extension. The identified problem was related to the variability of visual field assessment and to the differences between images from public datasets and photographs captured directly in the agricultural environment, due to factors such as lighting, background, shadows, distance, camera quality, and environmental conditions. This situation may affect the performance of computer vision models and limit the reliability of the results.

The study population consisted of 25 Agricultural Sciences students involved in practical activities at the ULEAM Experimental Farm, El Carmen Extension. Due to the small and accessible population size, a census sample including all 25 students was used. Methodologically, field, documentary, and applied research was conducted, with a quantitative and descriptive approach. Data collection was carried out through surveys administered to the students and a semi-structured interview with an agricultural area teacher. The module development was organized using the agile Scrum methodology, which allowed planning the activities related to data preparation, model training, evaluation, and mobile integration.

The model used the ResNetVD34 architecture, pretrained on ImageNet, together with a Domain-Adversarial Neural Network, known as DANN, to reduce the gap between the source and target domains. The public dataset was cleaned, balanced, and organized into two classes: healthy leaves and leaves with compromised health. In addition, 64 real field photographs were used as the target domain. The final evaluation was performed with 12 balanced images, achieving 83.33% accuracy, precision, recall, and F1-score, with an average confidence of 86.91%. The model was then converted to TensorFlow Lite using float16 optimization and integrated into a mobile application developed in Flutter, where camera and gallery use, offline inference, probability display, the 65% confidence threshold, and local storage of the evaluation history were verified.

Keywords: artificial intelligence, domain adaptation, ResNetVD34, plant health, TensorFlow Lite.

CAPÍTULO I

INTRODUCCIÓN

1.1 Introducción

El cultivo de plátano represento una actividad agrícola de importancia para el cantón El Carmen y para la provincia de Manabí, debido a su aporte productivo y a la participación de pequeños productores, estudiantes y técnicos vinculados al sector agropecuario. Sin embargo, las condiciones propias de las zonas de cultivo, como la humedad, la temperatura, la iluminación variable y la presencia de factores fitosanitarios, podían afectar el estado visual de las hojas y dificultar una evaluación oportuna y uniforme.

Durante las actividades de campo, las hojas del plátano eran revisadas mediante la observación directa. Esta práctica permitía detectar manchas, variaciones de color, daños físicos o pérdida de vigor, sin embargo, los resultados podían variar según la experiencia del observador y las condiciones presentes en el lugar. Por esta razón, se consideró conveniente contar con una herramienta tecnológica que complementara esta tarea y facilitara una primera valoración del estado general de la planta en el propio cultivo.

Una alternativa para apoyar esta revisión fue el uso de inteligencia artificial aplicada al análisis de fotografías, mediante técnicas de visión por computador y aprendizaje profundo se pueden procesar imágenes de las hojas y se reconocieron patrones relacionados con su condición visual. El inconveniente se presentó cuando el modelo ha sido entrenado con fotografías públicas, debido a que estas diferían de las capturadas en la Granja Experimental por aspectos como iluminación, el fondo, el ángulo o la calidad de la cámara. Para afrontar estas diferencias, se empleó la adaptación de dominio, con el propósito de ajustar el aprendizaje del modelo a las condiciones reales del campo.

Por esta razón, en el presente capítulo se expusieron el contexto inicial de la investigación, la problemática identificada en la Granja Experimental ULEAM, Extensión El Carmen, los objetivos que orientaron el desarrollo del módulo móvil, la justificación del estudio y los impactos esperados. De esta manera, se estableció la base investigativa para el desarrollo de una solución basada en ResNetVD34 y DANN, integrada en una aplicación móvil capaz de apoyar la evaluación automatizada del estado general de salud vegetal de las hojas de plátano, sin reemplazar el criterio de un profesional agrícola.

1.2 Presentación del tema

En el presente trabajo desarrollo un módulo móvil de inteligencia artificial orientado a la evaluación automatizada del estado general de salud vegetal de hojas de plátano en condiciones reales de campo del cantón El Carmen. Para la solución se utilizó una arquitectura

basada en ResNetVD34, inicializada con pesos preentrenados de ImageNet, como extractor de características visuales, conjuntamente con una Red Neuronal Adversaria para Adaptación de Dominio (DANN), con la finalidad de reducir las diferencias existentes entre las imágenes procedentes de conjuntos de datos públicos y las fotografías capturadas en el entorno agrícola local.

Las imágenes se organizaron en dos categorías: hoja saludable y hoja con salud comprometida. Si la probabilidad más alta no alcanza el umbral de confianza del 65 %, la aplicación mostraba el resultado no determinado. De esta manera, no se asignaba una categoría cuando la información visual de la fotografía resultaba insuficiente. Después del entrenamiento y la evaluación, el modelo se convirtió a TensorFlow Lite con optimización float16 y se incorporó a una aplicación desarrollada en Flutter. Esto permito procesar las imágenes directamente en el dispositivo móvil, incluso cuando existía conexión a Internet.

Desde la aplicación, se permitió al usuario tomar una fotografía con la cámara o elegir una imagen desde la galería, luego del análisis, se mostraban las probabilidades calculadas y algunas recomendaciones generales. Además, cada resultado quedaba almacenado localmente para que pudiera consultarse posteriormente. El módulo funciona como un componente independiente que podía integrarse al sistema informático principal destinado a la capacitación agrícola.

Mediante el módulo únicamente se realizó una evaluación general de la apariencia de la hoja, en consecuencia. El resultado no representaba la identificación de una enfermedad fitosanitaria específica ni determina su nivel de severidad. Tampoco se pretendió reemplazar la revisión efectuada por un profesional del área agrícola.

1.3 Ubicación y contextualización de la problemática

La presente investigación se contextualiza en la Granja Experimental de la Universidad Laica Eloy Alfaro de Manabí (ULEAM), extensión El Carmen. Este cantón representa un punto estratégico para el sector agroproductivo de la provincia de Manabí, zona que concentra el 51.4% del volumen total de la producción platanera a nivel nacional. La referida estación experimental opera como un centro de validación técnica y desarrollo tecnológico, cuyas condiciones edafoclimáticas y variabilidad fenotípica en el cultivo de plátano (*Musa AAB*) ofrecen un entorno óptimo para el entrenamiento, prueba y validación de modelos de visión artificial orientados al diagnóstico fitosanitario bajo escenarios reales de producción.

A pesar de este potencial productivo, las plantaciones de plátano (*Musa AAB*) en el cantón El Carmen enfrentan severas amenazas fitosanitarias debido a las condiciones de alta

humedad y temperatura características de la zona, factores que propician la propagación acelerada de patógenos como la Sigatoka Negra **Mycosphaerella fijiensis**.

La disponibilidad limitada de herramientas para detectar estos problemas de manera temprana en las zonas rurales cercanas a la granja reduce la capacidad de reacción de los pequeños productores, en muchos casos, la revisión de las plantas se basa en inspecciones visuales y en la experiencia de cada productor, por lo que los resultados pueden variar, esta situación respalda la selección de la Granja Experimental ULEAM como lugar de estudio. Sus cultivos se utilizaron como dominio objetivo para recopilar imágenes y evaluar un sistema informático ajustado a las condiciones agrícolas locales.

1.4 Planteamiento del problema

1.4.1 Problematicación

La evaluación visual del estado de salud de las hojas de plátano constituye una actividad importante para el monitoreo del cultivo, debido a que las alteraciones de color, textura y estructura foliar pueden indicar que la planta presenta algún tipo de afectación. En la Granja Experimental ULEAM, Extensión El Carmen, este proceso depende principalmente de la observación directa de estudiantes, técnicos o responsables del cultivo, por lo que los resultados pueden variar de acuerdo con la experiencia del observador, las condiciones ambientales y la calidad de la inspección realizada.

El empleo de sistemas de inteligencia artificial representa una alternativa para apoyar este proceso; sin embargo, los modelos de visión artificial entrenados con imágenes obtenidas de conjuntos de datos públicos pueden presentar dificultades cuando son utilizados con fotografías tomadas directamente en el campo. Las diferencias de iluminación, orientación de las hojas, fondos naturales, sombras, resolución y dispositivos de captura producen una variación entre el dominio de entrenamiento y el dominio donde finalmente se utiliza el modelo.

Esta diferencia, conocida como desplazamiento de dominio, puede reducir la capacidad del modelo para distinguir correctamente entre una hoja saludable y una hoja con alteraciones visuales. Además, la conectividad limitada o inestable en determinados entornos agrícolas dificulta el uso de soluciones que dependan permanentemente de un servidor externo.

¿De qué manera un módulo móvil de inteligencia artificial basado en ResNetVD34 y adaptación de dominio DANN puede contribuir a la evaluación automatizada del estado general de salud vegetal de hojas de plátano en las condiciones reales de campo de la Granja Experimental ULEAM, Extensión El Carmen?

1.4.2 Génesis del problema

En el trabajo de campo las hojas de plátano suelen revisarse a simple vista, durante esta actividad se observan manchas, amarillamiento, pérdida de color, cambios en la textura y daños físicos, sin embargo una misma señal puede estar relacionadas con distintas causas por lo que su valoración no siempre resulta sencilla, además el criterio puede variar según la experiencia de la persona encargada de la revisión, especialmente cuando no se dispone del acompañamiento de un profesional del área.

La diferencia entre las imágenes públicas empleadas durante para el entrenamiento y las fotografías tomadas en el entorno local produce una brecha de dominio que pueden reducir el desempeño de los modelos convencionales, ante este problema, se planteó desarrollar un modelo basado en aprendizaje por transferencia y adaptación de dominio, utilizando imágenes del dominio fuente y fotografías del dominio objetivo, con la finalidad de reconocer características visuales presentes en ambos contextos.

Asimismo, la necesidad de utilizar la herramienta directamente en las áreas de cultivo motivó el desarrollo de una solución móvil capaz de ejecutar el modelo de inteligencia artificial sin depender de una conexión permanente a Internet.

1.4.3 Estado actual del problema

Antes del desarrollo del presente módulo, no se identificó en la Granja Experimental ULEAM, Extensión El Carmen, una herramienta móvil de inteligencia artificial orientada específicamente a la evaluación automatizada del estado general de salud de las hojas de plátano. El monitoreo se realizaba principalmente mediante observación directa, sin un sistema que permitiera capturar una fotografía, analizarla automáticamente y conservar un historial local de los resultados.

Aunque existen modelos de visión artificial aplicados a la agricultura, su implementación directa en el entorno local presenta dificultades debido a las diferencias existentes entre las imágenes utilizadas para el entrenamiento y las fotografías obtenidas en condiciones reales de campo. Entre estas variaciones se encuentran los cambios de iluminación, la presencia de fondos complejos y sombras, el movimiento de las hojas, el ángulo de la cámara y las diferencias entre los dispositivos móviles utilizados

Asimismo, las soluciones que analizan imágenes mediante servicios alojados en la nube necesitan conexión para enviar la fotografía y obtener el resultado. Esta condición dificulta su uso en lugares donde el acceso a Internet es inestable. Por otra parte, implementar modelos de aprendizaje profundo directamente en teléfonos móviles exige reducir su tamaño y el consumo

de recursos, con el fin de evitar demoras en el procesamiento o problemas de compatibilidad con equipos de gama media.

En consecuencia, se identificó la necesidad de desarrollar un módulo que adaptara el aprendizaje obtenido a partir de imágenes públicas a las condiciones visuales locales y que, al mismo tiempo, pudiera ejecutarse directamente en un dispositivo móvil. La solución debía permitir clasificar el estado visual de las hojas en las categorías saludable y salud comprometida, presentar una clasificación no determinada cuando la confianza fuera insuficiente.

1.5 Diagrama causa – efecto del problema

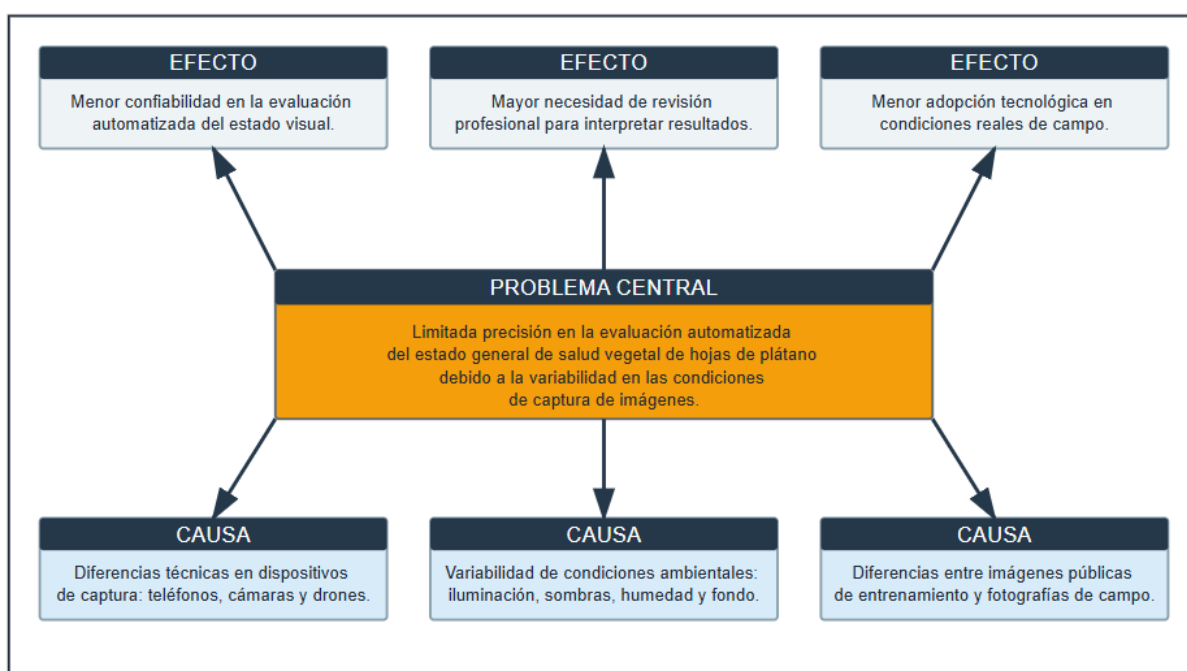


Ilustración 1 diagrama causa - efecto del problema

1.6 Objetivos

1.6.1 Objetivo general

Desarrollar e implementar un módulo móvil de inteligencia artificial basado en la arquitectura ResNetVD34 y la adaptación de dominio DANN para evaluar automáticamente el estado general de salud vegetal de hojas de plátano capturadas en condiciones reales de campo en la Granja Experimental ULEAM, Extensión El Carmen.

1.6.2 Objetivos específicos

Examinar las necesidades tecnológicas y definir los requerimientos funcionales para la evaluación del estado general de salud vegetal de las hojas de plátano en la Granja Experimental ULEAM, Extensión El Carmen

Establecer el sustento teórico y científico del módulo a partir de los principios de las redes neuronales convolucionales, ResNetVD34, el aprendizaje por transferencia, la adaptación de dominio mediante DANN y la inferencia móvil mediante TensorFlow Lite.

Preparar las imágenes de los dominios fuente y objetivo mediante su revisión, depuración, etiquetado, balanceo, para después distribuirlas en conjuntos de entrenamiento, validación y prueba.

Construir y entrenar un modelo que combine ResNetVD34 con DANN para clasificar las hojas de plátano como saludable o salud comprometida, y evaluar su desempeño mediante exactitud, precisión, recall, F1-score y matriz de confusión.

Implementar el modelo entrenado en una aplicación móvil desarrollada en Flutter, con funcionamiento offline, captura mediante cámara, selección desde la galería, presentación de probabilidades y almacenamiento local del historial de análisis.

1.7 Justificación

La evaluación visual del estado general de las hojas de plátano constituye una actividad importante para el seguimiento del cultivo, debido a que los cambios de coloración, textura y estructura pueden evidenciar posibles alteraciones en la planta. Aun así, esta revisión depende en gran medida de la observación humana, en consecuencia, el resultado puede cambiar según la experiencia de quien la realiza, las condiciones del entorno y la calidad de la imagen examinada.

La inteligencia artificial ofrece una opción tecnológica para complementar esta tarea por medio del análisis automático de fotografías. Sin embargo, los modelos entrenados solo con colecciones públicas pueden mostrar un menor desempeño al analizar fotografías tomadas en el campo, esto se relaciona con los cambios de iluminación, fondo, orientación, distancia y dispositivo utilizado para realizar la captura.

Ante esta situación, se consideró necesario desarrollar un modelo que integrara ResNetVD34 y adaptación de dominio DANN, su finalidad fue disminuir las diferencias entre las imágenes públicas del dominio fuente y las fotografías reales del dominio objetivo. Con esta estrategia se obtuvo una solución para clasificar las hojas en las categorías saludable y salud comprometida.

Al incorporar el modelo en una aplicación móvil mediante TensorFlow Lite, fue posible analizar fotografías en el propio dispositivo, sin utilizar una conexión a Internet ni de servidores externos. Esta característica resulta relevante para entornos agrícolas donde la conectividad puede ser limitada o inestable.

Desde el ámbito académico y tecnológico, el proyecto permitió aplicar conocimientos de inteligencia artificial, visión por computador, adaptación de dominio y desarrollo móvil. Además, el módulo quedó preparado para su incorporación dentro de la aplicación principal del Proyecto Integrador.

El sistema desarrollado constituye una herramienta de apoyo para la evaluación inicial del estado visual de las hojas. Sus resultados tienen carácter orientativo, debido a que no identifica enfermedades específicas, no calcula niveles de severidad y no reemplaza la valoración de un profesional agrícola.

1.8 Impactos esperados

El desarrollo e implementación del módulo móvil de inteligencia artificial para la evaluación del estado general de salud vegetal de las hojas de plátano puede generar aportes en los ámbitos tecnológico, social y ecológico. Estos impactos se plantean como resultados potenciales derivados del uso de la solución y no como efectos económicos o ambientales medidos directamente durante la investigación.

1.8.1 Impacto tecnológico

El principal impacto tecnológico consiste en la implementación de un modelo de inteligencia artificial adaptado para procesar fotografías capturadas en condiciones reales de campo. La utilización de ResNetVD34 preentrenada, junto con la estrategia DANN, permite abordar las diferencias visuales existentes entre los conjuntos de imágenes públicas y las fotografías obtenidas en la Granja Experimental ULEAM, Extensión El Carmen.

La integración realizada en Flutter permitió comprobar que un modelo de aprendizaje profundo puede incorporarse a una aplicación móvil y trabajar junto con funciones habituales del teléfono, entre ellas se encuentran el acceso a la cámara y la galería, la presentación de los resultados y el registro local de cada análisis.

El trabajo también deja una base técnica que puede reutilizarse para futuras mejoras, entre ellas se encuentra la incorporación de nuevos conjuntos de imágenes, el aumento de la precisión del modelo y su conexión con otros módulos especializados del sistema informático principal.

1.8.2 Impacto social

El impacto social previsto parte de brindar acceso a una herramienta digital de apoyo para la observación del estado visual de las hojas de plátano. La aplicación puede ser utilizada por estudiantes, técnicos y demás personas que participan en el monitoreo de la Granja Experimental, quienes podrán obtener una valoración inicial de forma rápida y fácil de comprender.

Al funcionar sin conexión, la herramienta puede utilizarse en zonas donde el servicio de Internet es limitado o inestable. Su uso también resulta sencillo, ya que permite tomar una fotografía con la cámara o escoger imágenes almacenadas en un teléfono de uso cotidiano.

Los resultados también pueden aprovecharse en los procesos de capacitación agrícola, ya que permiten practicar el uso responsable de herramientas de inteligencia artificial al análisis de las hojas, aun así, la respuesta del módulo debe tomarse como una orientación inicial y no sustituye la revisión ni el criterio de un profesional del área.

1.8.3 Impacto ecológico

El aporte ecológico del módulo se presenta de forma indirecta, principalmente al facilitar un seguimiento periódico de la apariencia de las plantas, cuando se detecta una hoja con alteraciones puede motivar una revisión técnica más oportuna para evitar que se tomen decisiones basadas únicamente en observaciones poco consistentes.

El módulo no indica qué tratamiento fitosanitario debe aplicarse ni calcula la cantidad de productos agrícolas que se necesita utilizar, por esta razón sus resultados no permiten afirmar que reducirá directamente el uso de agroquímicos, su utilidad se encuentra en ofrecer una primera referencia sobre la apariencia de las hojas y conserva cada análisis en un historial, esta información puede facilitar el seguimiento del cultivo y servir de apoyo al personal especializado antes de tomar una decisión.

De igual manera, el análisis realizado en el dispositivo evita enviar constantemente las imágenes a servidores externos, lo que reduce la necesidad de utilizar infraestructura en la nube durante el uso habitual del módulo. Su aporte ambiental debe entenderse como potencial y complementario, sujeto a futuras evaluaciones en condiciones reales de aplicación.

CAPÍTULO II

MARCO TEÓRICO

1.9 Antecedentes históricos

La aplicación de la inteligencia artificial en la agricultura ha evolucionado desde los primeros sistemas expertos basados en reglas hasta los actuales modelos de aprendizaje profundo. Soria Olivas et al. (2022) explican que estos modelos permiten aprender representaciones complejas directamente a partir de los datos. En el análisis de imágenes de hojas, las redes neuronales convolucionales permiten extraer automáticamente patrones visuales como bordes, colores, texturas y formas (Tugrul et al., 2022). Estas capacidades han favorecido el desarrollo de sistemas para la detección y clasificación automatizada de enfermedades vegetales mediante fotografías digitales (Chen et al., 2020).

Raschka et al. (2023) señalan que el aprendizaje por transferencia facilita el uso de modelos previamente entrenados con grandes conjuntos de imágenes, reduciendo la necesidad de desarrollar una red completamente desde cero. Esta estrategia permite aprovechar características visuales aprendidas previamente y adaptarlas al reconocimiento de enfermedades vegetales (Chen et al., 2020). Sin embargo, cuando estos modelos se utilizan con fotografías capturadas directamente en el campo, pueden presentarse diferencias de iluminación, fondo, orientación, resolución y calidad respecto a las imágenes empleadas durante el entrenamiento. Esta situación se conoce como desplazamiento de dominio (Liu et al., 2022). Estas variaciones también han sido identificadas en investigaciones que utilizaron imágenes agrícolas reales para reconocer enfermedades en plantas de plátano (Selvaraj et al., 2020).

De acuerdo con Liu et al. (2022), las técnicas de adaptación de dominio, como DANN, buscan obtener características útiles para la clasificación y menos dependientes del origen de las imágenes. Además, herramientas como TensorFlow Lite permiten optimizar y ejecutar modelos de inteligencia artificial directamente en dispositivos móviles (Muneeb et al., 2022). El uso de modelos móviles para detectar enfermedades en plantas de plátano constituye una alternativa viable para entornos agrícolas con acceso limitado a Internet (Sanga et al., 2020).

En el presente proyecto se utilizó un modelo ResNetVD34 preentrenado con ImageNet, combinado con adaptación de dominio DANN, siguiendo los fundamentos de las mejoras ResNet-D y de la adaptación adversaria de dominio, posteriormente, el modelo fue convertido al formato TensorFlow Lite con optimización float16 para su ejecución offline dentro de una aplicación desarrollada en Flutter, una estrategia coherente con la ejecución de modelos de visión en dispositivos móviles (Muneeb et al., 2022).

1.10 Antecedentes investigativos

En los últimos años, diferentes investigaciones han aplicado redes neuronales convolucionales y aprendizaje por transferencia al análisis de imágenes de hojas. Tugrul et al. (2022), mediante una revisión de estudios sobre detección visual en plantas, identificaron que las arquitecturas convolucionales permiten aprender características relacionadas con el color, la textura, los bordes y la forma de las hojas. Estos elementos visuales pueden utilizarse para clasificar diferentes condiciones presentes en los cultivos. De manera similar, los modelos de aprendizaje profundo han demostrado capacidad para identificar enfermedades vegetales a partir de características aprendidas directamente de las imágenes (Chen et al., 2020). Asimismo, se han propuesto procedimientos que combinan la segmentación y la clasificación para reconocer automáticamente las afectaciones visibles en las hojas de plátano (Krishnan et al., 2022).

Como antecedente específico del cultivo de plátano, Sanga et al. (2020) evaluaron modelos convolucionales preentrenados ResNet152 e InceptionV3 mediante imágenes de hojas de plátano. Los modelos alcanzaron exactitudes del 99,2 % y 95,41 %, respectivamente. Debido a su menor requerimiento de memoria, InceptionV3 fue seleccionado para implementarse en una aplicación móvil con funcionamiento offline. Aunque dicho estudio se orientó a condiciones específicas de las hojas, constituye un antecedente relevante porque demuestra la aplicación del aprendizaje por transferencia y la ejecución móvil de modelos de inteligencia artificial en el cultivo de plátano.

Otras investigaciones también han demostrado la utilidad del aprendizaje profundo para reconocer síntomas visibles de enfermedades y plagas del plátano. Selvaraj et al. (2020) desarrollaron modelos basados en redes neuronales convolucionales y aprendizaje por transferencia utilizando imágenes obtenidas en condiciones reales de campo. Más recientemente, se evaluaron arquitecturas preentrenadas como EfficientNetB0, ResNet50 y VGG19 para la detección de Sigatoka negra y Cordana en hojas de plátano (Jiménez et al., 2025). Estos estudios respaldan la aplicación de modelos previamente entrenados para el análisis automatizado de las condiciones fitosanitarias del cultivo.

No obstante, los modelos entrenados con imágenes obtenidas bajo determinadas condiciones pueden disminuir su rendimiento cuando son aplicados a fotografías procedentes de un entorno diferente. Las variaciones de iluminación, fondo, orientación, sombras, dispositivos de captura y calidad de imagen generan diferencias entre el dominio utilizado durante el entrenamiento y el entorno real de aplicación. Esta dificultad es conocida como desplazamiento de dominio (Liu et al., 2022).

Para enfrentar esta problemática, se han incorporado técnicas de aprendizaje por transferencia, ajuste fino y adaptación de dominio. De acuerdo con Liu et al. (2022), la adaptación de dominio no supervisada utiliza datos etiquetados del dominio fuente y datos del dominio objetivo para buscar un mejor desempeño en entornos cuyas distribuciones son diferentes. Entre estos enfoques se encuentran los métodos adversarios, orientados a aprender características menos dependientes del origen de las imágenes.

El uso de modelos de inteligencia artificial en aplicaciones móviles también ha aumentado. Muneeb et al. (2022) plantearon un procedimiento para trasladar modelos de clasificación de imágenes a teléfonos móviles y emplearon TensorFlow Lite junto con Flutter para evaluarlos en diferentes conjuntos de datos, entre ellos imágenes de hojas. De esta manera, el procesamiento puede efectuarse en el propio dispositivo, aunque deben tomarse en cuenta aspectos como la memoria disponible, la capacidad de procesamiento y el desempeño del hardware móvil. En esta misma línea, se desarrolló BananaSqueezeNet como una arquitectura convolucional ligera y de rápida ejecución para reconocer enfermedades visibles en hojas de plátano (Bhuiyan et al., 2023).

En relación con los conjuntos de datos de hojas de plátano, Arman et al. (2023) presentaron BananaLSD, un conjunto de imágenes organizado para la clasificación de Sigatoka, Cordana y Pestalotiopsis mediante técnicas de aprendizaje automático. Este recurso constituye un antecedente importante para la preparación de conjuntos de imágenes y la comparación de modelos destinados al reconocimiento de enfermedades foliares.

Por otra parte, Sunitha et al. (2023) publicaron un conjunto de imágenes etiquetadas que contiene hojas saludables y hojas con diferentes deficiencias nutricionales. Las fotografías fueron capturadas mediante dispositivos móviles en distintas condiciones ambientales y organizadas para apoyar el entrenamiento y la evaluación de modelos de aprendizaje automático y aprendizaje profundo. Este conjunto constituyó una fuente relevante para investigaciones orientadas al análisis visual de hojas de plátano.

Estos antecedentes respaldan el desarrollo del presente módulo, en el cual se utilizó ResNetVD34 preentrenado con ImageNet y adaptación de dominio DANN para clasificar las hojas de plátano en las categorías saludable y salud comprometida, sobre la base de las mejoras ResNet-D y de los principios de adaptación de dominio, posteriormente, el modelo fue integrado en una aplicación Flutter mediante TensorFlow Lite con optimización float16, permitiendo realizar el análisis de manera offline (Muneeb et al., 2022).

1.11 Definiciones conceptuales

1.11.1 Modelo de inteligencia artificial basado en ResNetVD34 y DANN

1.11.1.1 Aprendizaje profundo y redes neuronales convolucionales

Según Soria Olivas et al. (2022), el aprendizaje profundo utiliza modelos compuestos por varias capas, los cuales permiten obtener representaciones complejas a partir de los datos. En el análisis de imágenes, las redes neuronales convolucionales aplican filtros para reconocer elementos visuales como bordes, colores, texturas y formas.

Las primeras capas de una red convolucional reconocen patrones visuales sencillos, mientras que las capas más profundas combinan esta información para representar características de mayor complejidad. Este proceso permite analizar las variaciones visibles presentes en las hojas de plátano y apoyar su clasificación de acuerdo con su estado general de salud vegetal (Tugrul et al., 2022).

En el alcance del presente proyecto, el modelo no identifica una enfermedad específica ni determina niveles de severidad. Su función consiste en reconocer características visuales asociadas con una hoja saludable o con una hoja cuya salud se encuentra comprometida (Sunitha et al., 2023).

1.11.1.2 Arquitectura ResNetVD34 y aprendizaje por transferencia

ResNet es una arquitectura de red neuronal convolucional que utiliza conexiones residuales o conexiones de acceso directo entre sus capas. Estas conexiones facilitan el flujo de la información y de los gradientes durante el entrenamiento, disminuyendo las dificultades que pueden presentarse al entrenar redes profundas (Raschka et al., 2023).

En el presente proyecto se utilizó una configuración ResNetVD34 basada en bloques residuales para obtener representaciones relacionadas con colores, bordes, texturas, formas y otras características visuales presentes en las imágenes. El uso de conexiones residuales facilita el aprendizaje en redes profundas y permite reutilizar representaciones visuales mediante aprendizaje por transferencia (Raschka et al., 2023).

En el presente proyecto, ResNetVD34 fue utilizada como extractor de características visuales. El modelo fue inicializado con pesos preentrenados en ImageNet, aplicando aprendizaje por transferencia. Chollet (2020) explica que el aprendizaje por transferencia permite reutilizar conocimientos adquiridos previamente a partir de una gran colección de imágenes. De esta manera, no es necesario entrenar completamente la red desde cero, sino adaptar sus características al nuevo problema mediante el entrenamiento de capas adicionales y el ajuste fino de determinadas capas del modelo.

En este caso, las características preentrenadas fueron adaptadas para apoyar la clasificación del estado general de salud vegetal de las hojas de plátano en las categorías saludable y salud comprometida, siguiendo el principio de reutilización de representaciones aprendido mediante transferencia (Chollet, 2020).

1.11.1.3 Adaptación de dominio DANN y capa de inversión de gradiente

La adaptación de dominio es una técnica de aprendizaje automático utilizada cuando existen diferencias entre los datos empleados para entrenar un modelo y aquellos utilizados durante su aplicación. Estas diferencias pueden ocasionar que el rendimiento disminuya al procesar imágenes capturadas en un entorno distinto al del conjunto de entrenamiento (Liu et al., 2022).

En el presente proyecto, las imágenes obtenidas del conjunto de datos público representaron el dominio fuente, mientras que las fotografías capturadas en condiciones reales de campo correspondieron al dominio objetivo. Entre ambos grupos pueden presentarse variaciones en la iluminación, los fondos, la posición de las hojas, las sombras, la distancia desde la cámara y la calidad de la fotografía, factores característicos de un cambio de dominio (Liu et al., 2022).

De acuerdo con HassanPour Zonoozi y Seydi (2023), los métodos adversarios de adaptación de dominio buscan obtener características que sean útiles para la tarea principal y, al mismo tiempo, difíciles de diferenciar según su dominio de procedencia. Para abordar esta situación se utilizó una Red Neuronal Adversaria para Adaptación de Dominio, conocida como DANN.

La arquitectura DANN está conformada por tres componentes principales: un extractor de características, un clasificador de clases y un discriminador de dominio (HassanPour Zonoozi & Seydi, 2023). En el modelo desarrollado, ResNetVD34 cumplió la función de extractor de características, el clasificador determinó si la hoja pertenecía a la categoría saludable o salud comprometida, mientras que el discriminador intentó reconocer si las características procedían del dominio fuente o del dominio objetivo.

Para conectar el extractor de características con el discriminador de dominio, el modelo utiliza una capa de inversión de gradiente. Durante la propagación hacia adelante, esta capa transmite las características sin modificarlas; durante la retropropagación, invierte el signo del gradiente y lo multiplica por un factor previamente establecido (HassanPour Zonoozi & Seydi, 2023).

Este procedimiento genera un entrenamiento adversario en el que el discriminador intenta reconocer el dominio de procedencia, mientras que el extractor busca producir características que dificulten esta identificación. De esta manera, DANN se utiliza con la finalidad de reducir la influencia de las diferencias visuales existentes entre los dominios fuente y objetivo (Liu et al., 2022).

1.11.2 Evaluación automatizada del estado general de salud vegetal

1.11.2.1 Características visuales de las hojas de plátano

Sunitha et al. (2023) señalan que determinadas alteraciones en las hojas de plátano pueden presentar diferentes características visuales relacionadas con su estado general. Una hoja de plátano saludable suele presentar una coloración verde relativamente uniforme y una estructura foliar sin alteraciones visuales significativas; en cambio, una hoja con salud comprometida puede presentar amarillamiento, manchas, pérdida de coloración, cambios de textura, áreas secas o daños visibles.

Para realizar una clasificación mediante imágenes, las redes neuronales convolucionales pueden aprender características relacionadas con el color, la textura, los bordes y las formas presentes en las hojas (Tugrul et al., 2022). Sin embargo, la apariencia de estas características también puede verse afectada por las condiciones de captura, como la iluminación, el fondo, la orientación y la calidad de la cámara; por ejemplo, el conjunto de imágenes descrito por Sunitha et al. (2023) fue capturado con dispositivos móviles en plantaciones y bajo diferentes condiciones de luz solar.

Por esta razón, el resultado generado por la aplicación representa una evaluación del estado visual general de la hoja y no la identificación de la causa específica de la alteración observada, una limitación coherente con la variabilidad visual presente en las imágenes de hojas de plátano (Sunitha et al., 2023).

1.11.2.2 Categorías de clasificación y umbral de confianza

El modelo distingue las imágenes en dos categorías: saludable y salud comprometida. En la primera se agrupan las hojas que no presentan alteraciones visuales relevantes; la segunda reúne imágenes en las que se observan cambios de color, textura o estructura (Sunitha et al., 2023).

Cuando se analiza una fotografía, el sistema estima una probabilidad para cada categoría, la opción con el valor más alto se presenta como resultado únicamente si alcanza el nivel mínimo de confianza establecido.

En la aplicación se definió un umbral de confianza del 65 %. Cuando la probabilidad máxima es igual o superior a este valor, se presenta la categoría correspondiente. Cuando

ninguna categoría alcanza el umbral, la aplicación muestra el resultado no determinado, siguiendo el principio de clasificación con opción de rechazo (Hendrickx et al., 2024). La clasificación no determinada no constituye una tercera clase entrenada, sino una medida de control incorporada después de obtener las probabilidades del modelo.

1.11.2.3 Métricas de evaluación del modelo

El rendimiento del modelo se evalúa mediante métricas de clasificación que permiten analizar sus resultados sobre imágenes reservadas para prueba. La exactitud representa la proporción total de predicciones correctas, mientras que la precisión permite conocer cuántas de las imágenes asignadas a una categoría fueron clasificadas correctamente (Raschka & Mirjalili, 2024).

El recall mide la capacidad del modelo para reconocer las imágenes que realmente pertenecen a cada categoría. El F1-score combina la precisión y el recall, proporcionando una medida equilibrada del rendimiento. También se utiliza la matriz de confusión para observar los aciertos y errores obtenidos en las categorías saludable y salud comprometida (Raschka & Mirjalili, 2024).

Las métricas obtenidas ayudan a reconocer el desempeño técnico del modelo. Sin embargo, los resultados deben analizarse con cautela debido a que el conjunto de prueba contiene una cantidad limitada de imágenes; por ello, en investigaciones posteriores será necesario incorporar más fotografías tomadas en el entorno local (Raschka & Mirjalili, 2024).

1.11.3 Metodología de Desarrollo

1.11.3.1 Metodología Ágil Scrum para la Gestión del Software

Scrum es un marco ágil que organiza el trabajo en periodos cortos conocidos como sprints. Al comenzar cada sprint se seleccionan las actividades que se llevarán a cabo y, durante su ejecución, se desarrollan las funcionalidades previstas. Al finalizar, el equipo revisa lo conseguido y realiza los ajustes necesarios para continuar con la construcción del sistema (Schwaber & Sutherland, 2020).

Este marco de trabajo facilita la organización de los requerimientos mediante el Product Backlog y el Sprint Backlog. También permite realizar entregas progresivas del producto, supervisar el avance del proyecto y corregir posibles dificultades durante el desarrollo (Schwaber & Sutherland, 2020).

1.12 Conclusiones del marco teórico

A partir de los fundamentos teóricos revisados, se concluyó que las redes neuronales convolucionales constituían una alternativa apropiada para el procesamiento y la clasificación de imágenes de hojas. Asimismo, se determinó que el empleo de la arquitectura ResNetVD34,

preentrenada con ImageNet, resultaba pertinente para aprovechar características visuales previamente aprendidas y ajustarlas al reconocimiento de variaciones relacionadas con el color, la textura y la estructura de las hojas de plátano.

También se concluyó que las diferencias entre las imágenes públicas utilizadas durante el entrenamiento y las fotografías capturadas en condiciones reales de campo representaban uno de los principales desafíos para la capacidad de generalización del modelo. Ante esta situación, la incorporación de DANN y de la capa de inversión de gradiente se consideró adecuada, debido a que estas técnicas permitían reducir la dependencia del dominio de origen y favorecían el aprendizaje de características compartidas entre ambos dominios.

La revisión teórica permitió establecer que el alcance del módulo debía limitarse a una evaluación general del estado visual de las hojas mediante las categorías hoja saludable y hoja con salud comprometida. Además, se determinó que, cuando ninguna categoría alcanzara el nivel mínimo de confianza, debía mostrarse un resultado no determinado. Por consiguiente, la respuesta generada por el modelo no debía interpretarse como la identificación de una enfermedad específica, la determinación de su grado de severidad o el reemplazo de la valoración realizada por un profesional del área agrícola.

Finalmente, se concluyó que TensorFlow Lite con optimización float16 constituía una alternativa apropiada para ejecutar modelos de aprendizaje profundo en dispositivos móviles, mientras que Flutter facilitaba la integración de funciones como el acceso a la cámara, la selección de imágenes, la presentación de probabilidades y el almacenamiento local de los resultados. De igual manera, Scrum se consideró una metodología adecuada para organizar el desarrollo mediante iteraciones, desde la definición de los requerimientos hasta la implementación y las pruebas del módulo móvil sin conexión a internet.

CAPÍTULO III

MARCO INVESTIGATIVO

1.13 Introducción

El presente capítulo describe el proceso metodológico seguido para el desarrollo del módulo móvil de inteligencia artificial orientado a la evaluación del estado general de salud vegetal de hojas de plátano. En este apartado se presentan los tipos de investigación utilizados, los métodos aplicados, la población considerada y las técnicas empleadas para obtener y analizar la información, elementos que forman parte de la organización metodológica de un estudio.

La metodología permitió organizar las actividades relacionadas con la preparación de las imágenes, el entrenamiento del modelo, la evaluación mediante métricas de clasificación y las pruebas funcionales de la aplicación. El alcance de la investigación se limitó a clasificar las hojas en las categorías saludable y salud comprometida, sin identificar enfermedades específicas ni calcular niveles de severidad; esta delimitación permitió mantener coherencia entre los objetivos, las técnicas empleadas y el análisis realizado (Vizcaíno Zúñiga et al., 2023).

1.14 Tipos de investigación

1.14.1 Investigación aplicada

La investigación aplicada se orienta al uso de conocimientos científicos y tecnológicos para atender necesidades concretas mediante soluciones prácticas. Bajo este enfoque, el presente proyecto empleó fundamentos de redes neuronales convolucionales, aprendizaje por transferencia y adaptación de dominio para desarrollar un módulo móvil capaz de evaluar el estado general de salud vegetal de hojas de plátano. Como resultado, se construyó y entrenó un modelo basado en ResNetVD34 y DANN, posteriormente convertido a TensorFlow Lite e integrado en una aplicación desarrollada con Flutter para realizar el procesamiento de imágenes directamente en el dispositivo móvil y sin depender de una conexión permanente a Internet (Hernández-Sampieri & Mendoza Torres, 2023).

Al finalizar el trabajo se obtuvo un modelo que combina ResNetVD34 y DANN; luego se generó una versión en TensorFlow Lite y se incorporó a una aplicación desarrollada con Flutter. De esta forma, el estudio se concretó en un módulo móvil capaz de procesar imágenes sin conexión a Internet, una estrategia compatible con la ejecución local de modelos de clasificación en dispositivos móviles (Muneeb et al., 2022).

1.14.2 Investigación explicativa

La investigación explicativa busca identificar las razones por las cuales se produce un fenómeno y analizar las relaciones existentes entre las variables involucradas. En este proyecto,

este tipo de investigación se utilizó para examinar cómo las diferencias entre las imágenes públicas y las fotografías capturadas en condiciones reales de campo podían influir en el desempeño del modelo de inteligencia artificial, considerando factores como la iluminación, los fondos naturales, las sombras, la orientación de las hojas, la distancia de captura y la calidad de las imágenes. De esta manera, el análisis permitió relacionar las condiciones visuales de ambos dominios con las variaciones observadas durante la evaluación del modelo (Arias González & Covinos Gallardo, 2021).

También se examinó la relación entre la preparación de los conjuntos de imágenes, la aplicación de DANN y los resultados alcanzados durante la evaluación. De esta manera, se facilitó la interpretación del rendimiento del modelo y la identificación de las limitaciones presentes en la clasificación de las hojas frente a diferencias entre dominio fuente y dominio objetivo (Liu et al., 2022).

1.14.3 Investigación de campo

La investigación de campo se caracteriza por obtener información directamente en el lugar donde ocurre el fenómeno estudiado, sin modificar las condiciones naturales del entorno. En este proyecto, este tipo de investigación se aplicó mediante la captura de fotografías de hojas de plátano en condiciones reales del entorno agrícola local, donde se presentaron variaciones de iluminación, sombras, fondos naturales, orientación de las hojas y distancia de captura. La recopilación directa de estas imágenes permitió trabajar con información propia del escenario de estudio y considerar las condiciones reales en las que posteriormente sería utilizado el módulo móvil de inteligencia artificial (Arias González & Covinos Gallardo, 2021).

Las fotografías recopiladas fueron utilizadas como dominio objetivo durante el entrenamiento y la evaluación de la adaptación de dominio. Su incorporación permitió analizar el comportamiento del modelo frente a imágenes diferentes a las procedentes del conjunto público utilizado como dominio fuente, considerando la variación entre distribuciones como un aspecto central de la adaptación de dominio (Liu et al., 2022).

Las fotografías tomadas en el entorno local se agruparon en las categorías saludable y salud comprometida de acuerdo con las etiquetas asignadas. Esta organización se utilizó únicamente durante el desarrollo del módulo y corresponde a una valoración visual; por ello, no debe interpretarse como un diagnóstico realizado por un profesional agrícola, ya que diferentes alteraciones foliares pueden presentar características visuales similares (Sunitha et al., 2023).

1.15 Métodos de investigación

1.15.1 Método Analítico

El método analítico permite estudiar un fenómeno mediante la separación y revisión de los elementos que lo componen, con el propósito de comprender su funcionamiento de manera organizada. En esta investigación se aplicó este método al examinar individualmente el conjunto de imágenes, la arquitectura ResNetVD34, el clasificador de clases, el discriminador de dominio y la capa de inversión de gradiente correspondiente a DANN. También se analizaron por separado las etapas de preparación de datos, entrenamiento, evaluación y conversión del modelo a TensorFlow Lite, considerando métricas como exactitud, precisión, recall, F1-score y matriz de confusión para interpretar el comportamiento de la solución desarrollada (Arias Gonzáles & Covinos Gallardo, 2021).

También se aplicó para revisar las etapas de preparación de los datos, entrenamiento, evaluación y conversión del modelo a TensorFlow Lite. De esta manera, fue posible identificar el comportamiento de cada componente y analizar los resultados obtenidos mediante las métricas de exactitud, precisión, recall, F1-score y matriz de confusión, utilizadas para evaluar modelos de clasificación (Raschka & Mirjalili, 2024).

1.15.2 Método deductivo

El método deductivo parte de principios y conocimientos generales para aplicarlos al análisis de situaciones particulares. En esta investigación, este método permitió trasladar fundamentos de redes neuronales convolucionales, aprendizaje por transferencia y adaptación de dominio hacia el desarrollo de un modelo orientado a evaluar el estado general de salud de hojas de plátano, relacionando la base teórica con las necesidades específicas del proyecto (Arias Gonzáles & Covinos Gallardo, 2021).

A partir de estos fundamentos se seleccionó ResNetVD34 como extractor de características y se incorporó DANN para reducir las diferencias entre las imágenes públicas y las fotografías capturadas en campo. Posteriormente, el modelo fue convertido a TensorFlow Lite e integrado en una aplicación móvil desarrollada con Flutter, obteniendo una solución capaz de clasificar las hojas como saludable o salud comprometida sin identificar enfermedades específicas ni niveles de severidad (Liu et al., 2022).

1.16 Fuentes de información de datos

1.16.1 Fuentes primarias

Las fuentes primarias estuvieron conformadas por la información obtenida directamente de los participantes relacionados con las actividades agrícolas de la Granja Experimental ULEAM, Extensión El Carmen. También se consideraron las fotografías de hojas

de plátano capturadas en condiciones reales de campo, debido a que fueron recopiladas directamente dentro del entorno de estudio, característica propia de la información obtenida de primera mano (Arias Gonzáles & Covinos Gallardo, 2021).

1.16.1.1 Encuesta

La encuesta constituye una técnica estructurada para recopilar información de un grupo de participantes mediante preguntas previamente definidas, permitiendo conocer opiniones, experiencias y características relacionadas con el objeto de estudio. En la presente investigación se aplicó a estudiantes de la carrera de Agropecuaria vinculados con las actividades de la Granja Experimental ULEAM, Extensión El Carmen, con la finalidad de identificar las prácticas empleadas para observar el estado de las plantas, las dificultades presentes durante la revisión visual de las hojas, el uso de dispositivos móviles, la disponibilidad de conexión a Internet y la percepción sobre una aplicación basada en inteligencia artificial. La información obtenida también contribuyó a establecer requerimientos relacionados con la cámara, la galería, la presentación de resultados comprensibles, el funcionamiento sin conexión y el almacenamiento local del historial de análisis (Feria Avila et al., 2020).

1.16.1.2 Entrevista

La entrevista permite obtener información detallada mediante una interacción planificada con una persona que posee conocimientos o experiencia sobre el fenómeno estudiado, mientras que su modalidad semiestructurada facilita profundizar en determinados temas sin limitarse exclusivamente a preguntas cerradas. En este proyecto se aplicó al Ing. Paul González, docente relacionado con la Granja Experimental ULEAM, Extensión El Carmen, con el propósito de conocer las actividades de monitoreo, las dificultades presentes durante la observación de las hojas, las condiciones del entorno agrícola, el acceso a asistencia técnica y las funcionalidades consideradas necesarias en una aplicación móvil. Las respuestas obtenidas permitieron complementar la identificación de necesidades y requerimientos del sistema, aunque no se utilizaron para validar profesionalmente las etiquetas de las fotografías ni para confirmar la presencia de enfermedades específicas (Feria Avila et al., 2020).

1.16.2 Fuentes secundarias

Las fuentes secundarias estuvieron conformadas por libros, artículos científicos, publicaciones académicas, documentación técnica oficial y trabajos investigativos relacionados con inteligencia artificial, redes neuronales convolucionales, aprendizaje por transferencia, adaptación de dominio, desarrollo móvil y metodología de investigación. Este

tipo de fuentes permite sustentar teórica y metodológicamente un estudio mediante información previamente publicada (Arias Gonzáles & Covinos Gallardo, 2021).

También se utilizó un conjunto público de imágenes de hojas de plátano como dominio fuente. Este material proporcionó fotografías organizadas que sirvieron como punto de partida para preparar y entrenar el modelo. Antes de utilizarlo, el conjunto fue revisado para eliminar inconsistencias y luego se reorganizó en las categorías saludable y salud comprometida, siguiendo el uso de conjuntos de imágenes organizados para tareas de clasificación de hojas de plátano (Arman et al., 2023).

Por su parte, las fuentes secundarias sirvieron para sustentar el marco teórico, establecer el procedimiento metodológico y seleccionar las tecnologías empleadas durante el desarrollo del módulo, manteniendo correspondencia entre el problema, los objetivos y las decisiones metodológicas de la investigación (Vizcaíno Zúñiga et al., 2023).

1.16.3 Población

La población de una investigación comprende el conjunto de individuos o unidades de análisis que comparten características relacionadas con el fenómeno estudiado y sobre los cuales se pretende obtener información. Para este trabajo se estableció una población de 25 estudiantes de los últimos semestres de la carrera de Agropecuaria de la ULEAM, Extensión El Carmen, vinculados con las actividades prácticas de la Granja Experimental. La elección de este grupo respondió a su relación directa con la observación y seguimiento de los cultivos de plátano, por lo que su participación permitió obtener información pertinente para identificar necesidades y requerimientos asociados al desarrollo del módulo móvil de inteligencia artificial (Arias Gonzáles & Covinos Gallardo, 2021).

1.16.4 Muestra censal

Debido a que la población definida para la investigación estuvo conformada únicamente por 25 estudiantes y todos se encontraban disponibles para participar, se optó por trabajar con la totalidad del grupo mediante una muestra censal. Este procedimiento permitió incluir a los 25 estudiantes de los últimos semestres de Agropecuaria vinculados con las actividades prácticas de la Granja Experimental ULEAM, Extensión El Carmen, por lo que población y muestra coincidieron. En consecuencia, no fue necesario aplicar una fórmula para calcular un tamaño muestral ni establecer un margen de error para seleccionar participantes, debido a que la información fue recopilada de todas las unidades de análisis consideradas en el estudio (Vizcaíno Zúñiga et al., 2023).

1.16.5 Análisis de las herramientas de recolección de datos a utilizar

1.16.5.1 Encuesta

La encuesta utilizada como instrumento de recolección estuvo conformada por 18 preguntas cerradas organizadas mediante opciones de respuesta única y escalas de valoración. Su aplicación a los estudiantes de Agropecuaria de la Granja Experimental ULEAM, Extensión El Carmen, permitió obtener información sobre las prácticas utilizadas para observar los cultivos, las dificultades presentes durante la revisión visual de las hojas, el uso de dispositivos móviles, la disponibilidad de conexión a Internet y la aceptación de una aplicación móvil basada en inteligencia artificial. Asimismo, las respuestas permitieron establecer necesidades funcionales relacionadas con el procesamiento de fotografías, el funcionamiento sin conexión, la presentación comprensible de resultados y el almacenamiento local de los análisis realizados (Feria Avila et al., 2020).

1.16.5.2 Entrevista

La entrevista semiestructurada aplicada al Ing. Paul González se organizó mediante preguntas abiertas relacionadas con el monitoreo de los cultivos, la observación visual de las hojas, las condiciones ambientales de la Granja Experimental y las características que debería incorporar una herramienta móvil de apoyo. Este tipo de instrumento permitió obtener información complementaria sobre experiencias, necesidades y condiciones del entorno que difícilmente podrían abordarse con el mismo nivel de detalle mediante preguntas cerradas. Los resultados fueron utilizados para fortalecer la definición de requerimientos del módulo móvil y comprender las condiciones reales en las que podría utilizarse la aplicación, sin emplear la entrevista como mecanismo de validación de enfermedades o de las etiquetas asignadas a las imágenes (Feria Avila et al., 2020).

1.16.5.3 Estructura de los instrumentos de recolección de datos aplicados

1.16.5.3.1 Estructura de encuesta

La encuesta estuvo compuesta por 18 preguntas cerradas, organizadas mediante opciones de respuesta única y escalas de valoración. Las preguntas permitieron recopilar información sobre el uso de herramientas digitales, las prácticas de observación de los cultivos, la disponibilidad de conexión a Internet y la aceptación de una aplicación móvil basada en inteligencia artificial, manteniendo una estructura uniforme para facilitar la comparación de respuestas (Feria Avila et al., 2020).

1.16.5.3.2 Estructura de la Entrevista

La entrevista semiestructurada estuvo organizada mediante tres preguntas abiertas relacionadas con las actividades de monitoreo, la observación visual de las hojas, las condiciones del entorno agrícola y el uso de herramientas móviles. Este instrumento permitió conocer las necesidades y recomendaciones del Ing. Paul González para el desarrollo del módulo, aprovechando el carácter flexible de las preguntas abiertas en una entrevista semiestructurada (Feria Avila et al., 2020).

1.16.6 Plan de Recolección de datos

Tabla 1 Plan de Recolección de datos

ACTIVIDAD	RESPONSABLES	FECHA
Entrevista al docente Ing. Paul González , de la Granja experimental ULEAM, Extensión El Carmen	Zambrano Jean fredd Pita Adrián Pilataxi Diego	14/04/2026 – 15/04/2026
Encuesta a los estudiantes de 5to y 6to de Agropecuaria de la Granja Experimental Uleam extensión El Carmen	Zambrano Jean fredd Pita Adrián Pilataxi Diego	14/04/2026 – 15/04/2026

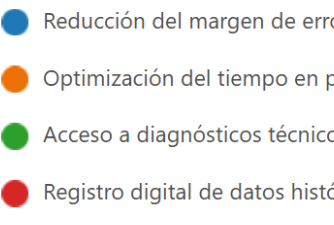
1.17 Análisis y presentación de resultados

1.17.1 Presentación y descripción de los resultados obtenidos

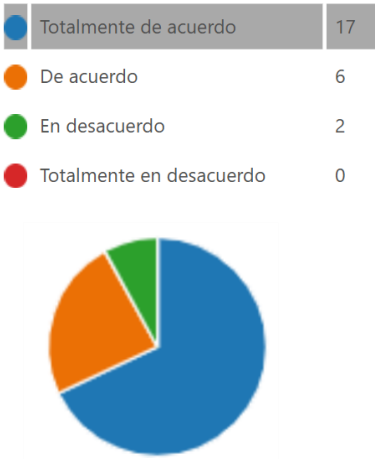
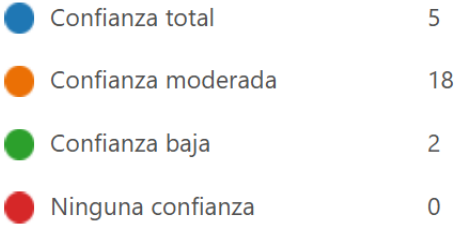
1.17.1.1 Análisis de encuesta realizada a estudiantes de la carrera Agropecuaria de la Uleam extensión El Carmen

La encuesta aplicada estuvo conformada por 18 preguntas: seis de carácter general sobre el tema del proyecto y doce específicas, distribuidas en cuatro preguntas para cada uno de los tres módulos correspondientes al grupo de trabajo. Para aplicar el instrumento, se conformaron dos grupos de tres estudiantes. En la presente investigación se analizaron y tabularon las seis preguntas generales y las cuatro relacionadas directamente con el módulo móvil basado en inteligencia artificial. Las ocho preguntas restantes correspondieron a los otros dos módulos y no se incluyeron en el análisis, debido a que se encontraban fuera del alcance de este trabajo.

Tabla 2 Análisis y presentación de resultados

PREGUNTA	GRAFICO	INTERPRETACION
Pregunta 1: ¿Cómo califica el nivel de integración de herramientas digitales en las prácticas agrícolas actuales de la Extensión El Carmen?	Muy avanzado Adecuado Insuficiente Inexistente 5 14 6 0 	Los resultados reflejan una valoración positiva del uso actual de herramientas digitales en las prácticas agrícolas. Esta percepción permite reconocer que existe apertura hacia la incorporación de soluciones tecnológicas más especializadas que apoyen las actividades desarrolladas en la Extensión El Carmen.
Pregunta 2: ¿Qué importancia tiene para su formación profesional el aprendizaje de soluciones basadas en Agricultura de Precisión?	Crucial para el mercado laboral Importante para la especializa... Complementaria, pero no ese... Irrelevante para mi perfil 8 16 1 0 	La información evidencia una opinión favorable sobre la importancia de la agricultura de precisión en la formación profesional. Los estudiantes reconocen que estas soluciones fortalecen su preparación técnica y responden a las exigencias actuales del sector agropecuario.
Pregunta 3: ¿Cuál considera que es la principal ventaja de utilizar aplicaciones con Inteligencia Artificial para el	Reducción del margen de erro... Optimización del tiempo en p... Acceso a diagnósticos técnico... Registro digital de datos histó... 7 12 6 0 	Las respuestas muestran que los participantes valoran tanto la reducción del tiempo de evaluación en campo como la obtención de resultados técnicos detallados. Esta información respalda el desarrollo del módulo, debido

PREGUNTA	GRAFICO	INTERPRETACION
monitoreo de cultivos?		a que la inteligencia artificial puede agilizar la revisión visual de las hojas y presentar información comprensible para el usuario.
Pregunta 4: Ante el desarrollo de una herramienta tecnológica, ¿cuál considera que es el mayor reto para su adopción entre los estudiantes?	● Limitaciones de hardware (telé... 10 ● Falta de capacitación técnica e... 7 ● Dificultad de acceso a internet... 7 ● Resistencia al reemplazo de m... 1 	Los resultados permiten identificar que la conectividad limitada en el campo constituye uno de los principales retos para la adopción de herramientas tecnológicas. Por ello, el módulo debe funcionar sin conexión a Internet, mantener una interfaz sencilla y responder adecuadamente a la realidad de la Granja Experimental.
Pregunta 5: ¿Qué funcionalidad valoraría más en una aplicación móvil diseñada para el entorno de la Granja Experimental?	● Interfaz intuitiva y de rápida n... 8 ● Capacidad de funcionamiento... 11 ● Alta precisión en la identificaci... 6 ● Generación de reportes auto... 0	La funcionalidad más valorada se relaciona con el funcionamiento sin conexión a Internet, seguida de la necesidad de contar con una interfaz intuitiva y una evaluación clara de las hojas. Esta tendencia confirma que

PREGUNTA	GRAFICO	INTERPRETACION
		la aplicación debe ser práctica, rápida y adecuada para entornos con limitaciones de conectividad.
Pregunta 6: ¿Cree usted que la universidad debería priorizar el desarrollo de software propio para la gestión de sus procesos agropecuarios?		Las respuestas reflejan una aceptación mayoritaria hacia el desarrollo de software propio para apoyar los procesos agropecuarios de la universidad. Esto evidencia la pertinencia de crear herramientas ajustadas a las necesidades reales de la Extensión El Carmen.
Pregunta 15: ¿En qué medida considera que las enfermedades fitosanitarias limitan el aprendizaje práctico en los cultivos de la granja?		Los resultados sugieren que las alteraciones fitosanitarias influyen de manera importante en el aprendizaje práctico dentro de los cultivos de la granja. Esta situación respalda la conveniencia de contar con herramientas tecnológicas que faciliten la observación y una evaluación preliminar de las hojas.
Pregunta 16: ¿Cuál es su nivel de confianza al emitir un diagnóstico fitosanitario sin ayuda tecnológica?		La presencia de distintos niveles de confianza al realizar una evaluación fitosanitaria sin apoyo tecnológico demuestra la necesidad de una herramienta

PREGUNTA	GRAFICO	INTERPRETACION
		que acompañe la revisión visual de las hojas. El módulo móvil puede contribuir a que el usuario disponga de una referencia inicial más clara durante el trabajo de campo.
Pregunta 17: ¿Considera que el uso de software de diagnóstico reduce el impacto ambiental al evitar aplicaciones químicas preventivas?	● Totalmente de acuerdo ● De acuerdo ● En desacuerdo ● Totalmente en desacuerdo 5 19 1 0 	Los encuestados considera que las herramientas tecnológicas podrían ayudar a disminuir el impacto ambiental relacionado con la aplicación preventiva de productos químicos, esta respuesta refleja una valoración positiva del software como recurso de apoyo para analizar información y tomar decisiones más cuidadosas en el manejo de los cultivos.
Pregunta 18: ¿Qué tan probable es que usted utilice una aplicación móvil para apoyar su trabajo de tesis o proyectos de aula?	● Muy probable ● Probable ● Poco probable ● Nada probable 9 13 3 0 	Los resultados muestran que de los estudiantes considera probable o muy probable utilizar la aplicación móvil en trabajos de tesis o proyectos académicos, esto refleja una alta disposición para emplear la herramienta como apoyo en actividades de aprendizaje e investigación.

**1.17.1.2 Análisis de entrevista realizada al Docente de la Granja Experimental Uleam
extensión El Carmen**

Tabla 3 Análisis de entrevista

PREGUNTA	RESPUESTA	INTERPRETACION
Pregunta1: ¿Qué características visuales se observan en las hojas de plátano para determinar si presentan un estado saludable o una condición de salud comprometida?	Bueno, cuando una hoja de plátano está saludable se observa con un color verde uniforme, una buena firmeza y una superficie limpia, sin manchas ni deformaciones, en cambio cuando la planta empieza a presentar problemas, la hoja cambia de color; presentan manchas amarillas, cafés o negras, bordes secos, zonas quemadas o partes marchitas, sin embargo también se puede notar que la hoja pierde vigor, se dobla o se rompe con mayor facilidad, en algunos de los casos las manchas comienzan pequeñas y luego se van extendiendo, lo que indica que la afectación está avanzando.	La respuesta del entrevistado señala que el estado de las hojas puede revisarse a partir de rasgos visibles, entre ellos el color, las manchas, el marchitamiento, la sequedad y las deformaciones, su experiencia en el ámbito laboral también destaca la importancia de la observación visual para reconocer posibles alteraciones de manera temprana, esta información aporta muchos criterios útiles para desarrollar una aplicación móvil que analice fotografías de hojas mediante inteligencia artificial. Esta interpretación coincide con la descripción de alteraciones visuales que pueden observarse en hojas de plátano.
Pregunta 2: ¿Qué dificultades se presentan durante la evaluación visual de las hojas de plátano en las	En campo no siempre es fácil evaluar las hojas, porque las condiciones cambian bastante, a veces hay mucha luz solar y eso altera el color	La respuesta evidencia que la evaluación visual en campo presenta varias dificultades, principalmente por factores ambientales como la

PREGUNTA	RESPUESTA	INTERPRETACION
condiciones reales de la Granja Experimental?	real de la hoja en la fotografía, también puede haber sombra, lluvia, polvo, humedad o daños causados por el viento, otro problema es que algunas enfermedades o deficiencias se parecen visualmente, entonces una mancha puede confundirse con otra afectación, además, no todas las hojas están en la misma posición ni tienen el mismo nivel de desarrollo, por eso se necesita experiencia para interpretar bien lo que se observa.	iluminación, la humedad, la lluvia, la sombra y la posición de las hojas, también se identifica que algunas afectaciones pueden tener síntomas similares, lo que dificulta un diagnóstico preciso, esto demuestra la necesidad de una herramienta tecnológica que ayude a estandarizar la evaluación y reduzca los errores durante el análisis visual de las hojas en condiciones reales de la Granja Experimental. Estas variaciones del entorno también son relevantes cuando las imágenes proceden de condiciones distintas a las utilizadas durante el entrenamiento del modelo.
Pregunta 3: ¿Qué funcionalidades debería incluir una aplicación móvil basada en inteligencia artificial para apoyar la evaluación del estado general de las hojas de plátano?	Una aplicación de ese tipo debería permitir tomar fotografías de las hojas directamente en campo y analizarlas de forma rápida, sería importante que indique si la hoja está saludable o si presenta algún nivel de afectación, ya que también	De acuerdo con el entrevistado, la aplicación debería ser sencilla, rápida y adecuada a las tareas realizadas en el campo, entre las funciones consideradas útiles se encuentran la captura de fotografías, el análisis automático, la

PREGUNTA	RESPUESTA	INTERPRETACION
	debería mostrar el porcentaje o nivel de daño, guardar un historial de evaluaciones y registrar la fecha y el lugar donde se tomó la imagen o se recolecto los datos, otra función útil sería dar recomendaciones básicas, como realizar seguimiento a la planta, revisar otras hojas o consultar con un técnico si el daño es avanzado, también la aplicación debe ser sencilla de usar, porque en campo se necesita algo práctico y rápido.	clasificación general del estado de la hoja y el almacenamiento de los resultados con su ubicación e historial, también se valoró la presentación de recomendaciones generales que sirvan como orientación para el usuario, estas funciones pueden apoyar el seguimiento de las hojas y la toma de decisiones en la Granja Experimental, sin sustituir la valoración de un profesional agrícola. La ejecución rápida y local del análisis es coherente con el uso de modelos de clasificación optimizados para dispositivos móviles.

1.17.2 Informe final del análisis de datos

El análisis de la encuesta aplicada a 25 estudiantes de la carrera de Agropecuaria y de la entrevista realizada al Ing. Paul González permitió identificar las principales necesidades relacionadas con el uso de herramientas tecnológicas en la Granja Experimental ULEAM, Extensión El Carmen. Los resultados evidenciaron una percepción favorable hacia la incorporación de aplicaciones móviles e inteligencia artificial como apoyo para las actividades agrícolas. La integración de ambas técnicas permitió complementar la información obtenida durante el estudio.

Al revisar las funciones preferidas por los participantes, el 44 % eligió el funcionamiento sin conexión a Internet, el 32 % dio mayor importancia a una interfaz rápida e intuitiva, mientras que el 24 % señaló la precisión en la evaluación de las hojas. Además, el 92 % apoyó que la universidad priorizara el desarrollo de software propio ajustado a sus

actividades agropecuarias. Estos datos corresponden a los resultados obtenidos mediante el instrumento de encuesta aplicado a la población censal.

Además, los resultados demostraron que el 88 % consideró que las alteraciones fitosanitarias limitan de manera considerable o en gran medida el aprendizaje práctico; asimismo, el 72 % manifestó una confianza moderada al efectuar una evaluación sin apoyo tecnológico y el 88 % expresó que probablemente utilizaría una aplicación móvil en trabajos académicos o proyectos de investigación. La interpretación se realizó a partir de las respuestas registradas en la encuesta.

La entrevista permitió complementar estos resultados al identificar las dificultades presentes durante la observación visual de las hojas en condiciones reales de campo. También se destacó la necesidad de que la aplicación presente resultados claros, sea fácil de utilizar y proporcione una respuesta rápida desde el dispositivo móvil, aspectos identificados mediante el análisis de la entrevista semiestructurada.

A partir de esta información se definieron los principales requerimientos del módulo: funcionamiento sin conexión, uso de la cámara y la galería, una interfaz fácil de comprender, visualización clara de los resultados y almacenamiento local del historial. Estos elementos orientaron el desarrollo del módulo móvil para clasificar visualmente las hojas de plátano como saludable y salud comprometida, como una herramienta de apoyo que no sustituye la valoración de un profesional agrícola; la definición de requerimientos se fundamentó en la información recopilada mediante los instrumentos aplicados.

CAPÍTULO IV

MARCO PROPOSITIVO

1.18 Introducción

El presente capítulo describe la propuesta tecnológica desarrollada para apoyar la evaluación automatizada del estado general de salud vegetal en hojas de plátano mediante un módulo móvil basado en inteligencia artificial. Esta propuesta surge a partir de la necesidad de contar con una herramienta funcional que permita analizar imágenes capturadas en condiciones reales de campo dentro de la Granja Experimental ULEAM, Extensión El Carmen, sin depender de una conexión permanente a Internet.

En este apartado se detallan los recursos utilizados, los requerimientos del sistema, el diseño de la solución, la arquitectura implementada y el proceso de desarrollo seguido mediante la metodología Scrum. Asimismo, se explica la preparación de los conjuntos de imágenes, el entrenamiento del modelo ResNetVD34 con adaptación de dominio DANN, su conversión a TensorFlow Lite y su integración en una aplicación móvil desarrollada con Flutter.

La propuesta se orienta a clasificar las hojas de plátano en las categorías saludable y salud comprometida, incorporando además un resultado no determinado cuando el nivel de confianza del modelo no alcanza el umbral establecido. De esta manera, el módulo funciona como una herramienta de apoyo para la observación visual general de las hojas, sin reemplazar la valoración de un profesional agrícola ni afirmar el diagnóstico de enfermedades fitosanitarias específicas.

1.19 Descripción de la propuesta

La propuesta consiste en el desarrollo de un módulo móvil de inteligencia artificial orientado a evaluar automáticamente el estado general de salud vegetal de hojas de plátano. Este módulo forma parte del Proyecto Integrador de la carrera de Ingeniería de Software de la Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen, y fue diseñado como un componente que puede integrarse posteriormente con la aplicación principal del proyecto.

La solución responde a la necesidad de contar con una herramienta tecnológica que apoye la observación visual de las hojas en condiciones reales de campo. La valoración tradicional puede variar según la experiencia de la persona, la iluminación, el estado de la hoja y las condiciones existentes durante la inspección.

Para apoyar este proceso se desarrolló un modelo basado en ResNetVD34 preentrenada con ImageNet y adaptación de dominio DANN. ResNetVD34 se utiliza para extraer características visuales relacionadas con el color, la textura, los bordes y la estructura de las

hojas, por otra parte, DANN busca reducir las diferencias existentes entre las imágenes públicas utilizadas como dominio fuente y las fotografías locales empleadas como dominio objetivo.

Para trabajar con el modelo, las fotografías se separaron en las categorías saludable y salud comprometida, antes del entrenamiento se tomó en cuenta depuradas, redimensionadas y divididas en datos de entrenamiento, validación y prueba, el modelo fue entrenado y evaluado mediante exactitud, precisión, recall, F1-score y matriz de confusión.

Después de completar el entrenamiento, el modelo se convirtió a TensorFlow Lite con optimización float16 y se incorporó a la aplicación móvil creada con Flutter, de esta manera las imágenes pueden procesarse en el propio dispositivo, sin enviarlas a un servidor ni depender de una conexión a Internet.

El usuario puede tomar una fotografía con la cámara o escoger una imagen desde la galería, una vez realizado el análisis, la aplicación muestra las probabilidades correspondientes a las categorías saludable y salud comprometida, si el valor alto no alcanza el 65 %, el resultado se presenta como no determinada, evitando presentar una clasificación cuando la confianza del modelo es insuficiente.

Además, la aplicación muestra recomendaciones generales y conserva localmente un historial de los análisis realizados. Esta información permite consultar resultados anteriores sin utilizar una base de datos externa.

El módulo se desarrolló como un componente independiente que puede incorporarse al sistema informático principal, esta organización facilita realizar mejoras posteriores, como ampliar el conjunto de imágenes, ajustar el rendimiento del modelo o también añadir nuevas categorías, sin tener que reconstruir toda la aplicación.

El alcance de la propuesta se centra únicamente en una revisión general de la apariencia de las hojas de plátano, el resultado no identifica enfermedades concretas, por lo tanto no calcula niveles de severidad, no prescribe tratamientos fitosanitarios y no sustituye la valoración de un profesional agrícola.

1.20 Determinación de recursos

El desarrollo del módulo móvil de inteligencia artificial requirió la utilización de recursos humanos, tecnológicos y económicos. Estos recursos permitieron ejecutar las actividades de planificación, preparación de las imágenes, entrenamiento del modelo, desarrollo de la aplicación móvil, realización de pruebas y documentación del proyecto.

Los recursos humanos estuvieron representados por el autor del módulo, el tutor del proyecto y las personas que colaboraron durante la recolección de información y fotografías,

su participación permitió organizar las actividades, revisar el avance y orientar el desarrollo de la solución tecnológica.

Para desarrollar el proyecto se utilizó un equipo informático, un teléfono móvil y las diferentes herramientas de software utilizadas durante el proyecto, las cuales son: Windows 11, Visual Studio Code, Python, TensorFlow, Keras, Flutter, Dart, TensorFlow Lite, Git y GitHub, las pruebas funcionales de la aplicación se realizaron en un teléfono móvil.

El presupuesto considero el uso de equipos, servicios digitales, conexión a Internet, movilización y horas de desarrollo, necesarios para completar el módulo, al registrar cada uno de los gastos fue posible calcular el costo aproximado para completar el módulo.

La disponibilidad y organización de estos recursos permitió entrenar el modelo basado en ResNetVD34 y adaptación de dominio DANN, convertirlo a TensorFlow Lite y su integración en una aplicación móvil sin conexión a internet.

1.20.1 Recursos Humanos

Tabla 4 Recursos Humanos

Recursos humanos	Cargo	Función
Jean Zambrano	Autor y Desarrollador	Desarrollo de la investigación, recopilación de información, preparación de los conjuntos de imágenes, entrenamiento y evaluación del modelo basado en ResNetVD34 y DANN, integración en Flutter, realización de pruebas y documentación del proyecto.
Ing. Carlos López	Tutor del proyecto	Supervisar el cumplimiento de los objetivos, orientar las decisiones metodológicas y validar el desarrollo académico de la investigación.

1.20.2 Recursos Tecnológicos

Tabla 5 Recursos Tecnológicos

Recurso	Característica	Descripción
Laptop	Sistema Operativo: Microsoft Windows 11 Home Single Language (64 bits). Procesador: Intel® Core™ i7-1255U de 12. ^a generación (10 núcleos, 12 hilos, 1.70 GHz). Memoria RAM: 16 GB. Arquitectura: x64. Modo BIOS: UEFI.	Equipo utilizado para preparar los conjuntos de imágenes, programar, entrenar y evaluar el modelo, desarrollar la aplicación móvil y elaborar la documentación.
Teléfono Infinix X6858	Dispositivo móvil con sistema operativo Android.	Utilizado para instalar el APK y realizar pruebas de la app, como cámara, escáner. Galería, ect.
Python	Lenguaje de programación utilizado para el desarrollo del modelo de inteligencia artificial.	Utilizado para la preparación de datos, el entrenamiento, la evaluación y la conversión del modelo
TensorFlow y Keras	Bibliotecas para el desarrollo y entrenamiento de modelos de aprendizaje profundo.	Utilizadas para implementar ResNetVD34, el clasificador y los componentes de adaptación de dominio DANN.
ResNetVD34	Red convolucional configurada con pesos aprendidos a partir de ImageNet.	Utilizada como extractor de características visuales de las imágenes de hojas de plátano.

Recurso	Característica	Descripción
GitHub	Plataforma para control de versiones basada en Git.	Utilizada para el almacenamiento, control de versiones y seguimiento del código fuente desarrollado durante la investigación.
DANN	Arquitectura para adaptación de dominio.	Empleada para reducir las diferencias entre las imágenes públicas y las fotografías capturadas en condiciones reales de campo.
Microsoft Word	Procesador de texto.	Utilizado para la elaboración y edición de la documentación correspondiente al trabajo de titulación.
Visual Studio Code	Entorno de desarrollo de software.	Utilizado para programar el modelo de inteligencia artificial y la aplicación desarrollada con Flutter.
TensorFlow Lite	Formato optimizado para modelos móviles.	Utilizado para convertir el modelo final con optimización float16 y se pueda usar sin conexión a Internet.
Flutter y Dart	Framework y lenguaje para desarrollo móvil.	Se utilizaron para desarrollar la interfaz y las funcionalidades de la cámara, galería, resultados e historial.
tflite_flutter	Paquete de Flutter para ejecutar los modelos TensorFlow Lite.	Se utilizó para realizar la inferencia directamente dentro del dispositivo móvil.
SharedPreferences	Herramienta de almacenamiento local.	Se utilizó para guardar y consultar el historial de los análisis realizados.

1.20.3 Recursos Económicos

Tabla 6 Recursos Económicos

Cantidad	Recurso	Precio unitario	Subtotal
1	Laptop	\$290,00	\$290,00
1	Teléfono celular (uso para pruebas y documentación)	\$175,00	\$175,00
6 meses	Servicio de Internet	\$30,00	\$180,00
200 horas	Horas de programación	\$5	\$ 1.000,00
6 meses	Energía eléctrica	\$15,00	\$90,00
1	Impresiones y encuadernación	\$70,00	\$70,00
1	Gastos de movilización	\$80,00	\$80,00
1 mes	agoSuscripción a Abacus.AI	\$10,00	\$10,00
1 mes	Suscripción a ChatGPT Plus	\$20,00	\$20,00
1	Flutter, Visual Studio Code	\$00,00	\$00,00
1	TensorFlow y Keras	\$00,00	\$00,00
1	Python	\$00,00	\$00,00
1	GitHub	\$00,00	\$00,00
TOTAL:			\$1.915,00

1.21 Desarrollo del sistema bajo la metodología SCRUM

Scrum fue seleccionado como marco de trabajo para organizar el desarrollo del módulo móvil de inteligencia artificial. Su enfoque iterativo permitió distribuir las actividades en sprints, revisar los avances obtenidos y realizar ajustes durante cada etapa del proyecto.

Scrum se utilizó para organizar las actividades del proyecto, desde definición de requerimientos hasta la integración de Flutter, dentro de este proceso se incluyeron la preparación de las imágenes, el entrenamiento del modelo con ResNetVD34 y DANN, la evaluación de los resultados y la conversión a TensorFlow Lite

El trabajo se distribuyó en cuatro sprints, cada uno reunió tareas y entregables concretos, comenzando con la planificación y finalización con la implementación y las pruebas funcionales en un dispositivo Android, esta división permitió revisar los avances al cierre de cada etapa y corregir los problemas encontrados antes de iniciar la siguiente.

Las historias de usuario, el Product Backlog y el Sprint Backlog ayudaron a registrar los requerimientos y dar seguimiento a las actividades, mediante esta organización se integró

el modelo de inteligencia artificial con la cámara, la galería, la visualización de resultados y el historial almacenado localmente.

Como resultado, se obtuvo un módulo móvil capaz de ejecutar la evaluación del estado general de salud de las hojas de plátano sin conexión a Internet y con posibilidad de integración dentro del sistema informático principal.

1.21.1 Descripción del producto

1.21.1.1 Propósito del producto

Desarrollar un módulo móvil de inteligencia artificial basado en ResNetVD34 y adaptación de dominio DANN, capaz de evaluar el estado general de salud vegetal de hojas de plátano mediante el análisis de imágenes.

El producto tiene como propósito proporcionar una herramienta tecnológica de apoyo que pueda utilizarse en condiciones reales de campo y en lugares con conectividad limitada. Asimismo, fue diseñado como un componente independiente que puede integrarse posteriormente dentro del sistema informático principal del Proyecto Integrador.

El módulo no identifica enfermedades específicas, no calcula niveles de severidad y no sustituye la valoración realizada por un profesional agrícola.

1.21.1.2 Funcionalidades claves

El módulo incorpora las siguientes funcionalidades:

- Captura de fotografías mediante la cámara del dispositivo móvil.
- Selección de imágenes almacenadas en la galería.
- Preprocesamiento automático de la imagen a un tamaño de $224 \times 224 \times 3$.
- Ejecución local del modelo TensorFlow Lite con funcionamiento offline.
- Clasificación de la hoja como saludable o salud comprometida.
- Presentación del resultado no determinada cuando la confianza es inferior al 65 %.
- Visualización de las probabilidades obtenidas para cada categoría.
- Presentación de recomendaciones generales según el resultado.
- Almacenamiento local del historial de análisis.
- Consulta de resultados registrados anteriormente.
- Integración del módulo con el sistema informático principal

1.21.1.3 Usuarios objetivos

Tabla 7 Usuarios objetivos

Tipo de usuario	Principales funcionalidades
Investigador	Gestiona el conjunto de datos, ejecuta el entrenamiento del modelo DANN, valida los resultados obtenidos y realiza mejoras durante el proceso de investigación.
Sistema Integrador	Envía imágenes foliares al módulo de Inteligencia Artificial y recibe el diagnóstico generado para su integración con los demás módulos del proyecto.
Técnico agrícola	Utiliza los diagnósticos generados por el módulo como apoyo para la interpretación del estado fitosanitario del cultivo y la toma de decisiones relacionadas con el monitoreo agrícola.

1.21.1.4 Condiciones de Éxito del Producto

Para considerar satisfactorio el desarrollo del módulo móvil de inteligencia artificial, se establecieron las siguientes condiciones de éxito:

El modelo debe procesar imágenes de hojas de plátano capturadas mediante la cámara o seleccionadas desde la galería.

El sistema debe clasificar cada fotografía en una de las dos opciones establecidas: saludable o saludable comprometida.

Cuando la probabilidad más alta sea inferior al 65 %, la ampliación debe presentar la respuesta no determinada.

El análisis debe efectuarse directamente en el teléfono mediante TensorFlow Lite, sin requerir acceso a internet ni el envío de información a servidores externos.

Después de procesar la imagen, la aplicación debe mostrar las probabilidades calculadas para ambas categorías y acompañar el resultado con recomendaciones generales.

Cada análisis debe conservarse en el almacenamiento interno del dispositivo para que el usuario pueda revisarlo posteriormente desde el historial.

El desempeño del modelo debe medirse mediante exactitud, precisión, recall, F1-score y matriz de confusión.

La aplicación debe instalarse y ejecutarse en un dispositivo Android sin errores que impidan utilizar sus funciones principales.

El módulo debe conservar su funcionamiento independiente para facilitar su incorporación posterior a la aplicación principal del proyecto integrado.

Los resultados deben presentarse como una evaluación del estado visual general de la hoja y no como la identificación de una enfermedad específica.

1.21.2 Requerimientos del sistema

1.21.2.1 Historias de Usuarios

Las historias de usuario constituyen una técnica utilizada en Scrum para describir, de manera breve y comprensible, las necesidades que debe cubrir un sistema. Estas permiten organizar las funcionalidades desde la perspectiva de las personas que utilizarán el producto y sirven como base para establecer prioridades dentro del Product Backlog.

Para este proyecto, las historias de usuario se elaboraron a partir de las funciones principales del módulo móvil, entre ellas se incluyeron la captura de fotografías y selección de imágenes, el análisis del estado visual de las hojas, la presentación de los resultados y la consulta del historial almacenado localmente.

En cada historia se registraron el tipo de usuario, la necesidad planteada, su propósito, la prioridad, el riesgo y los criterios de aceptación, estos datos permitieron organizar las actividades desarrolladas durante los sprints y con eso verificar el cumplimiento de los requerimientos definidos.

1.21.2.1.1 Historia de Usuario 1: Carga y organización del conjunto de imágenes

Tabla 8 *Historia de Usuario 1: Carga y organización del conjunto de imágenes*

HISTORIA DE USUARIO			
Id de historia	HU01	Título:	Carga y organización del conjunto de imágenes
Prioridad:	alto	Usuario:	Desarrollador del módulo
Riesgo de desarrollo:	medio	Tamaño de la tarea:	grande
Como:	Desarrollador responsable del modelo de inteligencia artificial.		
Quiero:	Cargar y organizar las imágenes del dominio fuente y del dominio objetivo.		
Para:	Preparar los conjuntos que serán utilizados durante el entrenamiento, validación y prueba.		
Criterios de aceptación:	Las imágenes deberán cargarse correctamente, organizarse en las categorías saludable y salud comprometida, identificarse según su dominio y verificarse para evitar archivos incompatibles.		

HISTORIA DE USUARIO	
Dependencias:	Conjunto público disponible, fotografías locales recopiladas y estructura de carpetas definida.

1.21.2.1.2 Historia de Usuario 2: Procesamiento del conjunto de imágenes

Tabla 9 *Historia de Usuario 2: Procesamiento del conjunto de imágenes*

HISTORIA DE USUARIO			
Id de historia	HU02	Título:	Preprocesamiento del conjunto de imágenes
Prioridad:	alto	Usuario:	Desarrollador del módulo
Riesgo de desarrollo:	alto	Tamaño de la tarea:	grande
Como:	Desarrollador responsable del modelo de inteligencia artificial		
Quiero:	Depurar, redimensionar, normalizar y organizar las imágenes		
Para:	Obtener conjuntos de datos adecuados para el entrenamiento y la evaluación del modelo		
Criterios de aceptación:	Se deberán eliminar imágenes duplicadas o incompatibles, redimensionar las fotografías a 224 × 224 píxeles, balancear las categorías y dividir los datos en entrenamiento, validación y prueba.		
Dependencias:	Conjuntos de imágenes cargados y librerías de procesamiento instaladas.		

1.21.2.1.3 Historia de Usuario 3: Entrenamiento del modelo ResNetVD34 con DANN

Tabla 10 *Historia de Usuario 3: Entrenamiento del modelo ResNetVD34 con DANN*

HISTORIA DE USUARIO			
Id de historia	HU03	Título:	Entrenamiento del modelo ResNetVD34 con DANN
Prioridad:	muy alto	Usuario:	Desarrollador del módulo
Riesgo de desarrollo:	alto	Tamaño de la tarea:	grande
Como:	Desarrollador responsable del modelo de inteligencia artificial		

HISTORIA DE USUARIO	
Quiero:	Entrenar un modelo basado en ResNetVD34 y adaptación de dominio DANN.
Para:	Clasificar las hojas en las categorías saludable y salud comprometida, considerando las diferencias entre imágenes públicas y fotografías locales.
Criterios de aceptación:	El entrenamiento deberá ejecutarse sin errores, registrar las pérdidas y métricas, generar un modelo funcional y permitir el ajuste de sus parámetros.
Dependencias:	Conjuntos preprocesados, Python, TensorFlow, Keras y arquitectura DANN configurada.

1.21.2.1.4 Historia de Usuario 4: Evaluación del modelo entrenado

Tabla 11 Historia de Usuario 4: Evaluación del modelo entrenado

HISTORIA DE USUARIO			
Id de historia	HU04	Título:	Evaluación del modelo entrenado
Prioridad:	alto	Usuario:	Desarrollador del módulo
Riesgo de desarrollo:	medio	Tamaño de la tarea:	mediana
Como:	Desarrollador responsable del modelo de inteligencia artificial		
Quiero:	Evaluar el rendimiento del modelo mediante un conjunto de imágenes de prueba		
Para:	Evaluar el rendimiento del modelo mediante un conjunto de imágenes de prueba.		
Criterios de aceptación:	El proceso deberá calcular exactitud, precisión, recall, F1-score, confianza promedio y matriz de confusión.		
Dependencias:	Modelo entrenado y conjunto de prueba independiente.		

1.21.2.1.5 Historia de Usuario 5: Conversión y exportación del modelo

Tabla 12 Historia de Usuario 5: Conversión y exportación del modelo

HISTORIA DE USUARIO			
Id de historia	HU05	Título:	Conversión y exportación del modelo
Prioridad:	alto	Usuario:	Desarrollador del módulo
Riesgo de desarrollo:	medio	Tamaño de la tarea:	mediana
Como:	Desarrollador responsable del módulo		
Quiero:	Convertir el modelo entrenado al formato TensorFlow Lite con optimización float16		
Para:	Ejecutarlo directamente en un dispositivo móvil sin depender de un servidor		
Criterios de aceptación:	El archivo TensorFlow Lite deberá generarse correctamente, mantener resultados similares al modelo original y ser compatible con la aplicación Flutter.		
Dependencias:	Disponibilidad de un modelo previamente entrenado y evaluado.		

1.21.2.1.6 Historia de Usuario 6: Integración del modelo en la aplicación móvil

Tabla 13 Historia de Usuario 6: Integración del modelo en la aplicación móvil

HISTORIA DE USUARIO			
Id de historia	HU06	Título:	Integración del modelo en la aplicación móvil
Prioridad:	muy alto	Usuario:	Desarrollador del módulo
Riesgo de desarrollo:	alto	Tamaño de la tarea:	grande
Como:	Desarrollador de la aplicación móvil		
Quiero:	Incorporar el modelo convertido a TensorFlow Lite en la aplicación móvil creada con Flutter.		
Para:	Permitir que el análisis de las imágenes se ejecute directamente en el dispositivo Android		
Criterios de aceptación:	La aplicación deberá cargar el modelo, aceptar imágenes desde la cámara o la galería, realizar la inferencia sin conexión a Internet y presentar las probabilidades obtenidas.		
Dependencias:	Modelo TensorFlow Lite, proyecto Flutter y paquete tflite_flutter.		

1.21.2.1.7 Historia de Usuario 7: Presentación del resultado e historial local

Tabla 14 Historia de Usuario 7: Presentación del resultado e historial local

HISTORIA DE USUARIO			
Id de historia	HU07	Título:	Presentación del resultado e historial local
Prioridad:	Alto	Usuario:	Usuario de la aplicación
Riesgo de desarrollo:	medio	Tamaño de la tarea:	mediano
Como:	Usuario del módulo móvil		
Quiero:	Visualizar el resultado del análisis y consultar los registros anteriores		
Para:	Conocer el estado visual general de la hoja y conservar un seguimiento de los análisis realizados		
Criterios de aceptación:	la aplicación deberá mostrar las probabilidades de saludable y salud comprometida, presentar no determinada cuando la confianza sea inferior al 65 %, mostrar recomendaciones generales y guardar el resultado en el historial local.		
Dependencias:	Modelo integrado, inferencia funcional y almacenamiento local configurado.		

1.21.2.2 Requerimientos funcionales

RF01: El módulo deberá permitir la carga y organización de las imágenes pertenecientes al dominio fuente y al dominio objetivo.

RF02: El módulo deberá preparar las imágenes tareas de depuración, redimensionamiento, normalización, balanceo, para luego distribuirlas en conjuntos de entrenamiento, validación y prueba.

RF03: El módulo deberá utilizar ResNetVD34 preentrenada con ImageNet para extraer automáticamente las características visuales de cada imagen.

RF04: Durante el entrenamiento se deberá utilizar DANN para ajustar el aprendizaje obtenido con las imágenes publicas a las condiciones presentes en las fotografías tomadas en el campo.

RF05: La evaluación deberá incluir las métricas de exactitud, precisión, recall, F1-score, confianza promedio y matriz de confusión para conocer el desempeño alcanzado por el modelo.

RF06: Una vez entrenado, el modelo deberá convertirse a TensorFlow Lite con optimización float16 y exportarse en un formato compatible con dispositivo móviles.

RF07: La aplicación deberá permitir capturar fotografías de hojas de plátano mediante la cámara del dispositivo móvil.

RF08: La aplicación deberá permitir seleccionar imágenes almacenadas en la galería del dispositivo.

RF09: La aplicación deberá clasificar las imágenes en las categorías saludable y salud comprometida, mostrando las probabilidades obtenidas para cada categoría.

RF10: La aplicación deberá presentar el resultado no determinada cuando la probabilidad máxima sea inferior al umbral de confianza del 65 %.

RF11: La aplicación deberá ofrecer recomendaciones generales de acuerdo con el resultado obtenido, sin diagnosticar enfermedades específicas ni prescribir tratamientos.

RF12: La aplicación deberá guardar en el dispositivo un historial de los análisis y permitir su consulta posterior.

RF13: El módulo deberá conservar una estructura independiente que permita incorporarlo con la aplicación principal del Proyecto Integrador.

1.21.2.3 Requerimientos no funcionales

RNF01: El módulo deberá mantener la correspondencia correcta de cada imagen y de sus etiquetas durante las etapas de preparación, entrenamiento y evaluación.

RNF02: El proceso de entrenamiento deberá desarrollarse en un entorno reproducible, utilizando versiones compatibles de Python, TensorFlow, Keras y las demás bibliotecas necesarias.

RNF03: La aplicación deberá ejecutar el modelo directamente en el dispositivo móvil sin depender de conexión a Internet ni de un servidor externo.

RNF04: El modelo deberá presentarse en un formato compatible con dispositivos Android mediante TensorFlow Lite.

RNF05: La aplicación deberá ofrecer una interfaz clara y fácil de comprender para poder utilizar la cámara, la galería, los resultados y el historial.

RNF06: El módulo deberá mostrar las probabilidades y el resultado del análisis sin alterar ni ocultar el nivel de confianza generado por el modelo.

RNF07: La arquitectura deberá permitir añadir nuevas imágenes y versiones del modelo mediante un nuevo proceso de entrenamiento.

RNF08: El módulo deberá conservar una organización modular que facilite su mantenimiento y conexión con otros componentes del Proyecto Integrador.

RNF09: Los resultados del modelo deberán poder evaluarse con métricas de clasificación reconocida y conservarse como evidencia del rendimiento del modelo.

RNF10: La aplicación deberá guardar el historial únicamente en el dispositivo móvil, sin depender de base de datos externa.

1.21.3 Diseño del sistema

1.21.3.1 Casos de uso

1.21.3.1.1 Caso de uso: Preparación del conjunto de imágenes

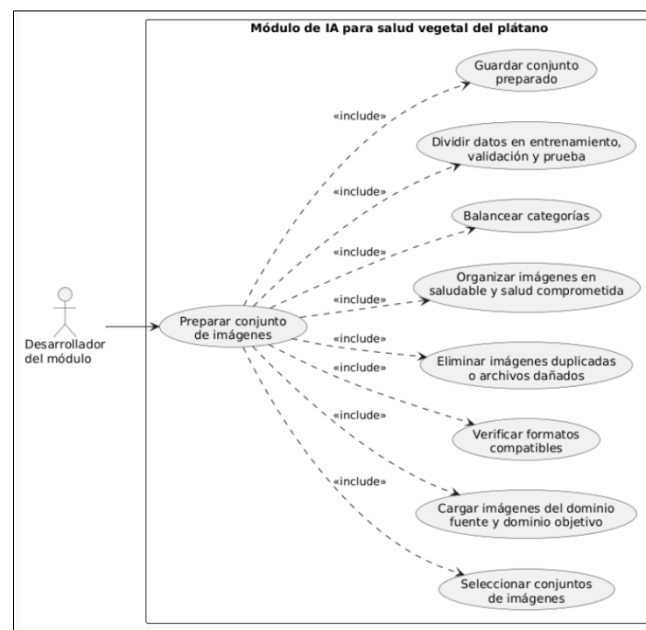


Ilustración 2 Preparación del conjunto de imágenes

1.21.3.1.2 Caso de uso: Entrenamiento y evaluación del modelo

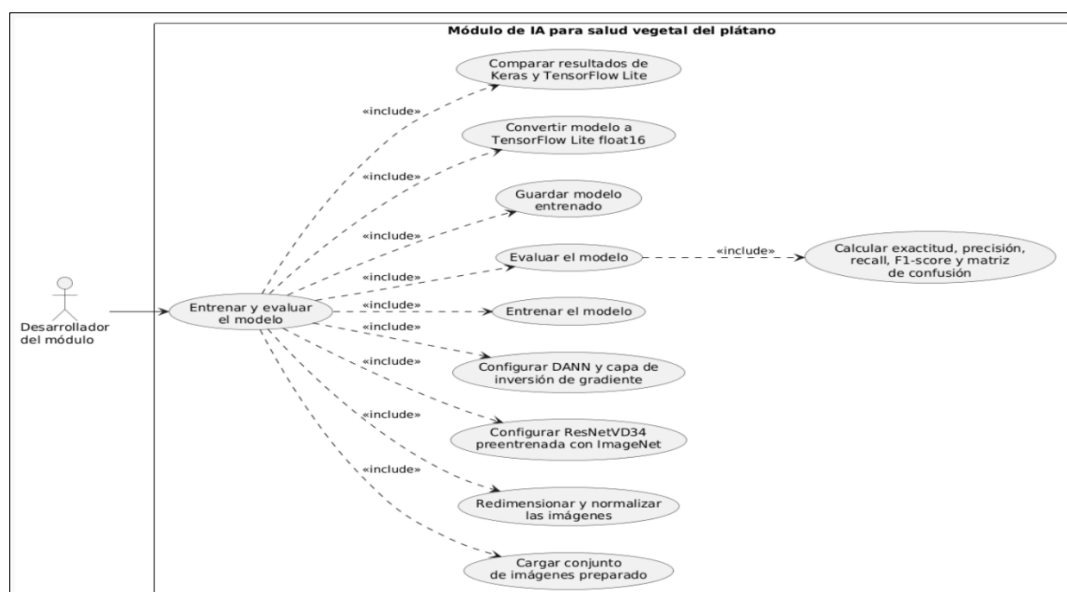


Ilustración 3 Entrenamiento y evaluación del modelo

1.21.3.1.3 Caso de uso: Análisis de una imagen y presentación de resultados

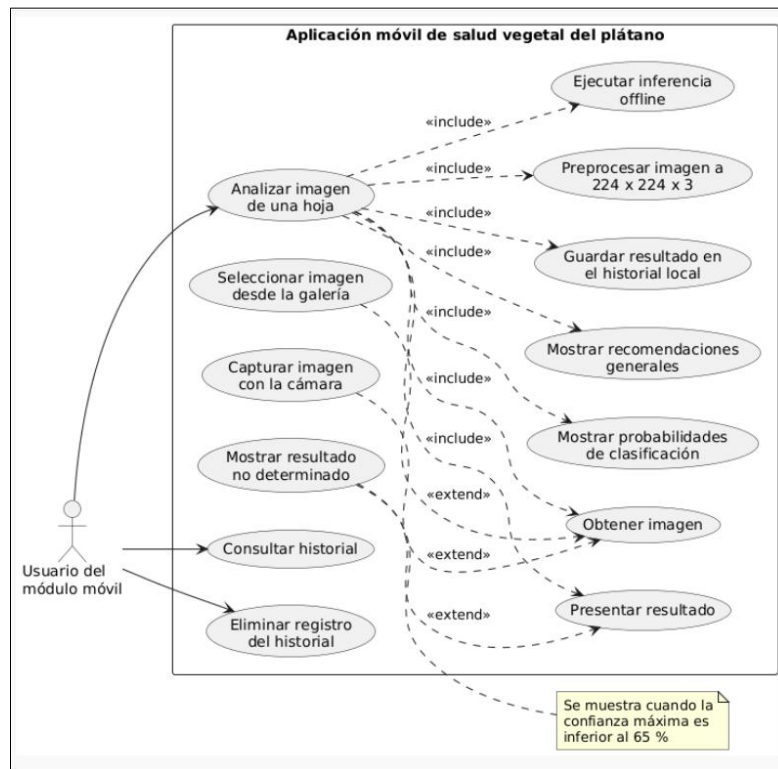


Ilustración 4 Análisis de una imagen y presentación de resultados

1.21.3.2 Diagramas de secuencias

1.21.3.2.1 Diagrama de secuencia Preparación del conjunto de imágenes

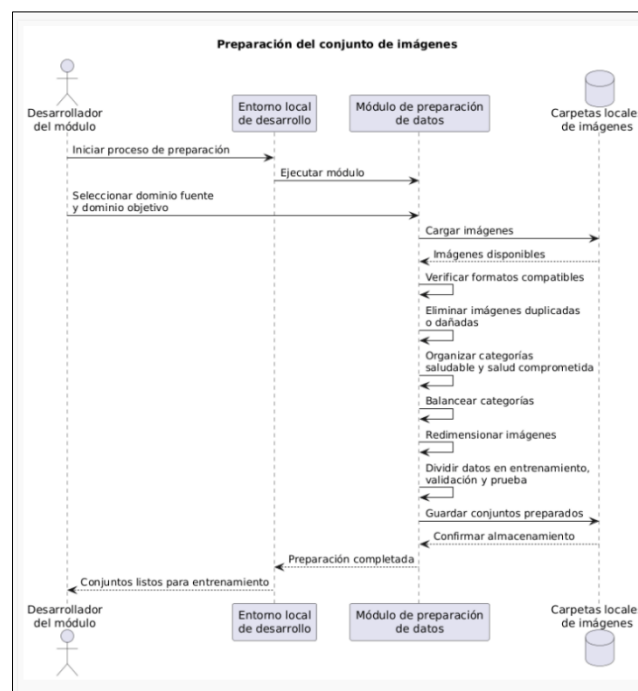


Ilustración 5 Diagrama de secuencia Preparación del conjunto de imágenes

1.21.3.2.2 Diagrama de secuencia: Entrenamiento, evaluación y conversión del modelo

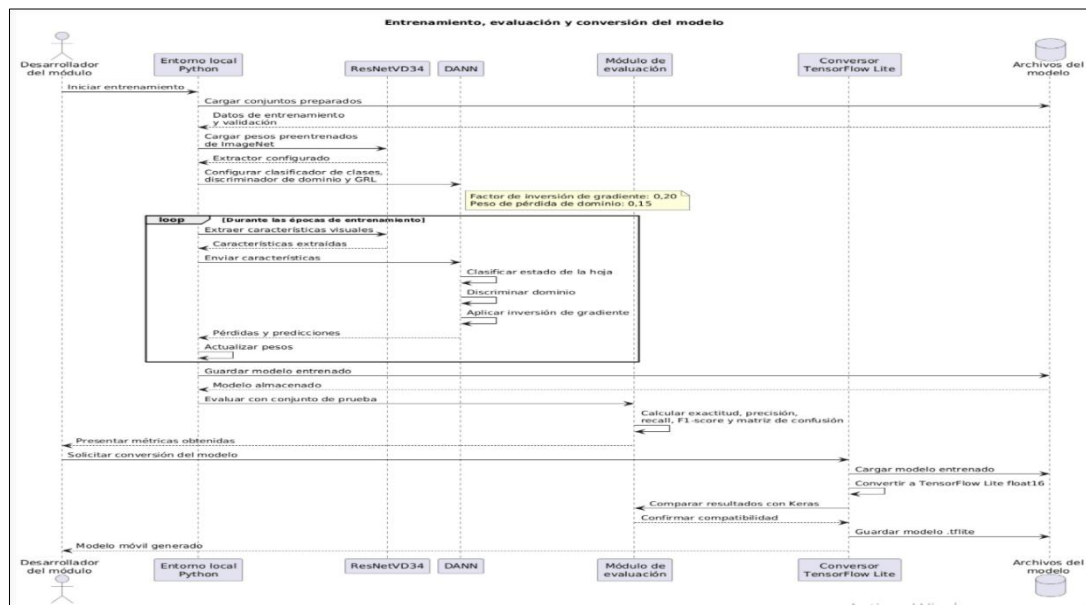


Ilustración 6 Diagrama de secuencia: Entrenamiento, evaluación y conversión del modelo

1.21.3.2.3 Diagrama de secuencia: Análisis de imagen en la aplicación móvil

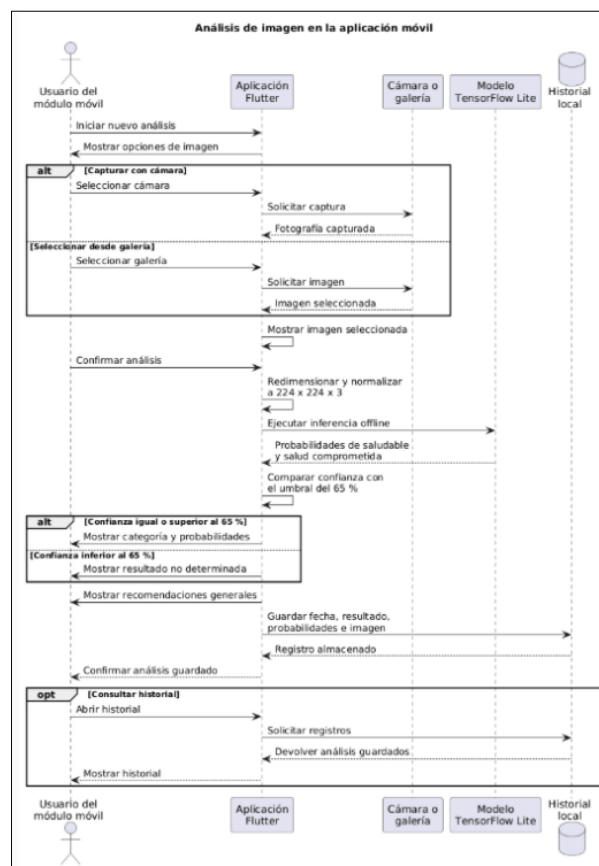


Ilustración 7 Diagrama de secuencia: Análisis de imagen en la aplicación móvil

1.21.3.3 Diagramas de Estado

1.21.3.3.1 Diagrama de estado: Preparación del conjunto de imágenes

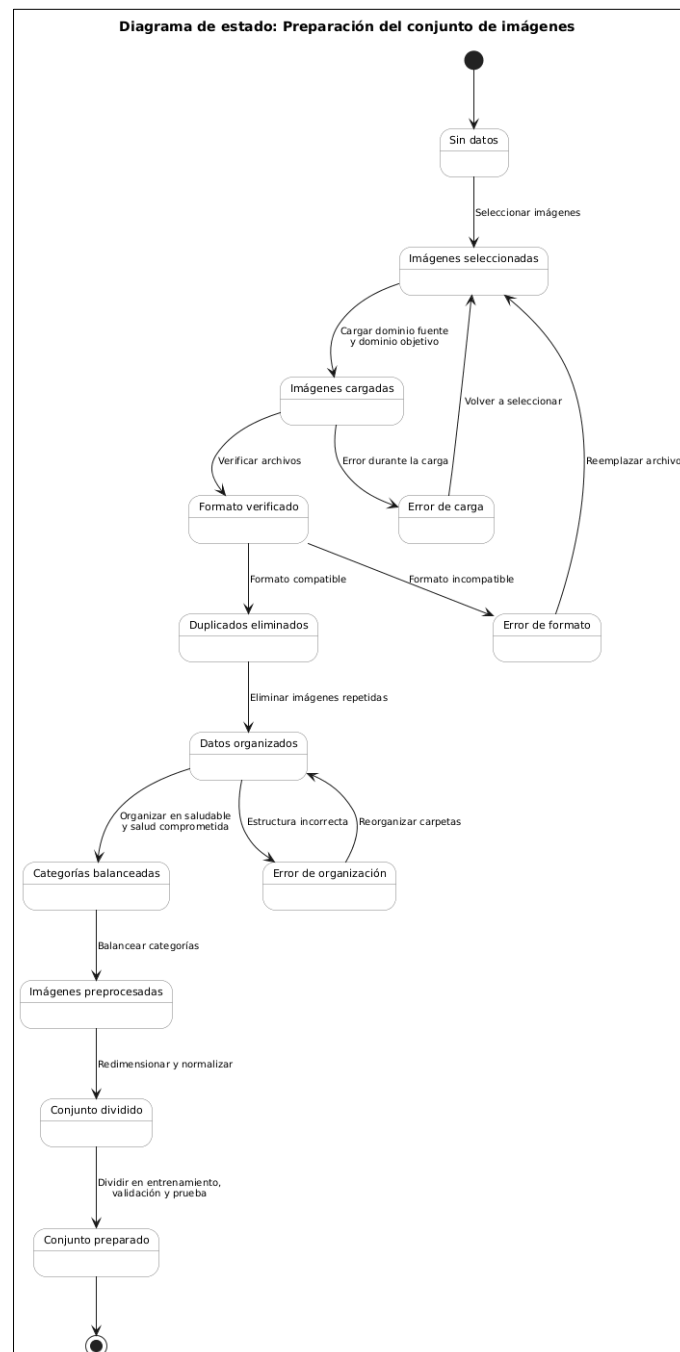


Ilustración 8 diagrama de estado: Preparación del conjunto de imágenes

1.21.3.3.2 Diagrama de estado: Entrenamiento y evaluación del modelo

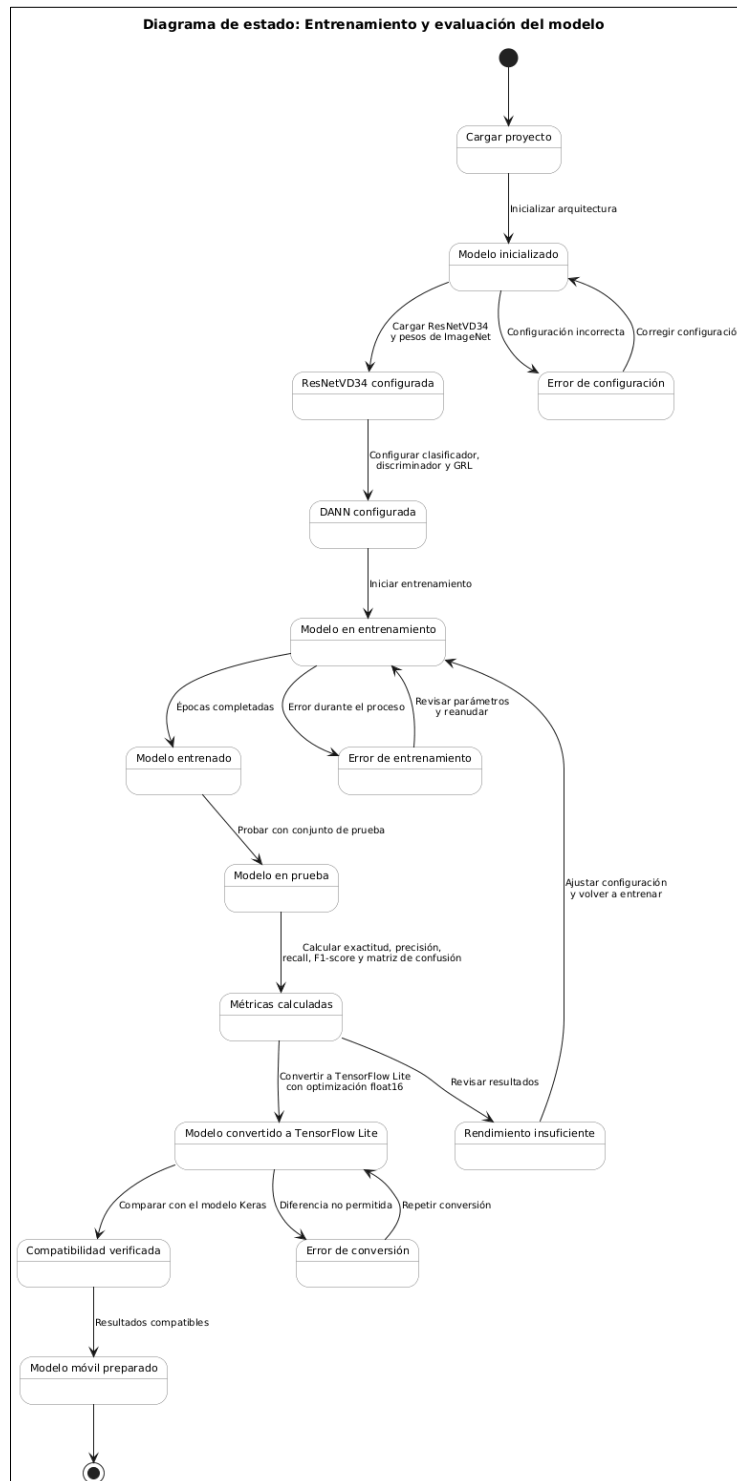


Ilustración 9 Diagrama de estado: Entrenamiento y evaluación del modelo

1.21.3.3 Diagrama de estado: Análisis de imagen y presentación de resultados

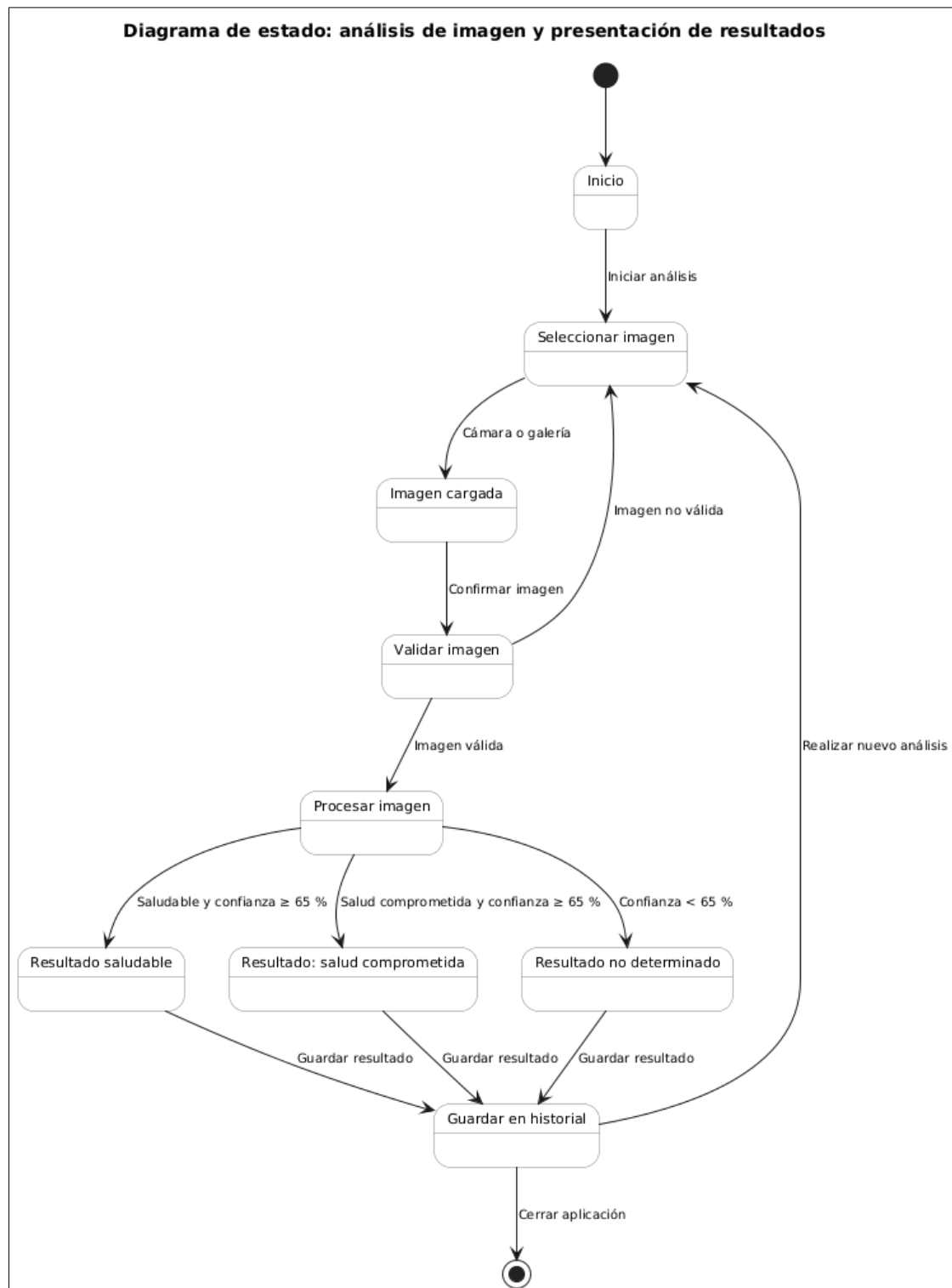


Ilustración 10 Diagrama de estado: Análisis de imagen y presentación de resultados

1.21.4 Arquitectura del sistema

1.21.4.1 Arquitectura general

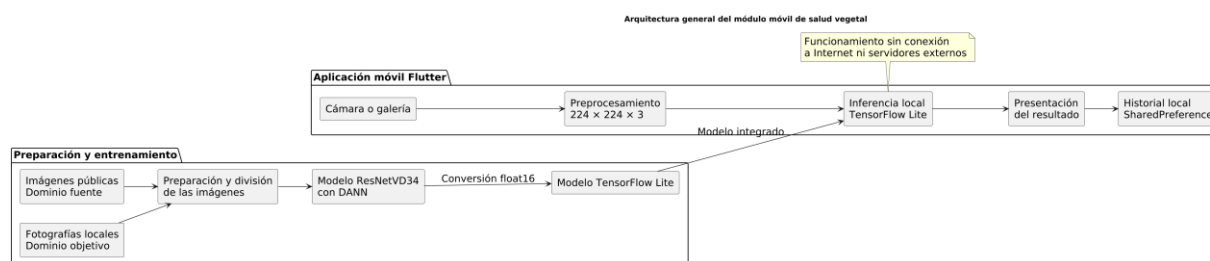


Ilustración 11 Arquitectura general del módulo de Inteligencia Artificial

La arquitectura general del módulo se estructuró en dos etapas principales: el entrenamiento del modelo de inteligencia artificial y su implementación dentro de la aplicación móvil, en esta primera etapa se trabajó con imágenes públicas como dominio fuente y con fotografías tomadas en condiciones reales de campo como dominio objetivo.

Antes de ingresar al modelo, las imágenes se revisaron para eliminar duplicados, la organizar las categorías, equilibrar las clases y ajustar sus dimensiones, después se distribuyeron en conjuntos de entrenamiento, validación y prueba. ResNetVD34, con pesos preentrenados en ImageNet, se empleó para extraer sus características visuales.

DANN incorpora dos componentes: un clasificador y un discriminador de dominio, relacionados mediante una capa de inversión de gradiente, el clasificador aprendió a diferenciar entre hojas saludables y hojas con salud comprometida, mientras que el discriminador ayudó a reducir las diferencias visuales entre las imágenes públicas y las fotografías capturadas localmente.

Después de completar el entrenamiento, se revisó el desempeño del modelo mediante las métricas de calificación, con los resultados obtenidos se generó una versión en TensorFlow lite aplicando optimización float16 y el archivo resultante se incorporó a la aplicación creada con Flutter.

Para comenzar un análisis, el usuario puede tomar una fotografía con la cámara o elegir una imagen almacenada en la galería, antes de ingresar al modelo la aplicación ajusta la imagen a un tamaño de 224 x 224 píxeles con tres canales de color, todo el procesamiento se realiza en el teléfono sin requerir conexión a internet.

Al terminar el análisis, la pantalla presenta las probabilidades calculadas para las categorías saludable y salud comprometida, si ninguna alcanza el 65 % de confianza, aparece el resultado no determinado, cada revisión queda registrada en el historial local además, el

módulo mantiene una estructura independiente para facilitar su futura incorporación a la aplicación principal del Proyecto Integrador.

1.21.4.2 Tecnologías Utilizadas

Tabla 15 Tecnologías Utilizadas

Herramienta	Uso dentro del proyecto
Python	Se utilizó para programar el módulo de la inteligencia artificial y preparar las imágenes antes del entrenamiento.
Visual Studio Code	Se utilizó para desarrollar el modelo de inteligencia artificial y la aplicación móvil.
TensorFlow y Keras	Permitieron implementar el modelo, realizar el entrenamiento y calcular las métricas utilizadas durante su evaluación.
Gradient Reversal Layer	Se empleó durante la adaptación de dominio para invertir el gradiente enviado al discriminador
ResNet V3	Se utilizó como extractor de características visuales mediante pesos aprendidos previamente con ImageNet.
Domain-Adversarial Neural Networks (DANN)	Estrategia de adaptación de dominio utilizada para reducir las diferencias entre las imágenes públicas y las fotografías locales.
NumPy	Biblioteca utilizada para realizar operaciones numéricas y manipular los datos empleados durante el entrenamiento y la evaluación.
Flutter y Dart	Tecnologías utilizadas para desarrollar la interfaz y las funcionalidades de la aplicación móvil..
TensorFlow Lite	Herramientas utilizadas para controlar las versiones, respaldar el código y compartir el módulo con el integrante responsable de la aplicación principal.
SharedPreferences	Herramienta utilizada para almacenar localmente el historial de los análisis realizados.
GitHub	Plataforma utilizada para el control de versiones y almacenamiento del código fuente del proyecto.

1.21.5 Roles Scrum

Tabla 16 Roles Scrum

Rol Scrum	Participante(s)	Función principal del proyecto
Product Owner	Ing. Carlos Vinicio López Rodríguez	Define y revisa el alcance académico del módulo, orienta los objetivos, prioriza los requerimientos y verifica que los avances desarrollados mantengan relación con los objetivos de la investigación.
Scrum Master	Jean Fredd Zambrano Quiroz	Encargado de organizar la aplicación de Scrum, planifica las actividades de cada sprint, da seguimiento al cumplimiento de las tareas e identifica dificultades que puedan afectar el avance del proyecto.
Scrum Team	Jean Fredd Zambrano Quiroz	Es el responsable de la preparación de las imágenes, diseño, programación, entrenamiento y evaluación del modelo basado en ResNetVD34 y DANN, conversión a TensorFlow Lite, desarrollo de la aplicación Flutter, realización de pruebas y documentación del módulo.

1.21.6 Planificación del Sprint

1.21.6.1 Sprint 0: Preparación del entorno y preprocesamiento del conjunto de imágenes.

Tabla 17 Sprint 0: Preparación del entorno y preprocesamiento del conjunto de imágenes.

SPRINT 0: Preparación del entorno y preprocesamiento del conjunto de imágenes			
Duración:	2 semanas (40 horas)	Prioridad:	Alta
Objetivo:	Preparar el entorno local de desarrollo y organizar los conjuntos de imágenes del dominio fuente y del dominio objetivo para el entrenamiento del modelo.		
Historias de usuario:	HU01: Carga y organización del conjunto de imágenes. HU02: Preprocesamiento del conjunto de imágenes.		

SPRINT 0: Preparación del entorno y preprocesamiento del conjunto de imágenes	
Tareas de desarrollo:	• Configuración del entorno local en Visual Studio Code • Instalación de Python 3.11, TensorFlow, Keras y las bibliotecas necesarias. • Recopilación de las imágenes públicas y fotografías locales. • Verificación de formatos. • Eliminación de imágenes duplicadas o dañadas. • Organización de las categorías saludable y salud comprometida. • Balanceo de las categorías. • Redimensionamiento y normalización. • División en entrenamiento, validación y prueba.
Plan de entrega:	• Configuración del entorno local. • Se configurará Visual Studio Code, Python 3.11 y las bibliotecas necesarias para desarrollar y entrenar el modelo. Tiempo de desarrollo: 6 horas. • Carga y organización del conjunto de imágenes. • Se recopilarán y organizarán las imágenes del dominio fuente y del dominio objetivo, identificando correctamente cada categoría Tiempo de desarrollo: 12 horas. • Depuración y balanceo de las imágenes • Se eliminarán archivos duplicados o incompatibles y se equilibrará la cantidad de imágenes utilizadas por categoría. Tiempo de desarrollo: 12 horas. • Preprocesamiento y división de los datos. • Las imágenes serán redimensionadas, normalizadas y separadas en conjuntos de entrenamiento, validación y prueba. Tiempo de desarrollo: 10 horas.

1.21.6.2 Sprint 1: Desarrollo y entrenamiento del modelo

Tabla 18 Sprint 1: Desarrollo y entrenamiento del modelo

SPRINT 1: Desarrollo y entrenamiento del modelo			
Duración:	2 semanas (55 horas)	Prioridad:	Muy Alta
Objetivo:	Desarrollar y entrenar el modelo basado en ResNetVD34 y adaptación de dominio DANN para clasificar el estado general de las hojas de plátano.		
Historias de usuario:	HU03: Entrenamiento del modelo ResNetVD34 con DANN.		
Tareas de desarrollo:	• Configuración de ResNetVD34 con pesos de ImageNet. • Implementación del clasificador de clases. • Implementación del clasificador de clases. • Implementación del clasificador de clases. • Configuración de los parámetros de entrenamiento. • Ejecución del entrenamiento inicial. • Ajuste fino de las últimas capas. • Almacenamiento del modelo entrenado.		
Plan de entrega:	• Configuración de ResNetVD34. • Se configurará la arquitectura preentrenada como extractor de características visuales. Tiempo de desarrollo: 15 horas. • Implementación de DANN • Se integrarán el clasificador, el discriminador de dominio y la capa de inversión de gradiente. Tiempo de desarrollo: 12 horas. • Entrenamiento del modelo. • Se entrenará el modelo utilizando las imágenes del dominio fuente y del dominio objetivo. Tiempo de desarrollo: 18 horas. • Ajuste fino y almacenamiento. • Se habilitarán las últimas capas del extractor, se ajustarán los parámetros y se guardará el modelo entrenado. Tiempo de desarrollo: 10 horas.		

1.21.6.3 Sprint 2: Evaluación y conversión del modelo

Tabla 19 Sprint 2: Evaluación y conversión del modelo

SPRINT 2: Evaluación y conversión del modelo			
Duración:	2 semanas (45 horas)	Prioridad:	Alta
Objetivo:	Evaluar el rendimiento del modelo y convertirlo al formato TensorFlow Lite con optimización float16.		
Historias de usuario:	HU04: Evaluación del modelo entrenado. HU05: Conversión y exportación del modelo.		
Tareas de desarrollo:	• Evaluación con el conjunto de prueba. • Cálculo de exactitud, precisión, recall y F1-score. • Generación de la matriz de confusión. • Cálculo de la confianza promedio. • Análisis de errores de clasificación. • Conversión a TensorFlow Lite float16. • Comparación entre Keras y TensorFlow Lite. • Almacenamiento del archivo. tflite.		
Plan de entrega:	• Evaluación del modelo • Se realizarán pruebas con las imágenes reservadas y se obtendrán las predicciones finales. Tiempo de desarrollo: 14 horas • Cálculo de métricas. • Se calcularán las métricas de clasificación y la matriz de confusión. Tiempo de desarrollo: 10 horas. • Conversión a TensorFlow Lite • El modelo será convertido al formato móvil con optimización float16. Tiempo de desarrollo: 11 horas. • Verificación de equivalencia • Se compararán las salidas del modelo Keras y TensorFlow Lite para comprobar que la conversión mantenga resultados		

SPRINT 2: Evaluación y conversión del modelo	
	similares. Tiempo de desarrollo: 10 horas.

1.21.6.4 Sprint 3: Integración móvil y pruebas finales

Tabla 20 Sprint 3 Integración móvil y pruebas finales

SPRINT 3: Integración móvil y pruebas finales			
Duración:	2 semanas (60 horas)	Prioridad:	Muy Alta
Objetivo:	Integrar el modelo TensorFlow Lite en Flutter e implementar las funcionalidades necesarias para ejecutar el análisis offline.		
Historias de usuario:	HU06: Integración del modelo en la aplicación móvil. HU07: Presentación del resultado e historial local.		
Tareas de desarrollo:	• Desarrollo de la interfaz en Flutter. • Integración del modelo mediante tflite_flutter. • Implementación de cámara y galería. • Preprocesamiento de imágenes en el dispositivo. • Ejecución de inferencia offline. • Presentación de probabilidades. • Aplicación del umbral del 65 %. • Presentación de recomendaciones generales. • Almacenamiento y consulta del historial local. • Pruebas en el teléfono Infinix X6858. • Generación e instalación del APK. • Publicación del código en GitHub. • Elaboración de la documentación técnica.		
Plan de entrega:	• Desarrollo de la interfaz. • Desarrollo de la interfaz • Tiempo de desarrollo: 15 horas. • Integración del modelo y selección de imágenes		

SPRINT 3: Integración móvil y pruebas finales	
	- Se integrará TensorFlow Lite y se implementarán las funciones de cámara, galería e inferencia offline. Tiempo de desarrollo: 18 horas. Resultados e historial local - Se mostrarán las probabilidades, el resultado, las recomendaciones y los registros anteriores. Tiempo de desarrollo: 12 horas. Pruebas y documentación - Se probará la aplicación en Android, se generará el APK y se documentará el módulo para su integración. Tiempo de desarrollo: 15 horas

1.21.7 Product Backlog

1.21.7.1 Backlog inicial

Tabla 21 Backlog Inicial

N.º	Tarea	Historia de usuario	Prioridad	Estado
1	Configurar el entorno local en Visual Studio Code con Python 3.11 y las bibliotecas necesarias.	HU01	Alta	Por hacer
2	Recopilar las imágenes públicas y las fotografías locales.	HU01	Alta	Por hacer
3	Verificar los formatos y eliminar imágenes duplicadas o dañadas.	HU01	Alta	Por hacer
4	Organizar las imágenes según el dominio fuente, dominio objetivo y categoría correspondiente.	HU01	Alta	Por hacer
5	Balancear las categorías saludables y salud comprometida.	HU02	Alta	Por hacer
6	Redimensionar y normalizar las imágenes.	HU02	Alta	Por hacer
7	Dividir los datos en conjuntos de entrenamiento, validación y prueba.	HU02	Alta	Por hacer

N.º	Tarea	Historia de usuario	Prioridad	Estado
8	Verificar que el conjunto de imágenes quede preparado correctamente.	HU02	Alta	Por hacer
9	Configurar ResNetVD34 con pesos preentrenados de ImageNet.	HU03	Muy alta	Por hacer
10	Implementar el clasificador para saludable y salud comprometida.	HU03	Muy alta	Por hacer
11	Implementar el discriminador de dominio y la capa de inversión de gradiente.	HU03	Muy alta	Por hacer
12	Configurar los parámetros de entrenamiento del modelo DANN.	HU03	Alta	Por hacer
13	Entrenar, ajustar y almacenar el modelo desarrollado.	HU03	Muy alta	Por hacer
14	Evaluar el modelo mediante el conjunto de imágenes de prueba.	HU04	Alta	Por hacer
15	Calcular exactitud, precisión, recall, F1-score y matriz de confusión.	HU04	Alta	Por hacer
16	Analizar las predicciones incorrectas y la confianza obtenida.	HU04	Alta	Por hacer
17	Convertir el modelo al formato TensorFlow Lite con optimización float16.	HU05	Alta	Por hacer
18	Comparar las predicciones de Keras y TensorFlow Lite y guardar el archivo .tflite.	HU05	Alta	Por hacer
19	Desarrollar la interfaz independiente del módulo en Flutter.	HU06	Alta	Por hacer
20	Integrar el modelo mediante el paquete tflite_flutter.	HU06	Muy alta	Por hacer
21	Implementar cámara, galería y ejecución offline de la inferencia.	HU06	Muy alta	Por hacer
22	Mostrar las probabilidades y aplicar el umbral de confianza del 65 %.	HU07	Alta	Por hacer

N.º	Tarea	Historia de usuario	Prioridad	Estado
23	Presentar recomendaciones y almacenar el historial local.	HU07	Alta	Por hacer
24	Probar la aplicación en Android, generar el APK y preparar el módulo para su integración.	HU06 y HU07	Alta	Por hacer

1.21.7.2 Sprint Backlog

1.21.7.2.1 Sprint Backlog: Preparación del entorno y del conjunto de imágenes

Tabla 22 Sprint Backlog: Preparación del entorno y del conjunto de imágenes

Historia de usuario.	Tarea	Duración estimada
HU01	Configurar el entorno local en Visual Studio Code con Python, TensorFlow y Keras.	6 horas
HU01	Recopilar las imágenes públicas y las fotografías locales.	8 horas
HU01	Verificar formatos y eliminar imágenes duplicadas o dañadas.	8 horas
HU01	Organizar las imágenes según el dominio y la categoría correspondiente.	6 horas
HU02	Balancear las categorías saludables y salud comprometida.	4 horas
HU02	Redimensionar y normalizar las imágenes.	4 horas
HU02	Redimensionar y normalizar las imágenes.	6 horas

1.21.7.2.2 Sprint Backlog: Desarrollo y entrenamiento del modelo

Tabla 23 Sprint Backlog: Desarrollo y entrenamiento del modelo

Historia de usuario.	Tarea	Duración estimada
HU03	Configurar ResNetVD34 con pesos preentrenados de ImageNet.	10 horas
HU03	Implementar el clasificador para las categorías saludable y salud comprometida.	8 horas
HU03	Implementar el discriminador de dominio y la capa de inversión de gradiente.	12 horas
HU03	Configurar los parámetros de entrenamiento del modelo DANN.	6 horas
HU03	Ejecutar el entrenamiento inicial del modelo.	10 horas
HU03	Realizar el ajuste fino y almacenar el modelo entrenado.	9 horas

1.21.7.2.3 Sprint Backlog: Evaluación y conversión del modelo

Tabla 24 Sprint Backlog: Evaluación y conversión del modelo

Historia de usuario.	Tarea	Duración estimada
HU04	Evaluar el modelo utilizando el conjunto de imágenes de prueba.	8 horas
HU04	Calcular exactitud, precisión, recall, F1-score y matriz de confusión.	8 horas
HU04	Analizar los errores de clasificación y la confianza promedio.	6 horas
HU05	Analizar los errores de clasificación y la confianza promedio.	10 horas
HU05	Comparar las predicciones de Keras y TensorFlow Lite.	8 horas
HU05	Guardar y documentar el archivo final del modelo móvil.	5 horas

1.21.7.2.4 Sprint Backlog: Integración móvil y pruebas finales

Tabla 25 Sprint Backlog: Integración móvil y pruebas finales

Historia de usuario.	Tarea	Duración estimada
HU06	Desarrollar la interfaz independiente del módulo en Flutter.	12 horas
HU06	Desarrollar la interfaz independiente del módulo en Flutter.	12 horas
HU06	Implementar cámara, galería y preprocesamiento de imágenes.	12 horas
HU07	Mostrar probabilidades, aplicar el umbral del 65 % y presentar recomendaciones. .	8 horas
HU07	Implementar el almacenamiento y consulta del historial local. .	6 horas
HU07	Probar la aplicación en Android, generar el APK y preparar el módulo para su integración.	10 horas

1.21.8 Desarrollo de los sprints

El desarrollo del módulo móvil de inteligencia artificial para la evaluación del estado general de salud vegetal de hojas de plátano se organizó en cuatro sprints, comprendidos desde el Sprint 0 hasta el Sprint 3. Cada sprint estuvo orientado al cumplimiento de las actividades específicas entrelazadas con la preparación del entorno local, organización de los conjuntos de imágenes, desarrollo y entrenamiento del modelo, evaluación de resultados, conversión a TensorFlow Lite e implementación en una aplicación Flutter independiente.

Esta planificación permitió desarrollar el módulo de manera progresiva y verificar los funcionamientos de los componentes implementados en cada etapa, los resultados obtenidos en un sprint sirvieron como base para continuar con las actividades del siguiente, desde la preparación de los datos hasta las pruebas finales realizadas en el dispositivo móvil.

Una vez finalizados los cuatro sprints, se obtuvo un módulo funcional con capacidad para utilizar fotografías de la cámara o la galería y ejecutar la inferencia sin conexión a Internet, presentar las probabilidades de clasificación y guardar localmente el historial de análisis. El módulo quedó preparado para su posterior integración dentro de la aplicación principal del proyecto integrador.

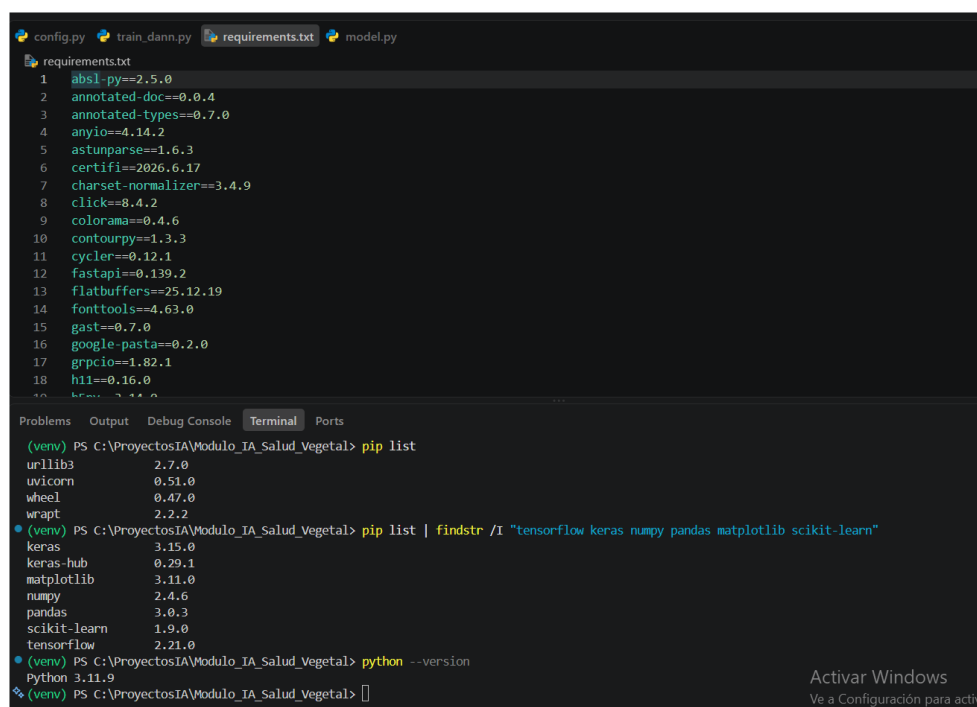
Para comprobar el funcionamiento del módulo, se desarrolló una aplicación móvil independiente, su incorporación en la aplicación principal del proyecto integrador se realizará posteriormente por parte del responsable de dicha interfaz.

1.21.8.1 Desarrollo del Sprint 0: Preparación del entorno y dataset

En el Sprint 0 se configuró el entorno local utilizado para desarrollar el módulo de inteligencia artificial, las tareas de preparación de imágenes, entrenamiento y evaluación del modelo se ejecutaron con Python 3.11 desde Visual Studio Code.

También se creó un entorno virtual de Python para mantener separadas y controladas las dependencias, en el cual se instalaron TensorFlow, Keras, NumPy, Pandas, Matplotlib y Scikit-learn, estas bibliotecas se utilizaron para construir el modelo, procesar las imágenes, revisar los resultados y calcular las métricas de evaluación

La configuración del entorno local y la activación del entorno virtual se evidenciaron en la Ilustración 12.



The screenshot shows the Visual Studio Code interface with a file explorer on the left displaying files: config.py, train_dann.py, requirements.txt, and model.py. The main editor area shows the contents of requirements.txt, which lists various dependencies with version constraints. Below the editor, the Terminal panel is active, showing the execution of pip commands to list installed packages and verify the installation of specific libraries. The output of the pip list command shows the following installed packages and versions:

Package	Version
urllib3	2.7.0
uvicorn	0.51.0
wheel	0.47.0
wrapt	2.2.2
keras	3.15.0
keras-hub	0.29.1
matplotlib	3.11.0
numpy	2.4.6
pandas	3.0.3
scikit-learn	1.9.0
tensorflow	2.21.0

The terminal also shows the output of the python --version command, which is Python 3.11.9. The bottom right corner of the terminal shows a Windows notification: "Activar Windows Ve a Configuración para activar Windows".

Ilustración 12 Configuración del entorno virtual y verificación de las dependencias instaladas

También se estableció una estructura de carpetas para organizar los conjuntos de imágenes, los archivos de entrenamiento, los modelos generados, los resultados obtenidos y los códigos fuente. Esta distribución permitió mantener separados los recursos utilizados en cada etapa y facilitó la ejecución ordenada del proyecto.

La estructura final de carpetas establecida para organizar los conjuntos de imágenes del dominio fuente y del dominio objetivo se presentó en la Ilustración 13.

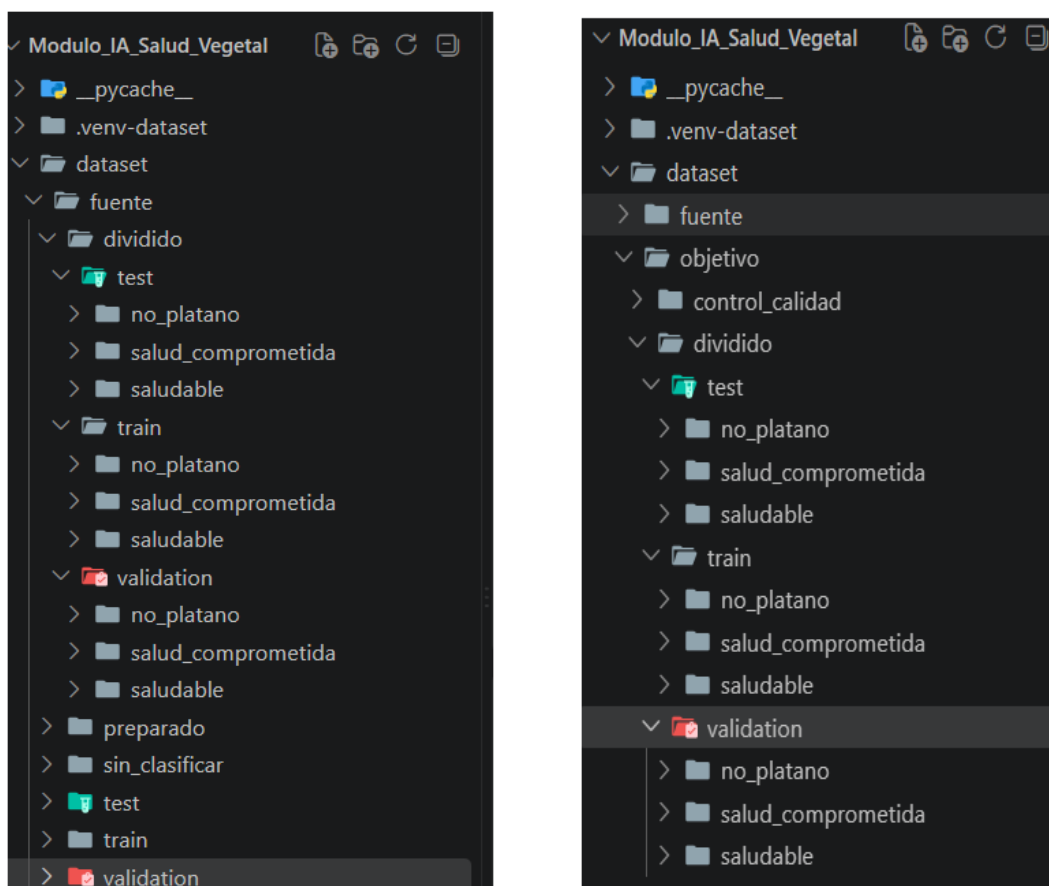


Ilustración 13 Estructura final de carpetas de los conjuntos de imágenes

En esta etapa se trabajó con dos grupos de imágenes. El dominio fuente estuvo conformado por fotografías obtenidas del conjunto público Images of Nutrient Deficient Banana Plant Leaves. El dominio objetivo estuvo compuesto por fotografías capturadas en condiciones reales de campo.

Una muestra de las imágenes utilizadas como dominio fuente se presentó en la Ilustración 14.

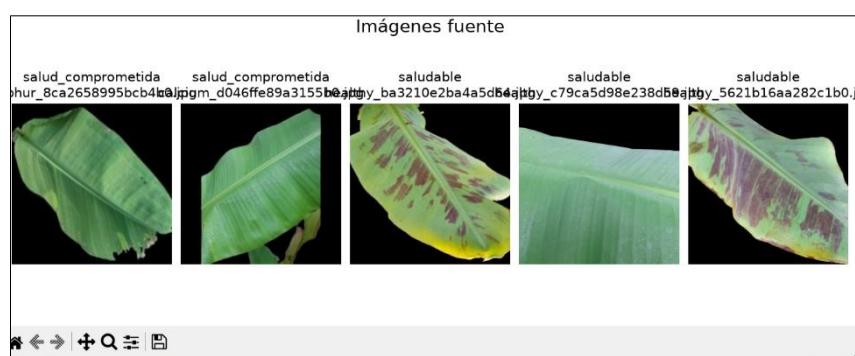


Ilustración 14 Muestra de las imágenes del dominio fuente utilizadas en la clasificación

El conjunto público fue sometido a un proceso de revisión y depuración, mediante el cual se excluyeron archivos duplicados o incompatibles. Posteriormente, se seleccionaron 1.575 imágenes de hojas saludables y 1.575 imágenes de hojas con salud comprometida, con el propósito de mantener el equilibrio entre ambas categorías.

Las imágenes del dominio fuente se dividieron aproximadamente en 1.102 fotografías para entrenamiento, 236 para validación y 237 para prueba por cada categoría. Esta separación permitió utilizar datos diferentes durante el aprendizaje y la evaluación del modelo.

Para el dominio objetivo se recopilaron inicialmente 74 fotografías. Estas imágenes fueron revisadas considerando el enfoque, la visibilidad de la hoja, la iluminación y la presencia de elementos que pudieran dificultar el análisis. Finalmente, se utilizaron 64 fotografías para la clasificación binaria, distribuidas en 31 hojas saludables y 33 hojas con salud comprometida.

Las imágenes locales se dividieron en 44 fotografías para entrenamiento, 8 para validación y 12 para prueba. El conjunto de prueba estuvo balanceado con 6 imágenes saludables y 6 imágenes con salud comprometida.

La distribución final de las imágenes del dominio fuente y del dominio objetivo se verificó mediante el conteo presentado en la Ilustración 15.

```
===== DOMINIO FUENTE =====
train | saludable : 1102
train | salud_comprometida : 1102
validation | saludable : 236
validation | salud_comprometida : 236
test | saludable : 237
test | salud_comprometida : 237

TOTALES DEL DOMINIO FUENTE
saludable : 1575
salud_comprometida : 1575

===== DOMINIO OBJETIVO =====
train | saludable : 21
train | salud_comprometida : 23
Total train : 44
validation | saludable : 4
validation | salud_comprometida : 4
Total validation : 8
test | saludable : 6
test | salud_comprometida : 6
Total test : 12

TOTAL DE FOTOGRAFIAS DEL DOMINIO OBJETIVO: 64
```

Ilustración 15 Verificación de la distribución de los conjuntos de imágenes

Con las evidencias presentadas se comprobó el cumplimiento de los entregables establecidos para el Sprint 0. Como resultado, se obtuvo un entorno local configurado y un conjunto de imágenes revisado, organizado, balanceado y dividido en datos de entrenamiento, validación y prueba, el cual quedó preparado para iniciar el entrenamiento del modelo durante el siguiente sprint.

1.21.8.2 Desarrollo del Sprint 1: Entrenamiento inicial del modelo

Durante el Sprint 1 se desarrolló y entrenó el modelo de inteligencia artificial orientado a la clasificación del estado general de salud vegetal de las hojas de plátano, para este proceso se utilizó la arquitectura ResNetVD34, inicializada con pesos preentrenados de ImageNet, como extractor de características visuales.

Con esta arquitectura fue posible analizar distintos rasgos visuales de las imágenes entre ellos cambios de color, textura, bordes, manchas y áreas secas, al emplear un modelo preentrenado se aprovecharon características aprendidas previamente, reduciendo la necesidad de iniciar el entrenamiento completamente desde cero.

Junto con ResNetVD34 se implementó una Red Neuronal Adversaria para Adaptación de Dominio, conocida como DANN. Esta estrategia se utilizó para reducir las diferencias existentes entre las imágenes públicas del dominio fuente y las fotografías reales del dominio objetivo, las cuales presentaban variaciones de iluminación, fondos, sombras y orientación.

El modelo estuvo conformado por un extractor de características y dos salidas principales. La primera salida realizó la clasificación de las imágenes en las categorías saludable y salud comprometida. La segunda intentó identificar si las características procedían del dominio fuente o del dominio objetivo.

Entre el extractor de características y el discriminador de dominio se incorporó una capa de inversión de gradiente. Durante el entrenamiento, esta capa permitió desarrollar un proceso adversario orientado a obtener características útiles para la clasificación y menos dependientes de las condiciones particulares de cada dominio.

Antes de ingresar al modelo, las imágenes fueron redimensionadas a un tamaño de 224×224 píxeles y normalizadas de acuerdo con el formato requerido por ResNetVD34. También se prepararon lotes combinados con imágenes públicas y fotografías locales para ejecutar el entrenamiento con ambos dominios.

La implementación del extractor de características mediante bloques residuales se presentó en la Ilustración 16.

```
def construir_resnet34() -> Model:
    """Construye ResNet34 como extractor de características."""

    entrada = layers.Input(
        shape=(ALTO_IMAGEN, ANCHO_IMAGEN, CANALES_IMAGEN),
        name="imagen_entrada",
    )

    salida = layers.Conv2D(
        filters=64,
        kernel_size=7,
        strides=2,
        padding="same",
        use_bias=False,
        name="conv_inicial",
    )(entrada)

    salida = layers.BatchNormalization(
        name="bn_inicial"
    )(salida)

    salida = layers.ReLU(
        name="relu_inicial"
    )(salida)

    salida = layers.MaxPooling2D(
        pool_size=3,
        strides=2,
        padding="same",
        name="max_pool_inicial",
    )(salida)

    # ResNet34 utiliza la distribución 3, 4, 6 y 3.
    salida = construir_etapa_residual(
        entrada=salida,
        filtros=64
```

```
# ResNet34 utiliza la distribución 3, 4, 6 y 3.
    salida = construir_etapa_residual(
        entrada=salida,
        filtros=64,
        cantidad_bloques=3,
        paso_inicial=1,
        nombre="etapa1",
    )

    salida = construir_etapa_residual(
        entrada=salida,
        filtros=128,
        cantidad_bloques=4,
        paso_inicial=2,
        nombre="etapa2",
    )

    salida = construir_etapa_residual(
        entrada=salida,
        filtros=256,
        cantidad_bloques=6,
        paso_inicial=2,
        nombre="etapa3",
    )

    salida = construir_etapa_residual(
        entrada=salida,
        filtros=512,
        cantidad_bloques=3,
        paso_inicial=2,
        nombre="etapa4",
    )
```

Ilustración 16 Implementación del extractor de características ResNet34

La integración del extractor de características con las salidas de clasificación y dominio se evidenció en la Ilustración 17.

```
def construir_modelo_dann() -> Model:
    """Construye el modelo completo ResNet34-DANN."""

    entrada = layers.Input(
        shape=(ALTO_IMAGEN, ANCHO_IMAGEN, CANALES_IMAGEN),
        name="entrada_modelo_dann",
    )

    extractor = construir_resnet34()
    caracteristicas = extractor(entrada)

    salida_clases = construir_clasificador_clases(
        caracteristicas
    )

    salida_dominio = construir_clasificador_dominio(
        caracteristicas
    )

    modelo = Model(
        inputs=entrada,
        outputs={
            "salida_clases": salida_clases,
            "salida_dominio": salida_dominio,
        },
        name="Modelo_DANN_Salud_Vegetal",
    )

    return modelo
```

Ilustración 17 Implementación del modelo ResNet34 con adaptación de dominio DANN

La implementación de la capa de inversión de gradiente utilizada para el aprendizaje adversario se presentó en la Ilustración 18.

```
model.py > bloque_residual
@tf.custom_gradient
def invertir_gradiente(
    entrada: tf.Tensor,
    factor: tf.Tensor,
) -> tuple[tf.Tensor, callable]:
    """Mantiene la entrada y cambia el signo del gradiente."""

    def gradiente(
        gradiente_salida: tf.Tensor,
    ) -> tuple[tf.Tensor, None]:
        return -factor * gradiente_salida, None

    return tf.identity(entrada), gradiente

@tf.keras.utils.register_keras_serializable(
    package="ModuloSaludVegetal"
)
class CapaInversionGradiente(layers.Layer):
    """Capa utilizada para el aprendizaje adversario de dominio."""

    def __init__(
        self,
        factor: float = 1.0,
        **kwargs,
    ) -> None:
        super().__init__(**kwargs)
        self.factor = float(factor)

    def call(self, entrada: tf.Tensor) -> tf.Tensor:
        factor_tensor = tf.cast(
            self.factor,
            dtype=entrada.dtype,
        )

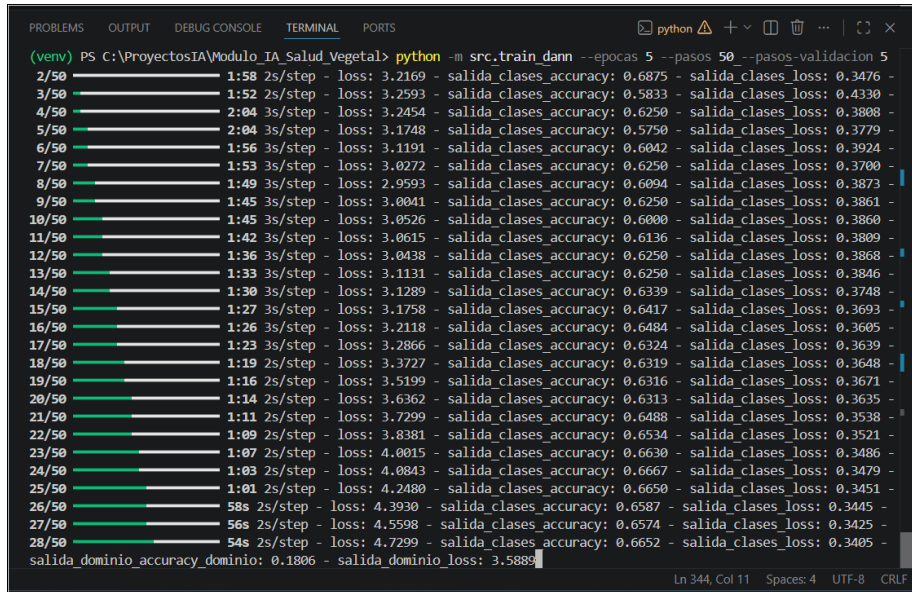
        return invertir_gradiente(
```

Ilustración 18 Implementación de la capa de inversión de gradiente

El entrenamiento se dividió en dos etapas, inicialmente el extractor de características permaneció congelado mientras se entrenaban las capas de clasificación y adaptación de dominio, después se habilitaron las últimas capas de ResNetVD34 para realizar un ajuste fino con una tasa de aprendizaje reducida.

La configuración final empleó un factor de inversión de gradiente de 0,20 y un peso de 0,15 para la pérdida de dominio, con estos valores se buscó limitar la influencia del entrenamiento adversario y conservar como objetivo principal la clasificación de las hojas.

La ejecución de las épocas y el registro de las pérdidas y métricas del entrenamiento se evidenciaron en la Ilustración 17



```
(venv) PS C:\ProyectosIA\Modulo_IA_Salud_Vegetal> python -m src.train_dann --epocas 5 --pasos 50 --pasos-validation 5
2/50 1:58 2s/step - loss: 3.2169 - salida_clases_accuracy: 0.6875 - salida_clases_loss: 0.3476 -
3/50 1:52 2s/step - loss: 3.2593 - salida_clases_accuracy: 0.5833 - salida_clases_loss: 0.4330 -
4/50 2:04 3s/step - loss: 3.2454 - salida_clases_accuracy: 0.6250 - salida_clases_loss: 0.3808 -
5/50 2:04 3s/step - loss: 3.1748 - salida_clases_accuracy: 0.5750 - salida_clases_loss: 0.3779 -
6/50 1:56 3s/step - loss: 3.1191 - salida_clases_accuracy: 0.6042 - salida_clases_loss: 0.3924 -
7/50 1:53 3s/step - loss: 3.0272 - salida_clases_accuracy: 0.6250 - salida_clases_loss: 0.3700 -
8/50 1:49 3s/step - loss: 2.9593 - salida_clases_accuracy: 0.6094 - salida_clases_loss: 0.3873 -
9/50 1:45 3s/step - loss: 3.0041 - salida_clases_accuracy: 0.6250 - salida_clases_loss: 0.3861 -
10/50 1:45 3s/step - loss: 3.0526 - salida_clases_accuracy: 0.6000 - salida_clases_loss: 0.3860 -
11/50 1:42 3s/step - loss: 3.0615 - salida_clases_accuracy: 0.6136 - salida_clases_loss: 0.3809 -
12/50 1:36 3s/step - loss: 3.0438 - salida_clases_accuracy: 0.6250 - salida_clases_loss: 0.3868 -
13/50 1:33 3s/step - loss: 3.1131 - salida_clases_accuracy: 0.6250 - salida_clases_loss: 0.3846 -
14/50 1:30 3s/step - loss: 3.1289 - salida_clases_accuracy: 0.6339 - salida_clases_loss: 0.3748 -
15/50 1:27 3s/step - loss: 3.1758 - salida_clases_accuracy: 0.6417 - salida_clases_loss: 0.3693 -
16/50 1:26 3s/step - loss: 3.2118 - salida_clases_accuracy: 0.6484 - salida_clases_loss: 0.3605 -
17/50 1:23 3s/step - loss: 3.2866 - salida_clases_accuracy: 0.6324 - salida_clases_loss: 0.3639 -
18/50 1:19 2s/step - loss: 3.3727 - salida_clases_accuracy: 0.6319 - salida_clases_loss: 0.3648 -
19/50 1:16 2s/step - loss: 3.5199 - salida_clases_accuracy: 0.6316 - salida_clases_loss: 0.3671 -
20/50 1:14 2s/step - loss: 3.6362 - salida_clases_accuracy: 0.6313 - salida_clases_loss: 0.3635 -
21/50 1:11 2s/step - loss: 3.7299 - salida_clases_accuracy: 0.6488 - salida_clases_loss: 0.3538 -
22/50 1:09 2s/step - loss: 3.8381 - salida_clases_accuracy: 0.6534 - salida_clases_loss: 0.3521 -
23/50 1:07 2s/step - loss: 4.0015 - salida_clases_accuracy: 0.6630 - salida_clases_loss: 0.3486 -
24/50 1:03 2s/step - loss: 4.0843 - salida_clases_accuracy: 0.6667 - salida_clases_loss: 0.3479 -
25/50 1:01 2s/step - loss: 4.2480 - salida_clases_accuracy: 0.6650 - salida_clases_loss: 0.3451 -
26/50 58s 2s/step - loss: 4.3930 - salida_clases_accuracy: 0.6587 - salida_clases_loss: 0.3445 -
27/50 56s 2s/step - loss: 4.5598 - salida_clases_accuracy: 0.6574 - salida_clases_loss: 0.3425 -
28/50 54s 2s/step - loss: 4.7299 - salida_clases_accuracy: 0.6652 - salida_clases_loss: 0.3405 -
salida_dominio_accuracy_dominio: 0.1806 - salida_dominio_loss: 3.5889
```

Ilustración 19 Registro de ejecución del entrenamiento inicial de ResNetVD34 con DANN

Con las evidencias presentadas se comprobó el cumplimiento de los entregables establecidos para el Sprint 1. Como resultado, se obtuvo un modelo ResNet34 con adaptación de dominio DANN entrenado, junto con los registros de pérdidas y métricas generados durante las épocas de entrenamiento. Estos resultados permitieron continuar con la evaluación y el ajuste del modelo en el siguiente sprint.

1.21.8.3 Desarrollo del Sprint 2: Verificación y perfeccionamiento del modelo

En el Sprint 2 se revisó el desempeño del modelo con 12 fotografías que no habían sido utilizadas durante el entrenamiento, el grupo estuvo formado por 6 imágenes de hojas saludables y 6 con salud comprometida, esta distribución permitió observar cómo respondía el modelo ante datos nuevos y mantener la misma cantidad de ejemplos para ambas categorías.

En una primera evaluación se observó una tendencia del modelo a clasificar la mayoría de las imágenes como saludables. Este comportamiento permitió identificar la necesidad de ajustar la configuración del entrenamiento y controlar la influencia de la adaptación de dominio sobre la tarea principal de clasificación.

A partir de los resultados iniciales se realizaron modificaciones en los parámetros del modelo. En la configuración final se estableció un peso de 0,15 para la pérdida de dominio y un factor de inversión de gradiente de 0,20. Posteriormente, se ejecutó nuevamente el entrenamiento y se evaluó la versión actualizada del modelo.

La evaluación final se realizó mediante las métricas de exactitud, precisión, recall, F1-score, confianza promedio y matriz de confusión. El modelo alcanzó una exactitud del 83,33

%, clasificando correctamente 10 de las 12 imágenes utilizadas en la prueba. La confianza promedio obtenida fue del 86,91 %.

Las métricas obtenidas durante la evaluación final se evidenciaron en la Ilustración 20.

```
=====
EVALUACIÓN FINAL EN EL DOMINIO OBJETIVO
=====
Found 12 files belonging to 2 classes.

1/2 ===== 1s 1s/step
2/2 ===== 0s 876ms/step
2/2 ===== 2s 884ms/step
Accuracy      : 83.33 %
Precision macro: 83.33 %
Recall macro  : 83.33 %
F1 macro      : 83.33 %
```

Ilustración 20 captura de las métricas

La matriz de confusión mostró que cinco imágenes saludables fueron clasificadas correctamente y una fue identificada como salud comprometida. De igual manera, cinco imágenes con salud comprometida fueron reconocidas correctamente y una fue clasificada como saludable.

La distribución de los aciertos y errores obtenidos con las imágenes del conjunto de prueba se presentó en la Ilustración 21.

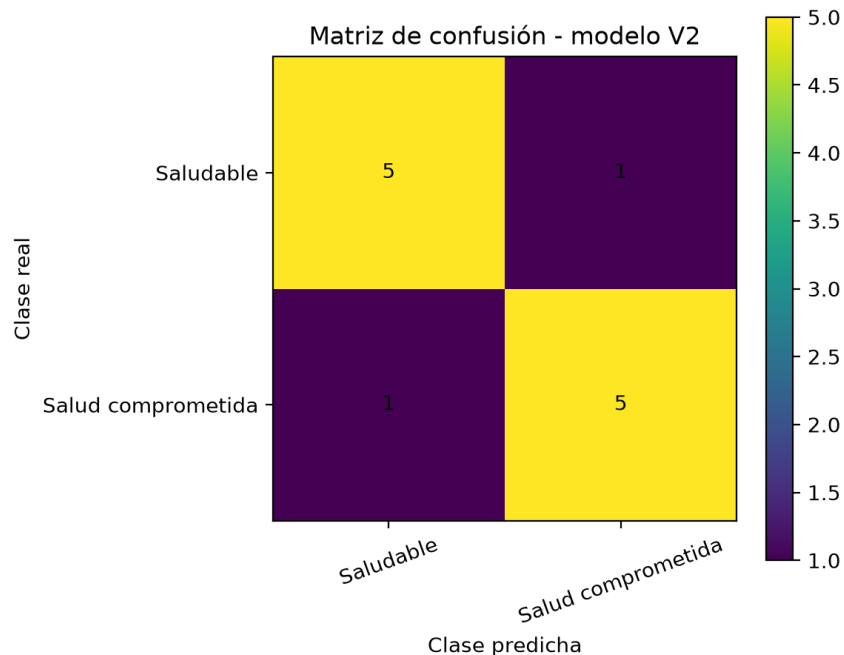
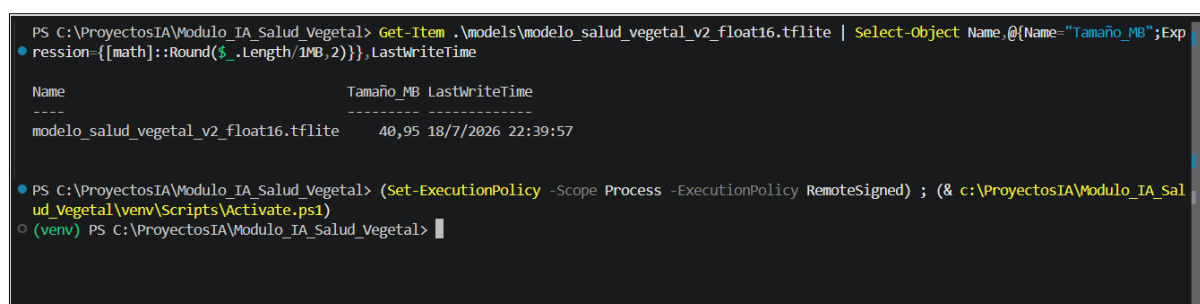


Ilustración 21 matriz de confusión

Debido al reducido número de fotografías utilizadas en el conjunto de prueba, estos resultados representan una evaluación técnica del modelo y no una validación definitiva de su desempeño en todas las condiciones agrícolas. También se reconoció la necesidad de ampliar el dominio objetivo y revisar las etiquetas mediante el acompañamiento de especialistas en trabajos posteriores.

Después de la evaluación, el modelo se guardó en formato Keras y luego se generó una versión en TensorFlow Lite con optimización float16, preparada para utilizar menos recursos durante su ejecución en el dispositivo móvil, para verificar que el cambio de formato no alterara de manera importante las predicciones, se analizaron las salidas de ambas versiones con las mismas imágenes, la mayor diferencia observada fue de 0,00112873, un valor pequeño que indicó un comportamiento similar entre el modelo Keras y el modelo TensorFlow Lite.

También se estableció un umbral de confianza del 65 % para su posterior implementación en la aplicación móvil. Cuando la probabilidad máxima no alcanza este valor, la aplicación presenta el resultado no determinado, evitando emitir una clasificación cuando la confianza es insuficiente.



```
PS C:\ProyectosIA\Modulo_IA_Salud_Vegetal> Get-Item .\models\modelo_salud_vegetal_v2_float16.tflite | Select-Object Name,@{Name="Tamaño_MB";Exp
ression={[math]::Round($_.Length/1MB,2)}},LastWriteTime

Name                                     Tamaño_MB LastWriteTime
-----
modelo_salud_vegetal_v2_float16.tflite  40,95 18/7/2026 22:39:57

PS C:\ProyectosIA\Modulo_IA_Salud_Vegetal> (Set-ExecutionPolicy -Scope Process -ExecutionPolicy RemoteSigned) ; (& c:\ProyectosIA\Modulo_IA_Sal
ud_Vegetal\venv\Scripts\Activate.ps1)
(venv) PS C:\ProyectosIA\Modulo_IA_Salud_Vegetal>
```

Ilustración 22 Resultados de la evaluación y conversión del modelo a TensorFlow Lite

Como resultado del Sprint 2 se obtuvo un modelo evaluado, almacenado y convertido al formato TensorFlow Lite float16, preparado para su integración en la aplicación Flutter durante el siguiente sprint.

1.21.8.4 Desarrollo del Sprint 3: Integración y validación del modelo

Durante el Sprint 3, el modelo se incorporó a una aplicación móvil independiente desarrollada con Flutter. Para realizar las predicciones se utilizó el archivo convertido a TensorFlow Lite con optimización float16, el cual se integró en el proyecto mediante el paquete `tflite_flutter`.

```

name: app_salud_vegetal
description: "Aplicación móvil para el análisis de salud vegetal del plátano."
publish_to: 'none'

version: 1.0.0+1

environment:
  sdk: ^3.10.1

dependencies:
  flutter:
    sdk: flutter

  cupertino_icons: ^1.0.8
  image_picker: ^1.2.3
  shared_preferences: ^2.5.5
  tf_lite_flutter: ^0.12.1
  image: ^4.8.0

dev_dependencies:
  flutter_test:
    sdk: flutter

  flutter_lints: ^6.0.0

flutter:
  uses-material-design: true

  assets:
    - assets/models/modelo_salud_vegetal.tflite
    - assets/models/labels.txt

```

Ilustración 23 Configuración del modelo TensorFlow Lite en el proyecto Flutter

Con esta configuración, el análisis se ejecutó completamente en el dispositivo móvil. La aplicación no necesitó conexión a Internet ni el envío de fotografías a un servidor externo, por lo que el módulo pudo funcionar de manera local e independiente.

La aplicación permitió capturar una fotografía mediante la cámara del dispositivo o seleccionar una imagen almacenada en la galería. Después de seleccionarla, la imagen se mostró en la pantalla para que pudiera revisarse antes de iniciar el análisis.

Antes de ejecutar la inferencia, la imagen fue redimensionada y preparada de acuerdo con el formato de entrada requerido por el modelo, correspondiente a $224 \times 224 \times 3$. Posteriormente, el modelo TensorFlow Lite generó las probabilidades asociadas con las categorías saludable y salud comprometida.

La aplicación presentó la categoría con mayor probabilidad, el porcentaje de confianza y las probabilidades obtenidas para ambas clases. Cuando la confianza máxima fue inferior al 65 %, el módulo mostró el resultado no determinado, evitando presentar una clasificación cuando la evidencia visual resultó insuficiente.

También se incorporaron recomendaciones generales relacionadas con el resultado. Estas recomendaciones tuvieron carácter informativo, no correspondieron a tratamientos fitosanitarios ni sustituyeron la evaluación de un profesional agrícola.

Además, se incorporó un historial para almacenar la información de los análisis realizados. Cada registro incluyó la fecha y hora del análisis, la imagen utilizada, el resultado,

las probabilidades y la recomendación presentada. Los datos se conservaron después de cerrar la aplicación y pudieron revisarse o eliminarse desde la pantalla del historial.



Ilustración 24 Presentación del resultado del análisis y del historial de la aplicación móvil

Para las pruebas se generó un APK de lanzamiento y se instaló en un teléfono Infinix X6858. En este dispositivo se verificaron la captura mediante la cámara, la selección de imágenes desde la galería, el funcionamiento del modelo sin conexión a Internet, la visualización de los resultados y el almacenamiento local del historial.



Ilustración 25 Ejecución de la aplicación móvil en modo sin conexión en el dispositivo de prueba

El módulo se desarrolló como una aplicación independiente para verificar su funcionamiento. Posteriormente, el código fuente y el modelo TensorFlow Lite se publicaron en GitHub y quedaron preparados para su incorporación en la aplicación principal por parte del integrante responsable de la integración del Proyecto Integrador.

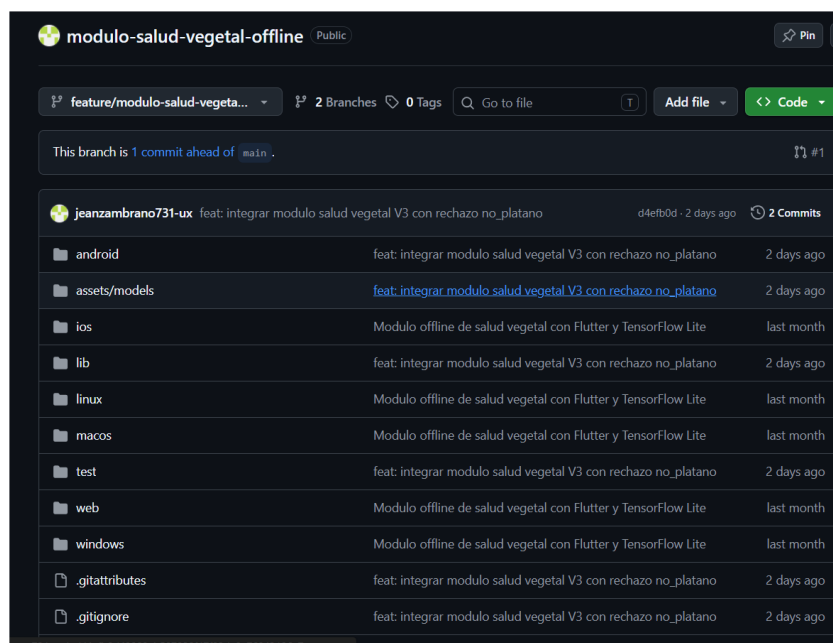


Ilustración 26 Publicación del código fuente y del modelo en GitHub

Con las evidencias presentadas se comprobó el cumplimiento de los entregables establecidos para el Sprint 3. Como resultado, se obtuvo una aplicación móvil funcional, con el modelo TensorFlow Lite integrado, capacidad de inferencia local, acceso a la cámara y la galería, presentación de resultados y almacenamiento del historial.

1.21.9 Resultado técnico del modelo entrenado

El proceso de desarrollo permitió obtener una versión final funcional del modelo basado en la arquitectura ResNetVD34 con adaptación de dominio DANN. El modelo fue implementado utilizando TensorFlow y Keras, y empleó pesos preentrenados de ImageNet para aprovechar características visuales aprendidas previamente.

Para el entrenamiento se utilizaron imágenes correspondientes a dos dominios. El dominio fuente estuvo conformado por imágenes públicas de hojas de plátano previamente revisadas, depuradas, organizadas y balanceadas. El dominio objetivo estuvo compuesto por fotografías capturadas en condiciones reales de campo.

Antes de ingresar al modelo, las imágenes fueron redimensionadas a 224×224 píxeles y normalizadas de acuerdo con el formato requerido por ResNetVD34. La salida principal del modelo generó las probabilidades correspondientes a las categorías saludable y salud comprometida.

Durante el proceso se desarrolló una primera versión del modelo que alcanzó una exactitud del 50 % y presentó una tendencia a clasificar todas las fotografías como saludables, este resultado inicial permitió identificar las necesidades para ajustar la influencia de la adaptación de dominio y así poder modificar los parámetros utilizados durante el entrenamiento.

Después se preparó una segunda versión de modelo con algunos cambios en la configuración, se asignó un peso de 0,15 a la pérdida de dominio y un factor de inversión de gradiente de 0,20, estos valores buscaron conservar la adaptación entre el dominio fuente y el dominio objetivo sin restar demasiada importancia a la clasificación de las hojas.

La evaluación final se realizó con un conjunto independiente compuesto por 12 fotografías reales, distribuidas de manera equilibrada entre las dos categorías: 6 imágenes saludables y 6 imágenes con salud comprometida.

Tabla 26 Métricas obtenidas en la evaluación final del modelo

Métrica	Resultado
Exactitud	83,33 %

Métrica	Resultado
Precisión ponderada	83,33 %
Recall ponderado	83,33 %
Medida F1 ponderada	83,33 %
Precisión macro	83,33 %
Medida F1 macro	83,33 %
Confianza promedio	86,91 %

Los resultados indicaron que el modelo clasificó correctamente 10 de las 12 fotografías utilizadas durante la evaluación. De las seis imágenes saludables, cinco fueron reconocidas correctamente y una fue clasificada como salud comprometida. De la misma manera, cinco de las seis imágenes con salud comprometida fueron identificadas correctamente, mientras que una fue clasificada como saludable.

La exactitud del 83,33 % representa una mejora frente a la primera versión del modelo. Sin embargo, este resultado debe interpretarse considerando que el conjunto de prueba estuvo formado únicamente por 12 fotografías reales. Por esta razón, la evaluación realizada permite comprobar técnicamente el funcionamiento del modelo, pero no representa una validación definitiva para todas las condiciones presentes en los cultivos de plátano.

También se reconoce como limitación que las etiquetas asignadas a las fotografías locales fueron establecidas de manera preliminar a partir de las características visuales observadas. En trabajos posteriores será necesario ampliar la cantidad de imágenes del dominio objetivo, organizar la separación de los datos por planta o por hoja y contar con la revisión de profesionales del área agrícola.

Una vez evaluado, el modelo se convirtió a TensorFlow Lite con optimización float16, el archivo final ocupó aproximadamente 40,95 MB y pudo incorporarse a la aplicación móvil para ejecutar el análisis directamente en el dispositivo.

Para comprobar que la conversión conservara las predicciones, se compararon las salidas del modelo Keras y del modelo TensorFlow Lite utilizando las mismas imágenes, la mayor diferencia registrada fue de 0,0011287, este valor reducido indicó que ambas versiones presentaban resultados similares.

La aplicación utiliza un umbral mínimo de confianza del 65 %, si la probabilidad más alta no alcanza ese porcentaje, muestra el resultado no determinado y recomienda tomar una nueva fotografía en mejores condiciones.

El desarrollo dejó establecido un proceso que comprendió la preparación de las imágenes, el entrenamiento de ResNetVD34 con DANN, la evaluación de sus resultados y la conversión del modelo a TensorFlow Lite, la versión final se incorporó a una aplicación Flutter capaz de realizar el análisis sin conexión a internet.

Desde la aplicación, el usuario puede tomar una fotografía o elegirla desde la galería, procesarla en el teléfono, consultar las probabilidades y guardar el resultado en un historial local, el alcance se limita a una valoración general de la hoja: no determina una enfermedad concreta, no mide su severidad y tampoco sustituye el criterio de un profesional agrícola.

La distribución de los aciertos y errores se presentó mediante la matriz de confusión de la Ilustración 21.

1.21.10 Integración del modelo TensorFlow Lite en Flutter

Una vez evaluado, el modelo fue convertido del formato Keras a TensorFlow Lite con optimización float16, esta versión se incorporó a una aplicación independiente desarrollada con Flutter, lo que permitió realizar las predicciones directamente en el dispositivo móvil, sin conexión a Internet ni comunicación con servidores externos, desde la aplicación el usuario puede tomar una fotografía mediante la cámara o seleccionar una fotografía desde la galería, antes de ingresar al modelo, la imagen se ajusta al tamaño $224 \times 224 \times 3$ y se procesa mediante el paquete `tflite_flutter`.

El sistema presenta las probabilidades de las categorías saludable y salud comprometida, junto con el porcentaje de confianza y una recomendación general. Cuando la confianza máxima es inferior al 65 %, se muestra el resultado no determinado.

También se implementó un historial local mediante `SharedPreferences`, en el que se almacenan la fecha, la hora, el resultado, las probabilidades y la recomendación de cada análisis.

La aplicación fue compilada en un APK de aproximadamente 106,3 MB e instalada en un teléfono Infinix X6858. Las pruebas permitieron comprobar el funcionamiento de la cámara, la galería, la inferencia offline, la presentación de resultados y el historial local.

Como resultado, se obtuvo un módulo móvil funcional e independiente, preparado para ser incorporado posteriormente en la aplicación principal del Proyecto Integrador.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\ProyectosIA\app_salud_vegetal_offline> Get-Item .\build\app\outputs\flutter-apk\app-release.apk
Mode LastWriteTime Length Name
----
-a---- 18/7/2026 18:36 111485799 app-release.apk

PS C:\ProyectosIA\app_salud_vegetal_offline> $adb = "$env:LOCALAPPDATA\Android\Sdk\platform-tools\adb.exe"
>> & $adb install -r .\build\app\outputs\flutter-apk\app-release.apk
Performing Streamed Install
Success
PS C:\ProyectosIA\app_salud_vegetal_offline> Copy-Item `
>> .\assets\models\modelo_salud_vegetal.tflite `
>> .\assets\models\modelo_salud_vegetal_v1_respaldo.tflite `
>> -Force
PS C:\ProyectosIA\app_salud_vegetal_offline> Copy-Item `
>> C:\ProyectosIA\Modulo_IA_Salud_Vegetal\models\modelo_salud_vegetal_v2_float16.tflite `
>> .\assets\models\modelo_salud_vegetal.tflite `
>> -Force
PS C:\ProyectosIA\app_salud_vegetal_offline> Copy-Item `
>> C:\ProyectosIA\Modulo_IA_Salud_Vegetal\models\labels_v2.txt `
>> .\assets\models\labels.txt `
>> -Force
PS C:\ProyectosIA\app_salud_vegetal_offline> Get-Childitem .\assets\models

```

Ilustración 27 Generación del APK de la aplicación

1.21.11 Pruebas funcionales del módulo móvil

Las pruebas funcionales se realizaron con la finalidad de comprobar el funcionamiento conjunto del modelo TensorFlow Lite y la aplicación desarrollada en Flutter. Se verificó que el usuario pudiera capturar o seleccionar una fotografía, ejecutar el análisis directamente en el dispositivo y visualizar el resultado sin conexión a Internet.

Durante las pruebas se comprobó el acceso a la cámara, la selección de imágenes desde la galería y la presentación previa de la fotografía. También se verificó el preprocesamiento de la imagen, la ejecución del modelo, la presentación de las probabilidades y la aplicación del umbral mínimo de confianza del 65 %.

El historial local permitió comprobar que los resultados permanecían almacenados después de cerrar y abrir nuevamente la aplicación. También se verificó la eliminación de los registros mediante una ventana de confirmación.

Tabla 27 Pruebas funcionales del módulo móvil

Código	Prueba realizada	Resultado
PF-01	Instalación y apertura de la aplicación	Correcto
PF-02	Selección de una fotografía desde la galería	Correcto
PF-03	Captura de una fotografía mediante la cámara	Correcto
PF-04	Presentación previa de la imagen seleccionada	Correcto
PF-05	Ejecución offline del modelo TensorFlow Lite	Correcto
PF-06	Presentación de las probabilidades de ambas categorías	Correcto
PF-07	Aplicación del umbral de confianza del 65 %	Correcto
PF-08	Presentación del resultado y la recomendación general	Correcto

Código	Prueba realizada	Resultado
PF-09	Almacenamiento y conservación del historial local	Correcto
PF-10	Eliminación de registros del historial	Correcto

Las pruebas realizadas en el teléfono Infinix X6858 permitieron comprobar que el módulo podía ejecutarse sin depender de una computadora, servidor externo o conexión a Internet. El flujo funcional quedó establecido de la siguiente manera:

Selección o captura de la imagen → preprocesamiento → inferencia con TensorFlow Lite → presentación del resultado → almacenamiento en el historial local.

Aunque las funcionalidades de la aplicación se ejecutaron correctamente, el rendimiento del modelo debe interpretarse considerando que la evaluación se realizó con un conjunto reducido de 12 fotografías reales. Por esta razón, los resultados tienen carácter orientativo y no sustituyen la revisión de un profesional agrícola.

Como resultado de esta etapa se obtuvo un módulo móvil funcional e independiente, preparado para su posterior incorporación dentro de la aplicación principal del Proyecto Integrador.

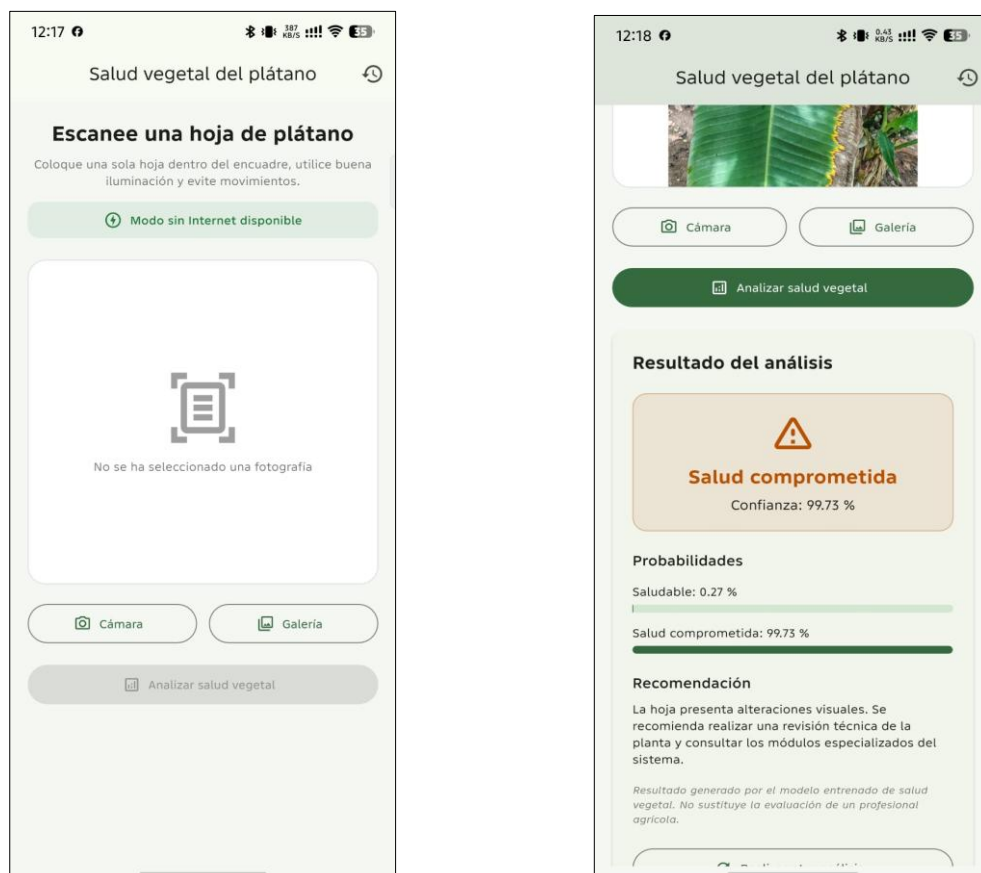


Ilustración 28 Funcionalidad de la app móvil

CAPÍTULO V

EVALUACIÓN DE RESULTADOS

1.22 Introducción

En este capítulo se presentan y analizan los resultados obtenidos durante el desarrollo del módulo móvil de inteligencia artificial orientado a la evaluación del estado general de salud vegetal en hojas de plátano. Mientras que en el capítulo anterior se describieron las actividades de preparación de los datos, entrenamiento, evaluación e integración del modelo, en esta sección se interpretan los resultados técnicos alcanzados.

La evaluación consideró dos componentes principales. El primero correspondió al rendimiento del modelo basado en la arquitectura ResNetVD34 con adaptación de dominio DANN. Para este análisis se utilizaron las métricas de exactitud, precisión, recall, medida F1, confianza promedio y matriz de confusión, obtenidas mediante un conjunto de prueba conformado por fotografías reales del dominio objetivo.

El segundo componente se centró en las pruebas funcionales del módulo móvil desarrollado en Flutter. En esta etapa se comprobó que el modelo TensorFlow Lite pudiera ejecutarse de forma local, además del correcto funcionamiento de la cámara, la galería, la visualización de resultados, la aplicación del umbral de confianza y el almacenamiento del historial de evaluaciones.

En la evaluación final, el modelo obtuvo una exactitud del 83,33 % al trabajar con 12 fotografías reales. Este resultado debe analizarse con cautela, debido al tamaño reducido del conjunto de prueba y al carácter provisional de las etiquetas utilizadas. El módulo puede funcionar sin conexión a Internet y permite realizar una valoración general del estado de las hojas; sin embargo, no identifica enfermedades específicas ni reemplaza el criterio de un profesional del área agrícola.

1.23 Presentación de los resultados

Los resultados obtenidos se organizaron en dos grupos. El primero correspondió al rendimiento del modelo de inteligencia artificial basado en ResNetVD34 con adaptación de dominio DANN. El segundo estuvo relacionado con las pruebas funcionales del módulo móvil desarrollado en Flutter.

Para analizar el modelo se consideraron los resultados obtenidos después del entrenamiento con imágenes del dominio fuente y del dominio objetivo. La evaluación se realizó con fotografías reales reservadas para prueba y permitió determinar la capacidad del modelo para clasificar las hojas en las categorías saludable y salud comprometida.

Para reconocer el desempeño del modelo se revisaron la exactitud, la precisión, recall, medida F1, confianza promedio y matriz de confusión, como parte de esta evaluación también se compararon las predicciones del modelo original en Keras con las obtenidas después de su conversión a TensorFlow Lite.

Las pruebas de la aplicación móvil incluyeron la toma de fotografías mediante la cámara, la selección de imágenes desde la galería y el análisis sin conexión a internet, además se verificó la presentación de las probabilidades, la ampliación del umbral de confianza del 65 % y el almacenamiento del historial local, finalmente, se generó el archivo APK, se instaló en un dispositivo Android y se comprobó su funcionamiento.

La presentación de los resultados consideró tanto los logros alcanzados como las condiciones en las que se realizaron las pruebas. Por esta razón, se tomó en cuenta la cantidad limitada de fotografías reales, el comportamiento de las probabilidades y las dificultades observadas en algunas clasificaciones.

1.23.1 Resultados del entrenamiento y evaluación del modelo

El proceso de entrenamiento permitió obtener un modelo funcional basado en la arquitectura ResNetVD34 con adaptación de dominio DANN. La red recibió imágenes con dimensiones de $224 \times 224 \times 3$ y estuvo conformada por un extractor de características, una salida para la clasificación del estado de salud vegetal y una salida para la identificación del dominio de procedencia de las imágenes.

Durante el entrenamiento se combinaron imágenes del dominio fuente con fotografías del dominio objetivo. El dominio fuente estuvo compuesto por imágenes públicas organizadas en las categorías saludable y salud comprometida, mientras que el dominio objetivo estuvo conformado por fotografías capturadas en condiciones reales de campo.

En una primera versión, el modelo obtuvo una exactitud del 50 % y presentó una tendencia a clasificar todas las imágenes como saludables. Este resultado permitió identificar la necesidad de ajustar la influencia de la adaptación de dominio sobre la clasificación principal.

Posteriormente, se desarrolló una segunda versión del modelo, en la cual se estableció un peso de 0,15 para la pérdida de dominio y un factor de inversión de gradiente de 0,20. La versión final fue evaluada mediante un conjunto independiente de 12 fotografías reales, distribuido de manera equilibrada entre 6 imágenes saludables y 6 imágenes con salud comprometida.

La exactitud obtenida fue del 83,33 %, lo que significa que el modelo clasificó correctamente 10 de las 12 fotografías utilizadas en la evaluación. Este resultado representó una mejora respecto a la primera versión del modelo.

En la categoría saludable, cinco de las seis imágenes fueron clasificadas correctamente y una fue identificada como salud comprometida. De igual manera, cinco de las seis imágenes con salud comprometida fueron reconocidas correctamente y una fue clasificada como saludable.

La confianza promedio alcanzó el 86,91 %, superando el umbral mínimo del 65 % establecido en la aplicación móvil. Cuando la probabilidad máxima es inferior a dicho umbral, la aplicación presenta el resultado no determinado y recomienda capturar una nueva fotografía en mejores condiciones.

Aunque los resultados muestran un comportamiento equilibrado entre ambas categorías, deben interpretarse considerando que el conjunto de prueba estuvo compuesto únicamente por 12 fotografías reales. Por esta razón, los valores obtenidos corresponden a una evaluación técnica del modelo y no representan una validación definitiva para todas las condiciones presentes en los cultivos de plátano

1.23.2 Interpretación objetiva

Los resultados obtenidos permitieron analizar por separado el rendimiento del modelo de inteligencia artificial y el funcionamiento técnico de la aplicación móvil. Esta distinción fue necesaria, debido a que la aplicación cumplió correctamente sus funciones, aunque el rendimiento predictivo del modelo todavía presentó limitaciones.

En la evaluación final, el modelo ResNetVD34 con adaptación de dominio DANN, la evaluación final alcanzó una exactitud del 83,33 %, clasificando correctamente 10 de las 12 fotografías reales utilizadas, además alcanzó una confianza promedio del 86,91 % y mostró un desempeño equilibrado en las categorías saludable y salud comprometida.

Estos resultados representan una mejora respecto a la primera versión del modelo, que alcanzó una exactitud del 50 % y presentó una tendencia a clasificar las imágenes como saludables. El ajuste de los parámetros de adaptación de dominio permitió obtener un comportamiento más equilibrado entre ambas categorías.

Sin embargo, el resultado del 83,33 % debe interpretarse con precaución, debido a que el conjunto de prueba estuvo formado únicamente por 12 fotografías reales. Asimismo, las etiquetas asignadas a las imágenes del dominio objetivo tienen carácter preliminar y requieren revisión posterior por parte de profesionales del área agrícola.

Las pruebas funcionales se realizaron desde la aplicación desarrollada con Flutter, las imágenes se obtuvieron mediante la cámara o la galería y fueron analizadas en el propio teléfono con TensorFlow Lite, durante este proceso no se utilizó conexión a internet ni se enviaron datos a servidores externos.

Después del análisis, la pantalla mostró las probabilidades de las categorías saludable y salud comprometida, junto con el nivel de confianza y una recomendación general, cuando la confianza máxima fue inferior al 65 %, se mostró el resultado no determinado, con la finalidad de evitar una clasificación concluyente cuando la probabilidad obtenida fuera insuficiente.

También se comprobó el funcionamiento del historial local mediante SharedPreferences, los resultados permanecieron almacenados después de cerrar la aplicación y pudieron ser consultados o eliminados desde la pantalla correspondiente.

Al finalizar el desarrollo, se generó un archivo APK de aproximadamente 106,3 MB e instalada en un teléfono Infinix X6858, durante las pruebas se comprobó el funcionamiento de la cámara, la selección de imágenes desde la galería, el análisis sin conexión a Internet, la presentación de los resultados y el registro de la información en el historial.

Con estas funciones, el flujo de la aplicación quedó organizado de la siguiente manera: **Captura o selección de la imagen → ajuste a $224 \times 224 \times 3$ → inferencia local con TensorFlow Lite → presentación de probabilidades y resultado → almacenamiento del análisis en el historial local.**

En términos generales, los resultados demostraron que fue posible integrar el modelo de inteligencia artificial en una aplicación móvil funcional e independiente. No obstante, el módulo evalúa únicamente el estado general visible de la hoja, no identifica enfermedades específicas, no determina niveles de severidad y no reemplaza la valoración de un profesional agrícola.

1.23.3 Resultados de las pruebas funcionales

Las pruebas funcionales tuvieron como finalidad comprobar que las principales opciones del módulo móvil respondieran de acuerdo con el comportamiento esperado. Se evaluaron las funciones relacionadas con la cámara, la galería, el análisis de imágenes, la visualización de resultados, el historial local y el manejo de errores.

Las pruebas se realizaron directamente en el teléfono Infinix X6858, después de instalar el archivo APK generado en modo de publicación. Debido a que el modelo se ejecuta mediante TensorFlow Lite, no fue necesario utilizar una computadora, servidor externo ni conexión a Internet.

Tabla 28 Resultados de las pruebas funcionales

Código	Prueba realizada	Resultado esperado	Resultado obtenido
PF-01	Abrir la aplicación	Mostrar la pantalla principal	Cumplido
PF-02	Seleccionar una fotografía desde la galería	Cargar y mostrar la imagen	Cumplido
PF-03	Capturar una fotografía mediante la cámara	Mostrar la fotografía capturada	Cumplido
PF-04	Analizar una fotografía	Ejecutar el modelo TensorFlow Lite en el dispositivo	Cumplido
PF-05	Mostrar el resultado	Presentar la categoría y el nivel de confianza	Cumplido
PF-06	Visualizar las probabilidades	Mostrar los porcentajes de ambas categorías	Cumplido
PF-07	Aplicar el umbral de confianza	Mostrar no determinada cuando la confianza sea menor al 65 %	Cumplido
PF-08	Mostrar una recomendación	Presentar una recomendación general	Cumplido
PF-09	Guardar un análisis	Registrar los datos en el historial local	Cumplido
PF-10	Cerrar y abrir la aplicación	Conservar los registros almacenados	Cumplido
PF-11	Eliminar el historial	Solicitar confirmación y borrar los registros	Cumplido
PF-12	Generar e instalar el APK	Ejecutar la aplicación en Android	Cumplido

Los resultados mostraron que las funciones principales del módulo se ejecutaron correctamente, fue posible tomar fotografías con la cámara o elegir las desde la galería, procesarlas en el propio dispositivo y visualizar el resultado sin que la aplicación se cerrara inesperadamente.

También se revisó el manejo de errores. Cuando no se seleccionaba una imagen o surgía un problema durante el análisis, el sistema mostraba un mensaje para informar al usuario, así

se evitó que aparecieran resultados incompletos o que la pantalla permaneciera cargando de manera indefinida.

Por último, se comprobó que los registros del historial continuaban disponibles después de cerrar y volver a abrir la aplicación, de esta manera, el usuario pudo consultar los análisis realizados anteriormente sin necesidad de ejecutar nuevamente el modelo.

A partir de las pruebas realizadas se determinó que el módulo cumple con los requerimientos funcionales establecidos para su funcionamiento independiente. Además, la inferencia se ejecuta sin conexión a Internet y el módulo quedó preparado para su posterior integración dentro de la aplicación principal del Proyecto Integrador.

1.24 Análisis e interpretación de los resultados

Los resultados obtenidos se analizaron desde dos perspectivas: el rendimiento del modelo de inteligencia artificial y el funcionamiento técnico del módulo móvil. Esta diferenciación fue necesaria porque el correcto funcionamiento de la aplicación no garantiza por sí solo que el modelo alcance un rendimiento adecuado en todas las condiciones de campo.

En el aspecto técnico, se completó el desarrollo del modelo ResNetVD34 con adaptación de dominio mediante DANN, después del entrenamiento y la evaluación, el modelo se convirtió a TensorFlow Lite y se incorporó a una aplicación independiente creada con Flutter, esta implementación permitió analizar las fotografías directamente en el teléfono, sin requerir conexión a Internet ni enviar información a servidores externos.

Durante las pruebas funcionales se aprobó el uso de la cámara, la selección de imágenes desde la galería, el procesamiento de las fotografías, la presentación de probabilidades y el almacenamiento del historial local, así mismo se generó el archivo APK, se instaló en un dispositivo Android y se verificó el funcionamiento de la aplicación..

Desde la perspectiva del rendimiento, el modelo final alcanzó una exactitud del 83,33 %, clasificando correctamente 10 de las 12 fotografías reales utilizadas durante la evaluación. La precisión, el recall y la medida F1, tanto macro como ponderados, también alcanzaron el 83,33 %, mientras que la confianza promedio fue del 86,91 %.

La matriz de confusión permitió observar resultados similares en dos categorías, de las seis imágenes saludables, cinco fueron clasificadas correctamente y una fue identificada como salud comprometida, en el caso contrario ocurrió lo mismo: cinco de las seis imágenes con salud comprometida fueron identificadas correctamente y una se registró como saludable.

Frente a la primera versión del modelo, que obtuvo una exactitud del 50 % y clasificaba todas las imágenes como saludables, el resultado final representó una mejora, aun así, debe interpretarse con cautela, debido a que la evaluación se realizó con un conjunto de prueba

pequeño, la cantidad de fotografías locales fue una de las principales limitaciones del estudio, aunque el dominio fuente contenía un mayor número de imágenes, para el dominio objetivo solo se dispuso de 64 fotografías reales destinadas a la clasificación binaria, de ese conjunto, 12 imágenes se reservaron para la evaluación final.

También existieron diferencias visuales entre ambos dominios, las imágenes públicas y las fotografías tomadas en el campo presentaban variaciones de iluminación, fondo, distancia, orientación, enfoque y presencia de elementos externos.

Otra limitación estuvo relacionada con el etiquetado de las fotografías locales, las categorías se asignaron inicialmente a partir de características visibles y deberán ser revisadas por un profesional del área agrícola, una etiqueta ambigua o incorrecta puede afectar tanto el entrenamiento como la evaluación.

La adaptación de dominio DANN se utilizó para reducir las diferencias entre las imágenes públicas y las fotografías reales, sin embargo al no contar con una comparación equivalente frente a un modelo entrenado sin esta adaptación, no es posible determinar cuantitativamente qué parte de la mejora corresponde únicamente a DANN.

Por esta razón, una evaluación futura deberá incluir una línea base sin adaptación de dominio. Esta comparación permitirá determinar con mayor precisión el efecto de DANN sobre las fotografías correspondientes al entorno objetivo.

El umbral del 65 % constituyó una medida preventiva dentro de la aplicación. Cuando la probabilidad máxima no alcanza este valor, el módulo presenta el resultado no determinada, evitando emitir una clasificación cuando la confianza es insuficiente.

En términos generales, los resultados demostraron la viabilidad técnica de integrar un modelo ResNetVD34 con DANN en una aplicación móvil offline. No obstante, el módulo evalúa únicamente el estado general visible de las hojas, no identifica enfermedades específicas, no determina niveles de severidad y no reemplaza la valoración de un profesional agrícola.

1.25 Cumplimiento de los objetivos de investigación

El desarrollo realizado permitió cumplir el objetivo general de implementar un módulo móvil de inteligencia artificial basado en ResNetVD34 y adaptación de dominio DANN para evaluar el estado general de salud vegetal en hojas de plátano capturadas en condiciones reales de campo.

En relación con el primer objetivo específico, se analizaron las necesidades y los requerimientos funcionales del módulo. A partir de este proceso se definieron las funciones

relacionadas con la captura y selección de imágenes, ejecución del análisis, presentación de probabilidades, aplicación del umbral de confianza y almacenamiento del historial local.

El segundo objetivo se cumplió mediante la revisión de los fundamentos relacionados con las redes neuronales convolucionales, ResNetVD34, transferencia de aprendizaje, adaptación de dominio DANN, capa de inversión de gradiente y TensorFlow Lite. Estos conceptos sirvieron como base para el diseño y la implementación del modelo.

Para el cumplimiento del tercer objetivo se prepararon los conjuntos de imágenes del dominio fuente y del dominio objetivo. Las fotografías fueron revisadas, depuradas, organizadas, balanceadas y separadas en conjuntos de entrenamiento, validación y prueba. Durante este proceso se eliminaron 137 imágenes duplicadas y se utilizaron 64 fotografías reales para la clasificación binaria.

El cuarto objetivo se cumplió mediante el desarrollo y entrenamiento del modelo ResNetVD34 con DANN. La evaluación final se realizó con 12 fotografías reales y permitió obtener una exactitud del 83,33 %, junto con valores equivalentes de precisión, recall y medida F1. También se generó la matriz de confusión y se obtuvo una confianza promedio del 86,91 %.

El quinto objetivo se cumplió mediante la conversión del modelo al formato TensorFlow Lite con optimización float16 y su integración en una aplicación independiente desarrollada con Flutter, este módulo permite utilizar la cámara o la galería, presentar las probabilidades de clasificación y almacenar los resultados en un historial local.

Durante las pruebas realizadas en el teléfono Infinix X6858, aplicación pudo analizar las imágenes directamente en el dispositivo, para completar este proceso no fue necesaria una conexión a Internet, el uso de servicios web ni la comunicación con servidores externos. También se verificó que el archivo APK se generara y se instalara correctamente.

En consecuencia, los objetivos planteados fueron cumplidos dentro del alcance definido para el módulo. No obstante, los resultados deben interpretarse considerando la cantidad limitada de fotografías reales y el carácter preliminar de sus etiquetas. Por esta razón, el módulo no identifica enfermedades específicas, no determina niveles de severidad y no reemplaza la evaluación de un profesional agrícola.

1.26 Limitaciones identificadas

Durante la evaluación se identificaron varias limitaciones que deben considerarse al interpretar los resultados obtenidos.

La primera limitación correspondió a la cantidad reducida de fotografías reales disponibles. El dominio objetivo estuvo conformado por 64 imágenes utilizables, cantidad

limitada para representar la diversidad de condiciones de iluminación, fondos, orientaciones, distancias y estados visuales que pueden presentarse en una plantación de plátano.

La segunda limitación estuvo relacionada con las etiquetas asignadas a las fotografías reales. Las categorías saludables y salud comprometida fueron establecidas mediante una revisión visual preliminar. Por esta razón, se requiere la participación de profesionales del área agrícola para comprobar que las imágenes se encuentren correctamente clasificadas.

También se identificaron diferencias entre las imágenes públicas del dominio fuente y las fotografías reales del dominio objetivo. El conjunto público presentaba condiciones distintas a las observadas en la Granja Experimental ULEAM, Extensión El Carmen, principalmente en aspectos relacionados con el fondo, la iluminación, la distancia y la calidad de captura.

Otra limitación correspondió a la división de las fotografías locales. Debido a que varias imágenes fueron capturadas de manera consecutiva, existe la posibilidad de que fotografías pertenecientes a una misma hoja o planta hayan sido distribuidas entre los conjuntos de entrenamiento, validación y prueba. En futuras evaluaciones será necesario realizar la separación mediante identificadores únicos de planta y hoja.

La evaluación final se realizó con un conjunto compuesto únicamente por 12 fotografías reales, distribuidas en 6 imágenes saludables y 6 imágenes con salud comprometida. Debido al tamaño reducido del conjunto, una sola clasificación incorrecta produce una variación considerable en las métricas y limita la posibilidad de generalizar el resultado del 83,33 % a todas las condiciones de campo.

Asimismo, no se realizó una comparación controlada entre un modelo ResNetVD34 sin adaptación de dominio y otro modelo ResNetVD34 con DANN. En consecuencia, no fue posible medir de forma independiente el efecto producido por la adaptación de dominio en el rendimiento final.

En esta etapa, el modelo únicamente permite distinguir si la apariencia de una hoja corresponde a la categoría saludable o a salud comprometida, por lo tanto el resultado no representa el diagnóstico de una enfermedad específica, tampoco indica su nivel de severidad ni recomienda tratamientos fitosanitarios, en cuanto a la aplicación móvil fue posible ejecutar el análisis sin conexión a internet mediante TensorFlow Lite; sin embargo, las pruebas se realizaron principalmente en un teléfono Infinix X6858. Por esta razón, todavía será necesario comprobar su funcionamiento en otros equipos con distintas versiones de Android y características de hardware.

Finalmente, los resultados presentados por el módulo tienen carácter orientativo y no sustituyen la evaluación de un profesional agrícola. Su utilización debe considerarse como una herramienta tecnológica de apoyo para la observación inicial del estado general de las hojas de plátano.

1.27 Síntesis del capítulo

En este capítulo se evaluaron los resultados obtenidos durante el entrenamiento del modelo de inteligencia artificial y la implementación del módulo móvil desarrollado en Flutter.

La evaluación final del modelo ResNetVD34 con adaptación de dominio DANN alcanzó una exactitud del 83,33 %, junto con valores de precisión, recall y medida F1 del 83,33 %. Asimismo, se obtuvo una confianza promedio del 86,91 %. El modelo clasificó correctamente 10 de las 12 fotografías reales utilizadas en el conjunto de prueba.

Desde el punto de vista funcional, se comprobó que la aplicación permite capturar fotografías mediante la cámara, seleccionar imágenes desde la galería, ejecutar el modelo TensorFlow Lite directamente en el dispositivo y presentar las probabilidades de las categorías saludable y salud comprometida, también se verificó la aplicación del umbral de confianza del 65 %, la presentación de recomendaciones generales y el almacenamiento del historial local.

Durante las pruebas, el módulo realizó el análisis directamente en el teléfono, sin conexión a Internet ni comunicación con servidores externos, también se generó el archivo APK y se comprobó su instalación en un dispositivo Android, los resultados deberán analizarse con cautela, ya que la evaluación utilizó pocas fotografías tomadas en condiciones reales y sus etiquetas todavía son preliminares.

En conclusión, se obtuvo un módulo móvil funcional para la evaluación del estado general visible de las hojas de plátano. Sin embargo, el sistema no identifica enfermedades específicas, no determina niveles de severidad y no reemplaza la evaluación de un profesional agrícola.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

1.28 Conclusiones

El análisis de las necesidades tecnológicas permitió identificar que el módulo debía funcionar sin conexión a Internet, presentar una interfaz comprensible y permitir el uso de la cámara y la galería. Estos requerimientos fueron considerados durante el desarrollo de la aplicación móvil.

La revisión teórica permitió fundamentar el uso de las redes neuronales convolucionales, la arquitectura ResNetVD34, el aprendizaje por transferencia, la adaptación de dominio DANN y TensorFlow Lite. También permitió delimitar el sistema a la clasificación general de las hojas como saludable o salud comprometida.

Durante la preparación del conjunto de datos se localizaron y eliminaron 137 imágenes duplicadas, después se equilibraron las categorías del conjunto público y se seleccionaron 64 fotografías reales, que fueron distribuidas entre las etapas de entrenamiento, validación y prueba

En la evaluación realizada con 12 fotografías reales, el modelo ResNetVD34 con adaptación de dominio mediante DANN obtuvo un 83,33 % en exactitud, precisión, recall y F1-score, aunque este resultado fue favorable, la cantidad limitada de imágenes de prueba no permite asegurar que el modelo mantenga el mismo desempeño en otras condiciones de campo.

La conversión a TensorFlow Lite permitió integrar el modelo en una aplicación desarrollada con Flutter. Las pruebas confirmaron el funcionamiento de la cámara, la galería, la presentación de probabilidades, el umbral de confianza del 65 % y el almacenamiento local del historial.

Se cumplió el objetivo general mediante el desarrollo de un módulo móvil funcional cumplió el objetivo general para evaluar el estado general de salud vegetal de las hojas de plátano, la aplicación funciona sin conexión a Internet y presenta tres posibles resultados: saludable, salud comprometida o no determinada. Su alcance no incluye la identificación de enfermedades específicas y tampoco reemplaza la revisión de un profesional del área agrícola

1.29 Recomendaciones

Para continuar con esta investigación será necesario reunir más fotografías tomadas en condiciones reales de campo, procurando incluir cambios de iluminación, fondo, orientación y distancia, también es importante que un profesional del área agrícola revise y valide las categorías asignadas a cada imagen.

Con un conjunto de prueba más amplio, convendría comparar el modelo actual con una versión de ResNetVD34 entrenada sin adaptación de dominio y bajo las mismas condiciones, esta comparación permitiría conocer con mayor precisión cuánto aporta DANN al desempeño del modelo.

En la parte móvil, se deberán realizar pruebas de usabilidad con estudiantes y personal de la Granja Experimental, asimismo, será necesario comprobar el funcionamiento de la aplicación en dispositivos Android con diferentes versiones del sistema operativo y capacidades de hardware.

Finalmente, el módulo deberá integrarse en la aplicación principal del Proyecto Integrador, manteniendo sus funciones principales: trabajo sin conexión a Internet, historial local y resultado no determinado cuando la confianza sea inferior al 65 %. El sistema debe conservarse como una herramienta de apoyo para la evaluación general de las hojas, sin presentarse como un diagnóstico profesional.

REFERENCIAS

- Arias Gonzáles, J. L., & Covinos Gallardo, M. (2021). *Diseño y metodología de la investigación*. Enfoques Consulting EIRL.
https://www.researchgate.net/publication/352157132_DISENO_Y_METODOLOGIA_DE_LA_INVESTIGACION
- Arman, S. E., Bhuiyan, M. A. B., Abdullah, H. M., Islam, S., Chowdhury, T. T., & Hossain, M. A. (2023). BananaLSD: A banana leaf images dataset for classification of banana leaf diseases using machine learning. *Data in Brief*, 50, 109608.
<https://doi.org/10.1016/j.dib.2023.109608>
- Bhuiyan, M. A. B., Abdullah, H. M., Arman, S. E., Rahman, S. S., & Al Mahmud, K. (2023). BananaSqueezeNet: A very fast, lightweight convolutional neural network for the diagnosis of three prominent banana leaf diseases. *Smart Agricultural Technology*, 4, 100214. <https://doi.org/10.1016/j.atech.2023.100214>
- Chen, J., Chen, J., Zhang, D., Sun, Y., & Nanekaran, Y. A. (2020). Using deep transfer learning for image-based plant disease identification. *Computers and Electronics in Agriculture*, 173, 105393. <https://doi.org/10.1016/j.compag.2020.105393>
- Chollet, F. (2020). *Transfer learning & fine-tuning*. Keras.
https://keras.io/guides/transfer_learning/
- Feria Avila, H., Matilla González, M., & Mantecón Licea, S. (2020). La entrevista y la encuesta: ¿Métodos o técnicas de indagación empírica? *Didasc@lia: Didáctica y Educación*, 11(3), 62–79. <https://dialnet.unirioja.es/servlet/articulo?codigo=7692391>
- HassanPour Zonoozi, M., & Seydi, V. (2023). A survey on adversarial domain adaptation. *Neural Processing Letters*, 55, 2429–2469. <https://doi.org/10.1007/s11063-022-10977-5>
- Hendrickx, K., Perini, L., Van der Plas, D., Meert, W., & Davis, J. (2024). Machine learning with a reject option: A survey. *Machine Learning*, 113(5), 3073–3110.
<https://doi.org/10.1007/s10994-024-06534-x>
- Hernández-Sampieri, R., & Mendoza Torres, C. P. (2023). *Metodología de la investigación: Las rutas cuantitativa, cualitativa y mixta*. McGraw-Hill Interamericana.
<https://opacbiblioteca.unibe.edu.do/bib/114084>
- Jiménez, N., Orellana, S., Mazon-Olivo, B., Rivas-Asanza, W., & Ramírez-Morales, I. (2025). Detection of leaf diseases in banana crops using deep learning techniques. *AI*, 6(3), 61. <https://doi.org/10.3390/ai6030061>

- Krishnan, V., Deepa, J., Rao, P., Divya, V., & Kaviarasan, S. (2022). An automated segmentation and classification model for banana leaf disease detection. *Journal of Applied Biology & Biotechnology*, 10(1), 213–220.
<https://doi.org/10.7324/JABB.2021.100126>
- Liu, X., Yoo, C., Xing, F., Oh, H., El Fakhri, G., Kang, J. W., & Woo, J. (2022). Deep unsupervised domain adaptation: A review of recent advances and perspectives. *APSIPA Transactions on Signal and Information Processing*, 11, e25.
<https://doi.org/10.1561/116.00000192>
- Muneeb, M., Feng, S. F., & Henschel, A. (2022). Deep learning pipeline for image classification on mobile phones. *Computer Science & Information Technology*, 1–20.
<https://doi.org/10.5121/csit.2022.120901>
- Raschka, S., & Mirjalili, V. (2024). *Python machine learning: Aprendizaje automático y aprendizaje profundo con Python, scikit-learn y TensorFlow*. Ediciones Anaya Multimedia. <https://elibro.net/es/lc/u/leam/titulos/280024>
- Raschka, S., Liu, Y., & Mirjalili, V. (2023). *Machine learning con PyTorch y Scikit-Learn: Desarrollo de modelos machine learning y deep learning con Python*. Marcombo. <https://www.marcombo.com/libro/libros-tecnicos-de-arte-y-cientificos/informatica-libros-tecnicos-y-cientificos/analitica-de-datos-e-internet-informatica/machine-learning-con-pytorch-y-scikit-learn/>
- Sanga, S., Machuve, D., & Jomanga, K. (2020). Mobile-based deep learning models for banana disease detection. *Engineering, Technology & Applied Science Research*, 10(3), 5674–5677. <https://doi.org/10.48084/etasr.3452>
- Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The definitive guide to Scrum: The rules of the game*. Scrum Guides.
<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
- Selvaraj, M. G., Vergara, A., Montenegro, F., Ruiz, H., Safari, N., Raymaekers, D., Ocimati, W., Ntamwira, J., Tits, L., Omondi, A. B., & Blomme, G. (2020). Detection of banana plants and their major diseases through aerial images and machine learning methods: A case study in DR Congo and Republic of Benin. *ISPRS Journal of Photogrammetry and Remote Sensing*, 169, 110–124. <https://doi.org/10.1016/j.isprsjprs.2020.08.025>
- Soria Olivas, E., Rodríguez Belenguer, P., García Vidal, Q., Vaquer Estarich, F., Vicent Camisón, J., & Vila Tomás, J. (2022). *Inteligencia artificial: Casos prácticos con aprendizaje profundo*. Ediciones de la U.

- Sunitha, P., Uma, B., Channakeshava, S., & Suresh Babu, C. S. (2023). A fully labelled image dataset of banana leaves deficient in nutrients. *Data in Brief*, 48, 109155.
<https://doi.org/10.1016/j.dib.2023.109155>
- Tugrul, B., Elfatimi, E., & Eryigit, R. (2022). Convolutional neural networks in detection of plant leaf diseases: A review. *Agriculture*, 12(8), 1192.
<https://doi.org/10.3390/agriculture12081192>
- Vizcaíno Zúñiga, P. I., Cedeño Cedeño, R. J., & Maldonado Palacios, I. A. (2023). Metodología de la investigación científica: Guía práctica. *Ciencia Latina Revista Científica Multidisciplinar*, 7(4), 9723–9762.
https://doi.org/10.37811/cl_rcm.v7i4.7658

1.30 Anexo A: Evidencias técnicas del desarrollo del módulo

	salud_comprometida_leve	C:\Usuarios\LENOVO\OneDrive - ULEAM\Documents
	salud_comprometida_alta	C:\Usuarios\LENOVO\OneDrive - ULEAM\Documents
	saludable	C:\Usuarios\LENOVO\OneDrive - ULEAM\Documents

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS python + - [ ] [X] ... | [ ] [X]
```

```
(venv) PS C:\ProyectosIA\Modulo_IA_Salud_Vegetal> python -m src_train_dann --epocas 5 --pasos 50 --pasos-validacion 5
```

Time	Step	Loss	Accuracy	Validation Loss
2/50	1:58	2s/step - loss: 3.2169	- valida_clases_accuracy: 0.6875	- valida_clases_loss: 0.3476
3/50	1:52	2s/step - loss: 3.2593	- valida_clases_accuracy: 0.5833	- valida_clases_loss: 0.4330
4/50	2:04	3s/step - loss: 3.2454	- valida_clases_accuracy: 0.6250	- valida_clases_loss: 0.3808
5/50	2:04	3s/step - loss: 3.1748	- valida_clases_accuracy: 0.5750	- valida_clases_loss: 0.3779
6/50	1:56	3s/step - loss: 3.1191	- valida_clases_accuracy: 0.6042	- valida_clases_loss: 0.3924
7/50	1:53	3s/step - loss: 3.0272	- valida_clases_accuracy: 0.6250	- valida_clases_loss: 0.3700
8/50	1:49	3s/step - loss: 2.9593	- valida_clases_accuracy: 0.6094	- valida_clases_loss: 0.3873
9/50	1:45	3s/step - loss: 3.0041	- valida_clases_accuracy: 0.6250	- valida_clases_loss: 0.3861
10/50	1:45	3s/step - loss: 3.0526	- valida_clases_accuracy: 0.6000	- valida_clases_loss: 0.3860
11/50	1:42	3s/step - loss: 3.0615	- valida_clases_accuracy: 0.6136	- valida_clases_loss: 0.3809
12/50	1:36	3s/step - loss: 3.0438	- valida_clases_accuracy: 0.6250	- valida_clases_loss: 0.3868
13/50	1:33	3s/step - loss: 3.1131	- valida_clases_accuracy: 0.6250	- valida_clases_loss: 0.3846
14/50	1:30	3s/step - loss: 3.1289	- valida_clases_accuracy: 0.6339	- valida_clases_loss: 0.3748
15/50	1:27	3s/step - loss: 3.1758	- valida_clases_accuracy: 0.6417	- valida_clases_loss: 0.3693
16/50	1:26	3s/step - loss: 3.2118	- valida_clases_accuracy: 0.6484	- valida_clases_loss: 0.3605
17/50	1:23	3s/step - loss: 3.2866	- valida_clases_accuracy: 0.6324	- valida_clases_loss: 0.3639
18/50	1:19	2s/step - loss: 3.3727	- valida_clases_accuracy: 0.6319	- valida_clases_loss: 0.3648
19/50	1:16	2s/step - loss: 3.5199	- valida_clases_accuracy: 0.6316	- valida_clases_loss: 0.3671
20/50	1:14	2s/step - loss: 3.6362	- valida_clases_accuracy: 0.6313	- valida_clases_loss: 0.3635
21/50	1:11	2s/step - loss: 3.7299	- valida_clases_accuracy: 0.6488	- valida_clases_loss: 0.3538
22/50	1:09	2s/step - loss: 3.8381	- valida_clases_accuracy: 0.6534	- valida_clases_loss: 0.3521
23/50	1:07	2s/step - loss: 4.0015	- valida_clases_accuracy: 0.6630	- valida_clases_loss: 0.3486
24/50	1:03	2s/step - loss: 4.0843	- valida_clases_accuracy: 0.6667	- valida_clases_loss: 0.3479
25/50	1:01	2s/step - loss: 4.2480	- valida_clases_accuracy: 0.6650	- valida_clases_loss: 0.3451
26/50	58s	2s/step - loss: 4.3930	- valida_clases_accuracy: 0.6587	- valida_clases_loss: 0.3445
27/50	56s	2s/step - loss: 4.5598	- valida_clases_accuracy: 0.6574	- valida_clases_loss: 0.3425
28/50	54s	2s/step - loss: 4.7299	- valida_clases_accuracy: 0.6652	- valida_clases_loss: 0.3405

```
valida dominio accuracy dominio: 0.1806 - valida dominio loss: 3.5889
```

Ilustración 31 Pantalla principal de la aplicación Flutter

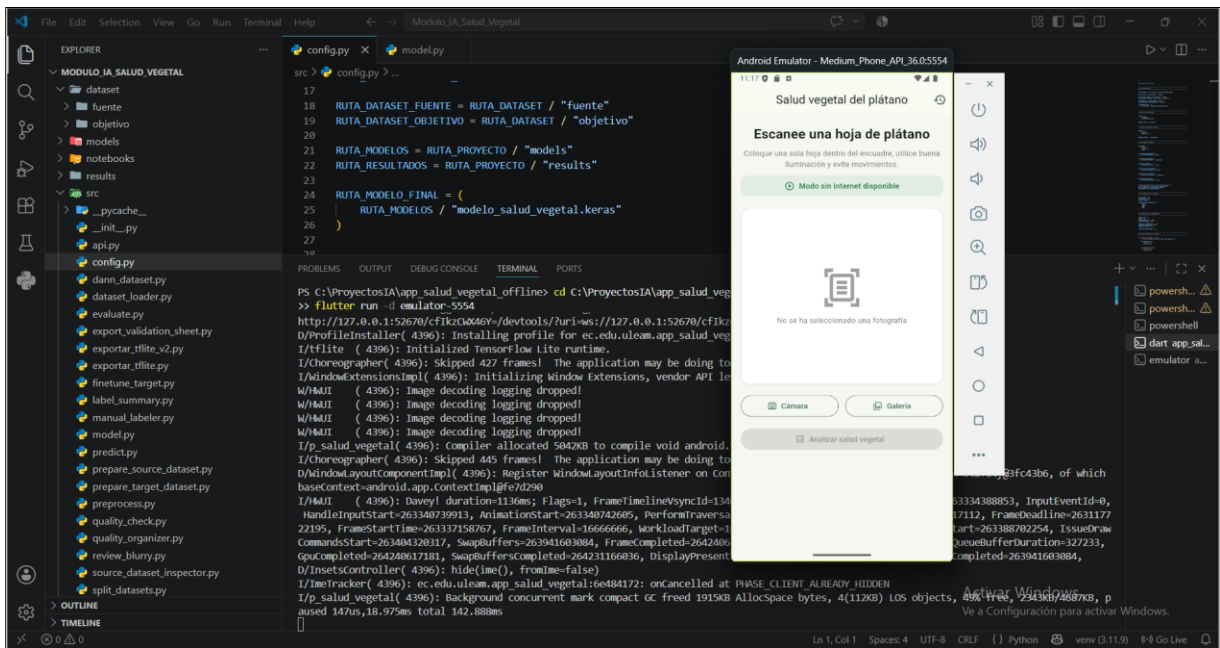


Ilustración 32 Resultado de una hoja saludable aplicación Flutter

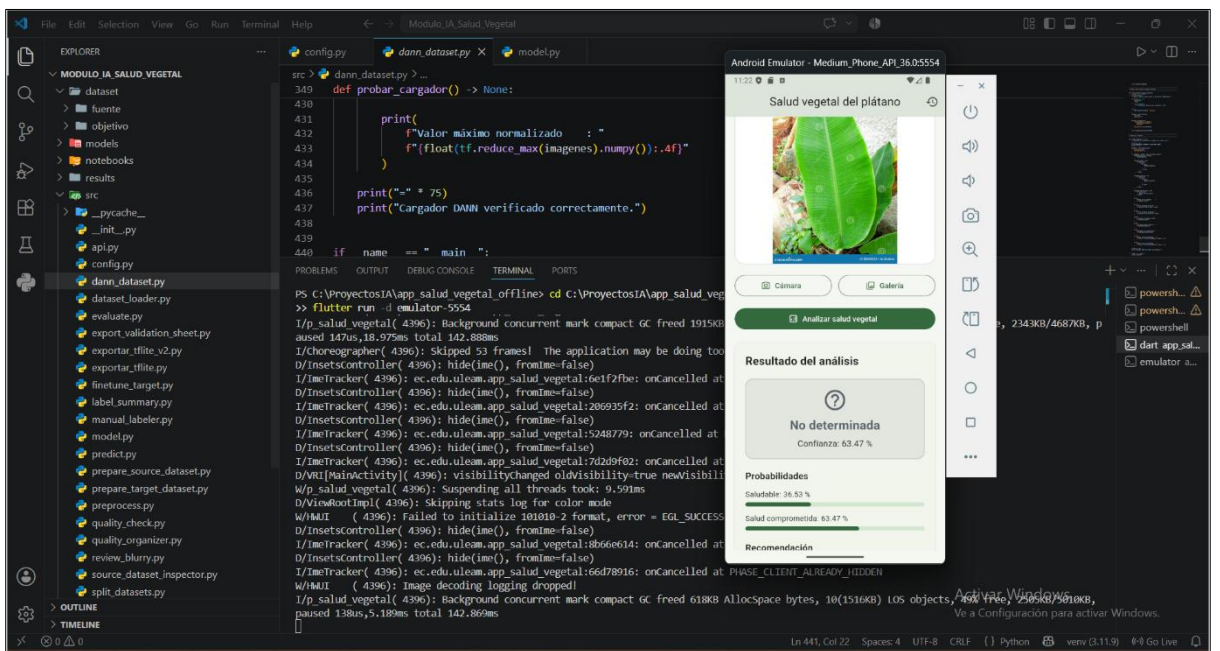


Ilustración 33 Resultado de una hoja con salud comprometida aplicación Flutter

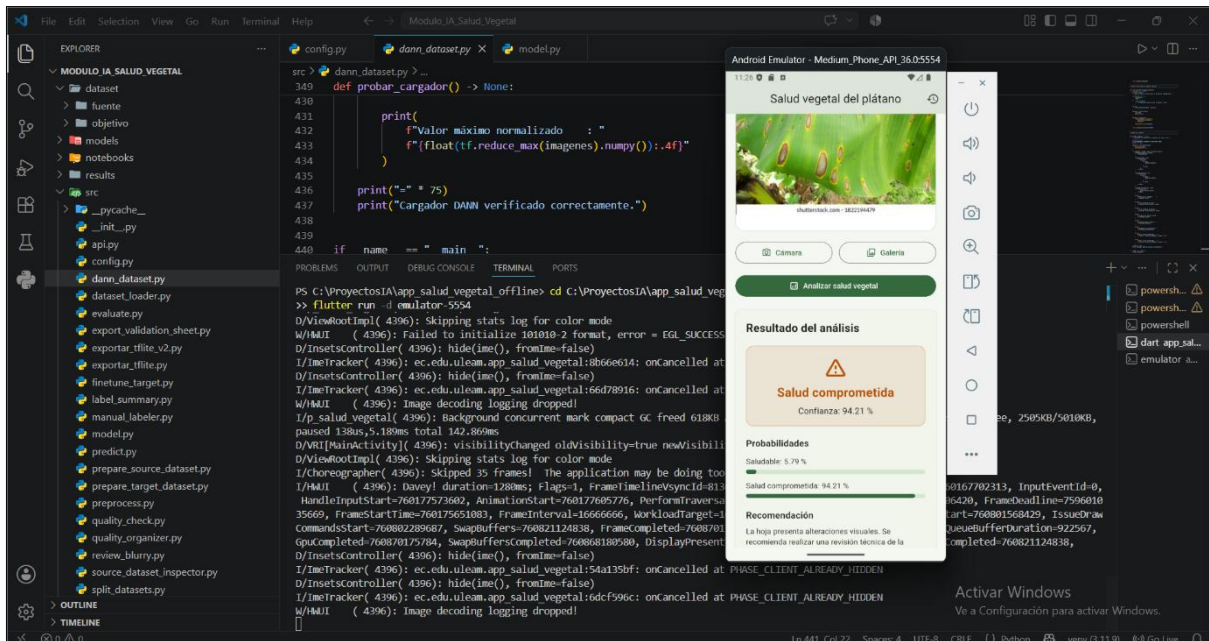
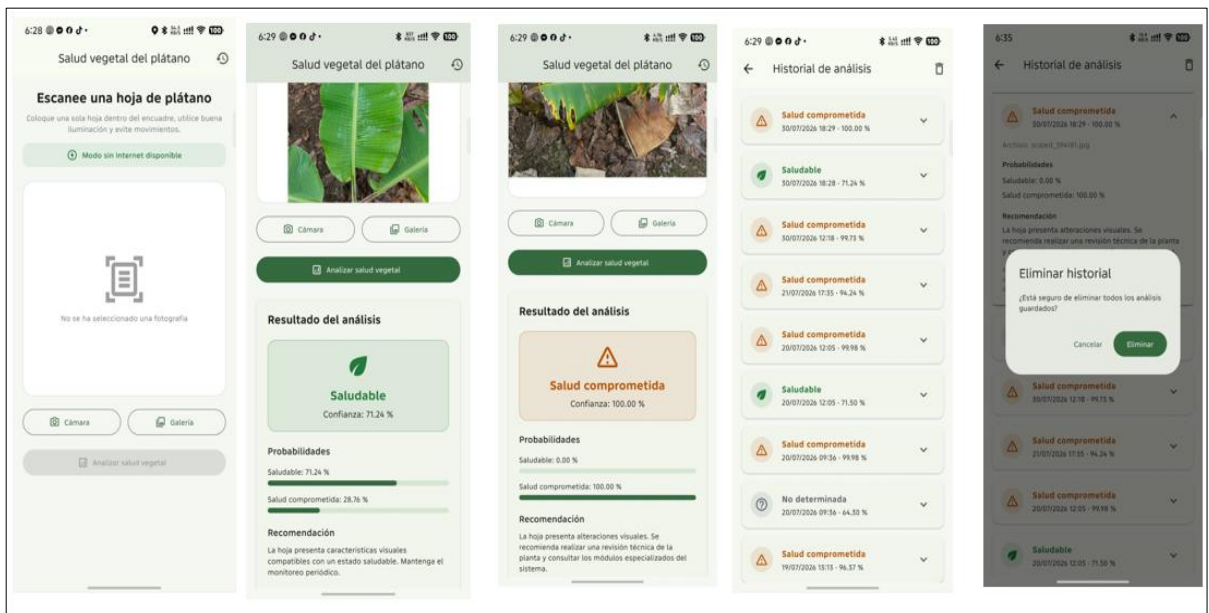


Ilustración 34 Resultado de la aplicación funcional en un dispositivo Android.



1.31 Anexo B: Evidencias de la entrevista y la encuesta

Ilustración 35 Evidencia de la entrevista semiestructurada aplicada al docente especialista de la Granja Experimental ULEAM El Carmen.



Ilustración 36 Evidencia de la encuesta aplicada a los estudiantes de agropecuaria de la Granja Experimental ULEAM El Carmen.



Ilustración 37 Estructura de la encuesta realizada a los estudiantes de agropecuaria de la Granja Experimental ULEAM El Carmen.

ENCUESTA DE REALIDAD AUMENTADA PARA LA CAPACITACIÓN AGRÍCOLA DEL PLÁTANO

Hola, JEAN FREDD. Cuando envíe este formulario, el propietario verá su nombre y dirección de correo electrónico.

1. ¿Cómo califica el nivel de integración de herramientas digitales en las prácticas agrícolas actuales de la Extensión El Carmen?

☐ Muy avanzado

☐ Adecuado

☐ Insuficiente

☐ Inexistente

2. ¿Qué importancia tiene para su formación profesional el aprendizaje de soluciones basadas en Agricultura de Precisión?

☐ Crucial para el mercado laboral

☐ Importante para la especialización

☐ Complementaria, pero no esencial

☐ Irrelevante para mi perfil

3. ¿Cuál considera que es la principal ventaja de utilizar aplicaciones con Inteligencia Artificial para el monitoreo de cultivos?

☐ Reducción del margen de error humano

☐ Optimización del tiempo en prácticas de campo

☐ Acceso a diagnósticos técnicos en tiempo real

☐ Registro digital de datos históricos

4. Ante el desarrollo de una herramienta tecnológica, ¿cuál considera que es el mayor reto para su adopción entre los estudiantes?

☐ Limitaciones de hardware (teléfonos de gama baja)

☐ Falta de capacitación técnica especializada

☐ Dificultad de acceso a internet en áreas de práctica

☐ Resistencia al reemplazo de métodos tradicionales

5. ¿Qué funcionalidad valoraría más en una aplicación móvil diseñada para el entorno de la Granja Experimental?

☐ Interfaz intuitiva y de rápida navegación

☐ Capacidad de funcionamiento en modo offline

☐ Alta precisión en la identificación de patologías

☐ Generación de reportes automáticos en PDF

6. ¿Cree usted que la universidad debería priorizar el desarrollo de software propio para la gestión de sus procesos agropecuarios?

☐ Totalmente de acuerdo

☐ De acuerdo

☐ En desacuerdo

☐ Totalmente en desacuerdo

7. ¿Cuál es su método principal para identificar la Sigatoka Negra (*Pseudocercospora fijiensis*) en las hojas de plátano?

☐ Inspección visual basada en la experiencia

☐ Comparación con guías y atlas fitosanitarios

☐ Consulta directa con docentes o técnicos

☐ Aún no logra identificarla con certeza

8. En sus prácticas de campo, ¿qué nivel de dificultad encuentra para diferenciar visualmente la Sigatoka temprana de una deficiencia nutricional?

☐ Dificultad muy alta

☐ Dificultad moderada

☐ Dificultad baja

☐ Ninguna dificultad

9. ¿Cuál considera que es el factor determinante para que esta enfermedad cause pérdidas económicas tan severas en la región?

☐ Diagnóstico tardío por métodos manuales

☐ Condiciones climáticas extremas de la zona

☐ Resistencia del hongo a los fungicidas actuales

☐ Falta de protocolos de monitoreo constante

10. ¿Estaría dispuesto a validar su criterio técnico comparándolo con el diagnóstico sugerido por una Inteligencia Artificial? ☐
- ☐ Totalmente dispuesto
- ☐ Muy dispuesto
- ☐ Poco dispuesto
- ☐ Nada dispuesto
11. ¿Qué nivel de conocimiento posee usted sobre el impacto de los nematodos en la morfología radicular del plátano? ☐
- ☐ Conocimiento profundo
- ☐ Conocimiento moderado
- ☐ Conocimiento superficial
- ☐ Ningún conocimiento
12. ¿Cómo califica la frecuencia de observación de daños radiculares causados por nematodos en la Granja Experimental? ☐
- ☐ Frecuencia muy alta
- ☐ Frecuencia moderada
- ☐ Escasa frecuencia
- ☐ Nunca ha observado daños
13. ¿Qué tan complejo le resulta identificar síntomas de nematodos sin recurrir a un análisis de laboratorio? ☐
- ☐ Extremadamente complejo
- ☐ Moderadamente complejo
- ☐ Poco complejo
- ☐ Muy sencillo
14. ¿Considera necesario automatizar el registro de áreas afectadas por nematodos dentro de la granja universitaria? ☐
- ☐ Totalmente necesario
- ☐ Muy necesario
- ☐ Poco necesario
- ☐ Innecesario

15. ¿En qué medida considera que las enfermedades fitosanitarias limitan el aprendizaje práctico en los cultivos de la granja? ☐
- ☐ En gran medida
- ☐ De forma considerable
- ☐ De forma mínima
- ☐ No limitan el aprendizaje
16. ¿Cuál es su nivel de confianza al emitir un diagnóstico fitosanitario sin ayuda tecnológica? ☐
- ☐ Confianza total
- ☐ Confianza moderada
- ☐ Confianza baja
- ☐ Ninguna confianza
17. ¿Considera que el uso de software de diagnóstico reduce el impacto ambiental al evitar aplicaciones químicas preventivas? ☐
- ☐ Totalmente de acuerdo
- ☐ De acuerdo
- ☐ En desacuerdo
- ☐ Totalmente en desacuerdo
18. ¿Qué tan probable es que usted utilice una aplicación móvil para apoyar su trabajo de tesis o proyectos de aula? ☐
- ☐ Muy probable
- ☐ Probable
- ☐ Poco probable
- ☐ Nada probable

Ilustración 38 Estructura de la entrevista semiestructurada aplicada al docente especialista de la Granja Experimental ULEAM El Carmen

Pregunta 1:

¿Cuáles son las enfermedades más críticas que afectan actualmente a los cultivos de plátano en la región —por ejemplo, Sigatoka Negra o Mal de Panamá— y cómo varían visualmente sus síntomas en las hojas según la etapa de la infección o la época del año?

Pregunta 2:

Para recomendar un tratamiento o control fitosanitario adecuado, ¿es suficiente con identificar qué enfermedad tiene la planta o es necesario determinar con precisión el porcentaje y el área de la hoja que está afectada? ¿Cómo evalúan ustedes actualmente esa severidad?

Pregunta 3:

Desde su experiencia en el sector agropecuario, ¿qué características clave debería tener una herramienta tecnológica de diagnóstico basada en inteligencia artificial para que tanto los técnicos como los agricultores confíen en sus resultados y la adopten en su día a día?

1.32 Anexo C: Asignación de tutor

Ilustración 39 Asignación de tutor

**Periodo 2025-2 - Notificación de tutor asignado - SOFTWARE
2024 - NS - (EL CARMEN)**

Estimad@
Docente y Estudiante
Uleam

En cumplimiento de lo establecido en la Ley, el Reglamento de Régimen Académico y las disposiciones estatutarias de la Uleam, por medio de la presente se oficializa la dirección y tutoría en el desarrollo del Trabajo de Integración curricular / Trabajo de Titulación del siguiente estudiante:

Tema: SISTEMA INFORMÁTICO CON DANN PARA EL DIAGNÓSTICO DE ENFERMEDADES FITOSANITARIAS EN CULTIVOS DE PLÁTANO DEL CANTÓN EL CARMEN

Estado de aprobación: Aprobado

Tipo de titulación: Trabajo de Integración Curricular

Tipo de proyecto: Trabajo de Integración Curricular / Trabajo de titulación se articula con proyectos y programas de Investigación.

Apellidos y nombres del tutor asignado: LOPEZ RODRIGUEZ CARLOS VINICIO

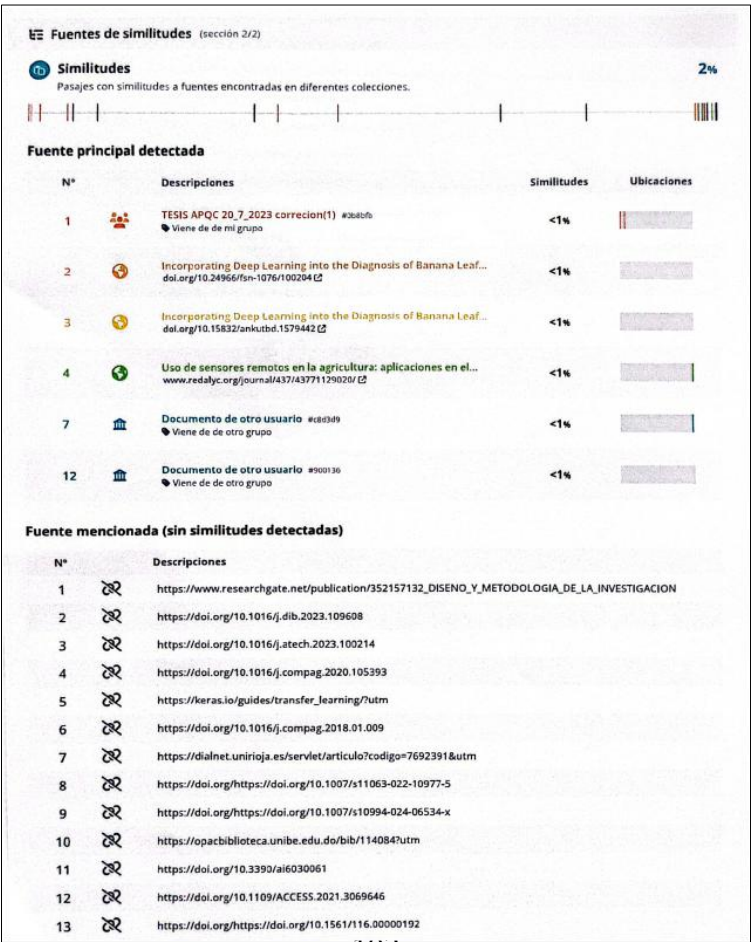
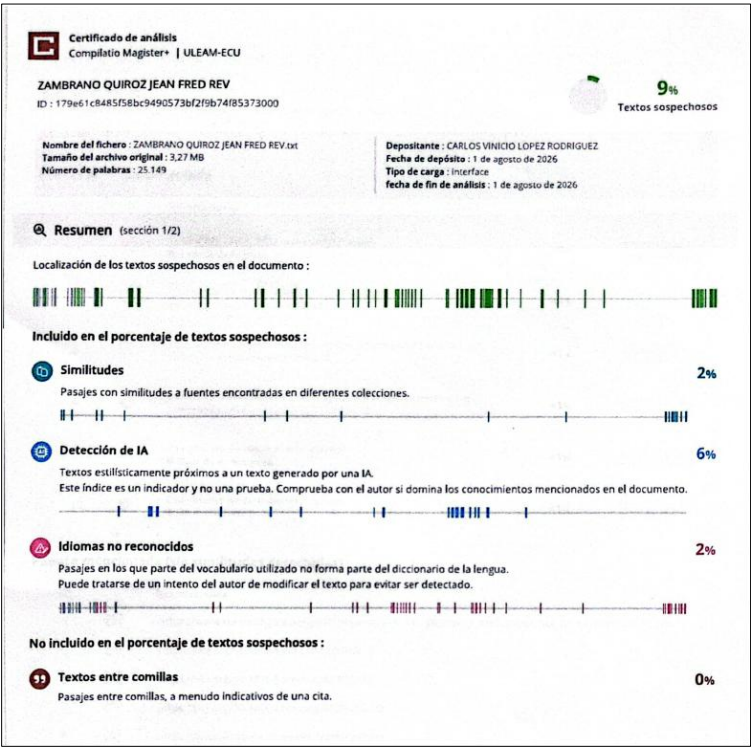
Apellidos y nombres del estudiante: ZAMBRANO QUIROZ JEAN FREDD

Carrera: SOFTWARE 2024 - NS - (EL CARMEN)

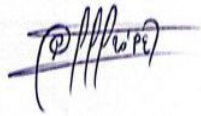
Periodo de inducción: Periodo 2025-2

1.33 Anexo D: Reporte de similitud

Ilustración 40 Reporte de similitud



N°	Descripciones
14	 https://doi.org/https://doi.org/10.5121/csit.2022.120901
15	 https://ebiro.net/es/ledeam/titulos/280024
16	 https://www.marcombo.com/libro/libros-tecnicos-de-arte-y-cientificos/informatica-libros-tecnicos-y-cient...
17	 https://doi.org/10.3390/plants8110468
18	 https://doi.org/10.1016/j.isprsjpts.2020.08.025
19	 https://doi.org/10.1186/s13097-019-0475-z
20	 https://books.google.com.ec/books?id=XI-lugfAAAQBAJ&printsec=frontcover&source=pls_auth#v=onepage...
21	 https://doi.org/https://doi.org/10.1016/j.dib.2023.109155
22	 https://doi.org/https://doi.org/10.3390/agriculture12081192
23	 https://doi.org/https://doi.org/10.37811/cl_rcm.v7i4.7658
24	 https://keras.io/keras_hub/getting_started/



GLOSARIO

Adaptación de dominio: Técnica de aprendizaje automático que busca reducir las diferencias entre un conjunto de datos de origen y otro conjunto de datos capturado en condiciones diferentes, con el fin de mejorar el rendimiento del modelo en escenarios reales.

Aprendizaje profundo: Rama de la inteligencia artificial que utiliza redes neuronales con múltiples capas para aprender patrones complejos a partir de datos, como imágenes, sonidos o textos.

Aprendizaje por transferencia: Técnica que permite utilizar un modelo previamente entrenado en una gran cantidad de datos para adaptarlo a una nueva tarea específica, reduciendo el tiempo y los recursos necesarios para el entrenamiento.

Arquitectura ResNetVD34: Modelo de red neuronal convolucional utilizado para extraer características visuales de las imágenes. En esta investigación se emplea como base para analizar hojas de plátano.

Balanceo de datos: Proceso mediante el cual se organiza un conjunto de imágenes para que las clases tengan una cantidad similar de ejemplos, evitando que el modelo aprenda con mayor preferencia una clase sobre otra.

Brecha de dominio: Diferencia existente entre las imágenes utilizadas para entrenar un modelo y las imágenes capturadas en el entorno donde será aplicado.

Capa de inversión de gradiente: Componente utilizado en DANN que permite entrenar el modelo de forma adversaria, ayudando a que las características aprendidas sean menos dependientes del dominio de origen de las imágenes.

Clasificación binaria: Proceso mediante el cual un modelo clasifica los datos en dos categorías. En esta investigación, las clases son hoja saludable y hoja con salud comprometida.

Conjunto de datos: Grupo organizado de imágenes utilizado para entrenar, validar y evaluar el modelo de inteligencia artificial.

DANN: Siglas de Domain-Adversarial Neural Network. Es una red neuronal adversaria para adaptación de dominio, utilizada para reducir las diferencias entre imágenes públicas y fotografías reales de campo.

Depuración de datos: Proceso de revisión y limpieza del conjunto de imágenes, eliminando archivos duplicados, defectuosos o no adecuados para el entrenamiento del modelo.

Dominio fuente: Conjunto de imágenes públicas utilizadas como base inicial para el entrenamiento del modelo.

Dominio objetivo: Conjunto de fotografías capturadas en condiciones reales de campo, correspondiente al entorno donde se desea aplicar el modelo.

Exactitud: Métrica que indica el porcentaje total de predicciones correctas realizadas por el modelo.

Extractor de características: Parte del modelo encargada de identificar patrones visuales relevantes en una imagen, como color, textura, bordes y forma.

Flutter: Framework de desarrollo utilizado para construir aplicaciones móviles. En esta investigación se emplea para integrar el modelo de inteligencia artificial en una aplicación Android.

F1-score: Métrica que combina precisión y recall, permitiendo evaluar el rendimiento general del modelo cuando se analizan sus aciertos y errores.

Fitosanitario: Término relacionado con la salud de las plantas y el control de enfermedades, plagas o alteraciones que pueden afectar los cultivos.

Funcionamiento offline: Capacidad de una aplicación para ejecutarse sin conexión a Internet. En esta investigación, permite analizar imágenes directamente desde el dispositivo móvil.

Historial local: Registro almacenado en el dispositivo móvil que conserva los resultados de los análisis realizados por la aplicación.

ImageNet: Base de datos de imágenes utilizada comúnmente para preentrenar modelos de visión artificial.

Inferencia móvil: Proceso mediante el cual un modelo de inteligencia artificial realiza predicciones directamente en un dispositivo móvil.

Inteligencia artificial: Área de la informática que permite desarrollar sistemas capaces de realizar tareas que requieren análisis, aprendizaje y toma de decisiones automatizada.

Matriz de confusión: Tabla que permite comparar las predicciones del modelo con los valores reales, mostrando aciertos y errores por cada clase.

Modelo de inteligencia artificial: Sistema computacional entrenado para reconocer patrones y realizar predicciones a partir de datos.

Musa AAB: Grupo taxonómico al que pertenece el plátano, cultivo objeto de estudio en la presente investigación.

Precisión: Métrica que indica cuántas de las predicciones positivas realizadas por el modelo fueron correctas.

Preprocesamiento de imágenes: Conjunto de pasos aplicados antes del entrenamiento o análisis, como redimensionamiento, organización, limpieza y normalización de imágenes.

Recall: Métrica que indica la capacidad del modelo para identificar correctamente los casos pertenecientes a una clase determinada.

Red neuronal convolucional: Tipo de red neuronal especializada en el procesamiento de imágenes, capaz de identificar patrones visuales como bordes, formas, colores y texturas.

Salud comprometida: Categoría utilizada para clasificar hojas de plátano que presentan alteraciones visibles, como manchas, cambios de color, daños o signos de posible afectación.

Sigatoka Negra: Enfermedad fitosanitaria que afecta al cultivo de plátano y banano, asociada al hongo *Mycosphaerella fijiensis*.

TensorFlow Lite: Herramienta que permite ejecutar modelos de inteligencia artificial en dispositivos móviles o sistemas con recursos limitados.

Umbral de confianza: Valor mínimo de probabilidad establecido para aceptar una predicción del modelo. En esta investigación se utiliza un umbral del 65 %.

Visión artificial: Campo de la inteligencia artificial que permite a los sistemas computacionales analizar e interpretar imágenes.