



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE SOFTWARE

TRABAJO DE INTEGRACIÓN CURRICULAR PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN SOFTWARE

TEMA:

**“ DESARROLLO DE UNA APLICACIÓN MÓVIL PARA DETECTAR
ENFERMEDADES EN CULTIVOS DE PAPA MEDIANTE IMÁGENES
DIGITALES E INTELIGENCIA ARTIFICIAL PARA MEJORAR LA
PRODUCCIÓN AGRÍCOLA ”**



AUTOR:

Diego Ernesto Cuaycal Tirira

DIRECTOR:

PhD. Marco Remigio PUSDÁ Chulde

Ibarra – Ecuador
2026



UNIVERSIDAD TÉCNICA DEL NORTE

DIRECCIÓN DE BIBLIOTECA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO	
APELLIDOS Y NOMBRES:	CUAYCAL TIRIRA DIEGO ERNESTO

DATOS DE LA OBRA	
TÍTULO:	DESARROLLO DE UNA APLICACIÓN MÓVIL PARA DETECTAR ENFERMEDADES EN CULTIVOS DE PAPA MEDIANTE IMÁGENES DIGITALES E INTELIGENCIA ARTIFICIAL PARA MEJORAR LA PRODUCCIÓN AGRÍCOLA
AUTOR (ES):	DIEGO ERNESTO CUAYCAL TIRIRA
FECHA: DD/MM/AAAA	16/06/2026
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	INGENIERO EN SOFTWARE
DIRECTOR / ASESOR:	PHD. MARCO REMIGIO PUSDÁ CHULDE

CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 16 días del mes de junio de 2026.

EL AUTOR:

Firma.....

Nombre: Diego Ernesto Cuaycal Tirira

CI: 0401904735

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 16 de junio de 2026.

PhD. Marco Remigio PUSDÁ Chulde
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

PhD. Marco Remigio PUSDÁ Chulde

C.C: 0401200951

APROBACIÓN DEL COMITÉ CALIFICADOR

El Comité Calificador del trabajo de Integración Curricular “Desarrollo de una aplicación móvil para detectar enfermedades en cultivos de papa mediante imágenes digitales e inteligencia artificial para mejorar la producción agrícola” elaborado por Diego Ernesto Cuaycal Tirira , previo a la obtención del título de INGENIERO EN SOFTWARE, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:

PhD. Marco Remigio Pusedá Chulde

C.C: 0401200951

PhD. Iván Danilo García Santillán

C.C: 1002292603

DEDICATORIA

“Haz de tu vida un sueño, y de un sueño, una realidad.”

– **Antoine de Saint-Exupéry**

Dedico este trabajo, en primer lugar, a Dios, mi Padre Celestial, por ser mi guía, mi fortaleza y mi refugio a lo largo de este camino. Gracias por acompañarme y brindarme la fuerza necesaria para alcanzar esta meta.

A mis queridos padres, Henni Tirira y Ernesto Cuaycal, por ser el pilar fundamental de mi vida y por haberme brindado, con amor y sacrificio, la oportunidad de llegar hasta aquí. Gracias por su apoyo incondicional, por cada esfuerzo realizado, por su comprensión en los momentos más difíciles y por guiarme con su ejemplo y sus consejos. A ustedes les debo gran parte de este logro, porque me dieron la educación, los valores y las oportunidades necesarias para formarme personal y profesionalmente. Gracias por confiar en mí, por impulsarme a seguir adelante aun en medio de las dificultades y por no soltarme la mano en ningún momento. Este triunfo también es suyo, porque sin su entrega, dedicación y amor, nada de esto habría sido posible.

A mis hermanas, Diana y Alexandra, por su cariño, compañía y apoyo durante todo este proceso. Gracias por estar presentes en una etapa tan importante de mi vida, por acompañarme en los cambios, en los momentos de dificultad y en la experiencia de vivir lejos de casa. Su presencia, sus palabras y su apoyo fueron un impulso importante para seguir adelante.

Finalmente, dedico este logro a todas aquellas personas que, de una u otra manera, estuvieron presentes a lo largo de mi vida universitaria y contribuyeron con su apoyo, consejos y compañía para que este sueño hoy se haga realidad.

Diego Ernesto Cuaycal Tirira

AGRADECIMIENTO

Expreso mi más sincero agradecimiento a la Universidad Técnica del Norte por abrirme sus puertas y brindarme la oportunidad de formarme como profesional, ofreciéndome una educación de calidad y un espacio que ha contribuido a mi desarrollo académico y personal. Asimismo, agradezco a todo el cuerpo docente de la carrera por compartir sus conocimientos, experiencias y valores, los cuales fueron fundamentales en mi formación durante esta etapa universitaria.

De manera especial, expreso mi más profunda gratitud a mi director, PhD. Marco Pusdá, y a mi asesor, PhD. Iván García, por la confianza depositada en mí para el desarrollo de este proyecto de titulación. Agradezco su guía, paciencia y acompañamiento constante durante todo el proceso, así como sus valiosos aportes y observaciones que contribuyeron significativamente a la culminación de este trabajo.

Agradezco también de manera especial al MSc. Jorge Pozo por su tiempo, colaboración y valiosas recomendaciones, especialmente por su apoyo en la recolección de imágenes y sus conocimientos sobre el cultivo de papa. Asimismo, expreso mi gratitud a quienes, con su experiencia agrícola, contribuyeron al desarrollo de este trabajo.

De igual manera, expreso mi agradecimiento a mis compañeros de aula y amistades con quienes compartí esta etapa universitaria, por su apoyo, colaboración y los conocimientos intercambiados en cada proyecto y experiencia vivida.

Finalmente, agradezco a todas las personas que, de una u otra manera, formaron parte de este proceso y contribuyeron con su apoyo, confianza y colaboración para hacer posible la culminación de esta etapa tan importante en mi vida.

Diego Ernesto Cuaycal Tirira

ÍNDICE DE CONTENIDOS

DEDICATORIA	6
AGRADECIMIENTO	7
ÍNDICE DE FIGURAS	13
ÍNDICE DE TABLAS	15
RESUMEN	18
ABSTRACT	19
CAPÍTULO I	
1.1 Problema de investigación	20
1.2 Objetivos	21
1.2.1 Objetivo General	21
1.2.2 Objetivos Específicos	22
1.3 Alcance	22
1.4 Metodología	24
1.5 Justificación	25
1.5.1 Justificación Tecnológica	25
1.5.2 Justificación Social	25
1.5.3 Justificación Académica	25
CAPÍTULO II	
2.1 Antecedentes Generales	26
2.1.1 Problemática de las enfermedades en cultivos de papa	26
2.1.2 Principales enfermedades que afectan al cultivo de papa	27
2.1.3 Implementación y adopción de la agricultura de precisión en Ecuador	28
2.1.4 Aplicaciones móviles en el sector agropecuario	30
2.2 Fundamentos Teóricos	31

2.2.1	Inteligencia Artificial y su aplicación en la agricultura	31
2.2.2	Visión por computadora en el diagnóstico vegetal.	32
2.2.3	Redes Neuronales Convolucionales (CNN).	33
2.2.4	Modelos de detección en tiempo real (YOLO)	35
2.2.5	Evaluación del rendimiento de modelos de IA (precisión, recall, F1-score, ISO 25010).	36
2.3	Herramientas tecnológicas revisadas en la literatura y su integración al proyecto	36
2.3.1	YOLOv8: modelo para detección de enfermedades.	36
2.3.2	TensorFlow como entorno para diseñar y ejecutar redes neuronales.	37
2.3.3	Creación de interfaces móviles con Flutter	39
2.3.4	Google Cloud para almacenamiento y acceso remoto	39
2.3.5	Integración de Scrum y CRISP-DM para proyectos de IA	40
2.4	Trabajos Relacionados	41
2.4.1	Revisión de tesis y artículos científicos	41
 CAPÍTULO III		
3.1	Metodología general del proyecto	43
3.2	Metodología Scrum para el desarrollo de la aplicación móvil	44
3.2.1	Aplicación de Scrum en el desarrollo de CultivoScan.	44
3.2.2	Definición del equipo Scrum	44
3.2.3	Fase de Pre-Juego: planificación inicial.	45
3.2.3.1	Definición de nomenclatura.	45
3.2.3.2	Levantamiento de requerimientos	46
3.2.3.3	Requerimientos funcionales.	46
3.2.3.4	Requerimientos no funcionales	47
3.2.3.5	Historias de usuario	47
3.2.3.6	Diagrama general de casos de uso	54
3.2.3.7	Product Backlog.	56
3.2.3.8	Prototipado de interfaces en Figma	57
3.2.3.9	Arquitectura de la aplicación móvil.	58

3.2.3.10	Estructura general del proyecto	59
3.2.4	Fase de Juego: desarrollo por sprints	60
3.2.4.1	Planificación general de sprints	61
3.2.4.2	Sprint 0: configuración del entorno y servicios base	62
3.2.4.3	Sprint 1: Bienvenida, autenticación y acceso de usuarios	63
3.2.4.4	Sprint 2: Gestión de parcelas	66
3.2.4.5	Sprint 3: Registro de actividades por parcela	68
3.2.4.6	Sprint 4: Servicios de información agrícola.	70
3.2.4.7	Sprint 5: Captura de imágenes y diagnóstico con inteligencia artificial.	72
3.2.4.8	Sprint 6: Historial, monitoreo y configuración	76
3.2.4.9	Matriz de trazabilidad Scrum.	79
3.2.5	Fase de Post-Juego: pruebas, ajustes y publicación	80
3.2.5.1	Pruebas funcionales de la aplicación móvil.	81
3.2.5.2	Corrección de errores y ajustes finales	82
3.2.5.3	Publicación en Google Play Store	83
3.2.5.4	Preparación para evaluación con usuarios	83
3.3	Metodología CRISP-DM para el desarrollo del modelo de inteligencia artificial	84
3.3.1	Aplicación de CRISP-DM en el desarrollo del modelo	85
3.3.2	Fase 1: Comprensión del negocio	85
3.3.3	Fase 2: Comprensión y adquisición de los datos	86
3.3.3.1	Fuentes del conjunto de datos	87
3.3.3.2	Recolección de imágenes propias en campo	87
3.3.3.3	Justificación de la recolección de la clase Septoria	89
3.3.3.4	Clases consideradas para el diagnóstico	89
3.3.3.5	Distribución del conjunto de datos	90
3.3.4	Fase 3: Preparación de los datos.	91
3.3.4.1	Organización y depuración de imágenes	92
3.3.4.2	Anotación de imágenes y estrategia de etiquetado en Roboflow	92

3.3.4.3	División del conjunto de datos	93
3.3.4.4	Preprocesamiento aplicado al conjunto de datos	94
3.3.4.5	Aumento de datos	95
3.3.5	Fase 4: Modelado	96
3.3.5.1	Entorno de entrenamiento local	96
3.3.5.2	Selección de arquitecturas de detección	97
3.3.5.3	Configuración de los experimentos	98
3.3.5.4	Matriz de diseño experimental	99
3.3.5.5	Entrenamiento de los modelos	100
3.3.6	Fase 5: Evaluación	101
3.3.6.1	Métricas de evaluación utilizadas	101
3.3.6.2	Protocolo de evaluación en validación y prueba	102
3.3.6.3	Evaluación comparativa de modelos	102
3.3.6.4	Criterio de selección del modelo para la aplicación móvil	103
3.3.7	Fase 6: Despliegue del modelo	103
3.3.7.1	Exportación del modelo a TensorFlow Lite	104
3.3.7.2	Integración del modelo en Flutter	104
3.3.7.3	Inferencia local en el dispositivo móvil	105
3.3.7.4	Postprocesamiento visual de zonas afectadas	106

CAPÍTULO IV

4.1	Resultados de la aplicación móvil	107
4.1.1	Instrumento de evaluación aplicado	107
4.1.2	Criterios de interpretación de la escala de evaluación	108
4.1.3	Relación de la evaluación con ISO/IEC 25010	109
4.1.4	Caracterización de los participantes	109
4.1.5	Resultados de eficacia percibida	111
4.1.6	Resultados de eficiencia percibida	112
4.1.7	Percepción general y oportunidades de mejora	112

4.1.8	Análisis general de la evaluación de la aplicación móvil.	113
4.2	Resultados del modelo de inteligencia artificial	114
4.2.1	Resultados generales de los experimentos	114
4.2.2	Comparación de arquitecturas de detección.	117
4.2.3	Análisis del efecto de la resolución de entrada	118
4.2.4	Análisis del modelo seleccionado	119
4.2.5	Resultados por clase del modelo seleccionado.	121
4.2.6	Dinámica de entrenamiento del modelo seleccionado	122
4.2.7	Análisis gráfico del rendimiento del modelo seleccionado	122
4.2.8	Análisis visual de predicciones	124
4.2.9	Interpretabilidad visual mediante EigenCAM	125
4.2.10	Comparación con trabajos relacionados.	126
4.2.11	Análisis final de los resultados del modelo de inteligencia artificial	127
	CONCLUSIONES	128
	RECOMENDACIONES	130
	BIBLIOGRAFÍA	
	ANEXOS	142
	Anexo A. Instrumento de evaluación de la aplicación móvil.	142
	Anexo B. Matriz de respuestas de la encuesta	144
	Anexo C. Publicación de la aplicación móvil en Google Play Store	145
	Anexo D. Repositorio de código y cuadernos de experimentación	146
	Anexo E. Publicación del conjunto de datos	147

ÍNDICE DE FIGURAS

Figura 1	Árbol causal	21
Figura 2	Arquitectura funcional del sistema	23
Figura 3	Ciclo de la agricultura de precisión	29
Figura 4	Procesamiento de imágenes para diagnóstico vegetal	33
Figura 5	Arquitectura CNN para clasificación de enfermedades foliares	35
Figura 6	Diseño de los componentes de YOLOv8 para detección en cultivos.	37
Figura 7	Funcionalidades principales de TensorFlow para modelos de IA	38
Figura 8	Relación entre Scrum y CRISP-DM en proyectos de inteligencia artificial	41
Figura 9	Diagrama general de casos de uso de la aplicación móvil CultivoScan . . .	55
Figura 10	Prototipo general de interfaces diseñado en Figma	57
Figura 11	Arquitectura de la aplicación móvil	59
Figura 12	Estructura general del proyecto Flutter	60
Figura 13	Pantallas de bienvenida de la aplicación móvil	64
Figura 14	Pantallas de autenticación de la aplicación móvil	65
Figura 15	Pantallas del módulo de gestión de parcelas	67
Figura 16	Pantallas del módulo de actividades por parcela	69
Figura 17	Pantallas del módulo de pronóstico climático	71
Figura 18	Pantallas del módulo de precio de mercado de la papa	72
Figura 19	Diagrama de secuencia del diagnóstico con inteligencia artificial	74
Figura 20	Pantallas del módulo de diagnóstico con inteligencia artificial	75
Figura 21	Pantallas de historial, monitoreo y configuración	78
Figura 22	Publicación de la aplicación móvil mediante Google Play Store.	83
Figura 23	Recolección de imágenes en parcelas agrícolas de la provincia del Carchi .	88
Figura 24	Ejemplo de anotación de hoja en Roboflow	93
Figura 25	Ubicación del modelo <i>TensorFlow Lite</i> en la estructura del proyecto Flutter	105

Figura 26	Ejemplo del postprocesamiento visual de zonas afectadas mediante HSV .	106
Figura 27	Comparación del mAP50 en el conjunto de prueba para los experimentos evaluados	116
Figura 28	Estudio de ablación del efecto de la resolución de entrada en la estrategia BoxGain	119
Figura 29	Curvas de entrenamiento del modelo seleccionado	122
Figura 30	Curva precisión-recall del modelo seleccionado	123
Figura 31	Curva F1-confianza correspondiente al modelo seleccionado	123
Figura 32	Matriz de confusión normalizada del modelo seleccionado	124
Figura 33	Ejemplos de predicción del modelo seleccionado en el conjunto de prueba	125
Figura 34	Mapa de activación EigenCAM generado sobre una predicción del modelo seleccionado	126
Figura 35	Sección de datos generales del instrumento de evaluación	142
Figura 36	Sección de evaluación de la precisión del sistema	143
Figura 37	Sección de evaluación de la eficiencia de la aplicación	143
Figura 38	Sección de comentarios y percepción general	144

ÍNDICE DE TABLAS

Tabla 1	Enfermedades foliares consideradas para la detección visual en hojas de papa	27
Tabla 2	Comparativa de aplicaciones móviles agropecuarias orientadas al diagnóstico de enfermedades	31
Tabla 3	Metodologías utilizadas en el desarrollo del proyecto	43
Tabla 4	Equipo Scrum del proyecto	44
Tabla 5	Nomenclatura utilizada para la planificación Scrum	45
Tabla 6	Requerimientos funcionales	46
Tabla 7	Requerimientos no funcionales	47
Tabla 8	Formato de historias de usuario	48
Tabla 9	Estimación de esfuerzo	48
Tabla 10	Historia de usuario HU01	49
Tabla 11	Historia de usuario HU02	49
Tabla 12	Historia de usuario HU03	50
Tabla 13	Historia de usuario HU04	51
Tabla 14	Historia de usuario HU05	51
Tabla 15	Historia de usuario HU06	52
Tabla 16	Historia de usuario HU07	52
Tabla 17	Historia de usuario HU08	53
Tabla 18	Historia de usuario HU09	53
Tabla 19	Historia de usuario HU10	54
Tabla 20	Product Backlog de la aplicación móvil	56
Tabla 21	Distribución general de sprints del proyecto	61
Tabla 22	Definición de terminado aplicada durante los sprints	62
Tabla 23	Sprint Backlog – Sprint 0	62
Tabla 24	Sprint Backlog – Sprint 1	63

Tabla 25	Revisión de criterios de aceptación – Sprint 1	65
Tabla 26	Sprint Backlog – Sprint 2	66
Tabla 27	Revisión de criterios de aceptación – Sprint 2	67
Tabla 28	Sprint Backlog – Sprint 3	68
Tabla 29	Revisión de criterios de aceptación – Sprint 3	70
Tabla 30	Sprint Backlog – Sprint 4	70
Tabla 31	Revisión de criterios de aceptación – Sprint 4	72
Tabla 32	Sprint Backlog – Sprint 5	73
Tabla 33	Revisión de criterios de aceptación – Sprint 5	76
Tabla 34	Sprint Backlog – Sprint 6	77
Tabla 35	Revisión de criterios de aceptación – Sprint 6	79
Tabla 36	Matriz de trazabilidad Scrum de la aplicación móvil	80
Tabla 37	Pruebas funcionales de la aplicación móvil	81
Tabla 38	Correcciones y ajustes realizados en la aplicación móvil	82
Tabla 39	Fases de CRISP-DM aplicadas al modelo de inteligencia artificial	85
Tabla 40	Parámetros considerados durante la recolección de imágenes propias	88
Tabla 41	Clases consideradas para el diagnóstico del modelo	90
Tabla 42	Conteo de imágenes por clase y procedencia	91
Tabla 43	División final del conjunto de datos utilizado en el entrenamiento	94
Tabla 44	Técnicas de preprocesamiento aplicadas en Roboflow	94
Tabla 45	Técnicas de aumento de datos aplicadas en Roboflow	95
Tabla 46	Configuración local empleada en el entrenamiento de los modelos	97
Tabla 47	Arquitecturas consideradas durante el modelado	98
Tabla 48	Parámetros e hiperparámetros considerados en la configuración de los experimentos	99
Tabla 49	Síntesis del diseño experimental de los modelos evaluados	100
Tabla 50	Métricas utilizadas para la evaluación de modelos de detección	102
Tabla 51	Componentes del instrumento de evaluación aplicado	107
Tabla 52	Valores asignados a la escala Likert	108
Tabla 53	Criterios de valoración según la media obtenida	108
Tabla 54	Relación entre la evaluación aplicada e ISO/IEC 25010	109

Tabla 55	Caracterización de los participantes encuestados	110
Tabla 56	Resultados de eficacia percibida	111
Tabla 57	Resultados de eficiencia percibida	112
Tabla 58	Síntesis de oportunidades de mejora identificadas	113
Tabla 59	Comparación entre resultados de validación y prueba en modelos representativos	114
Tabla 60	Resultados generales de los experimentos en el conjunto de prueba	115
Tabla 61	Comparación resumida de arquitecturas de detección	117
Tabla 62	Comparación del desempeño según la resolución de entrada	118
Tabla 63	Resumen técnico del modelo con mejor rendimiento experimental	120
Tabla 64	Criterios técnicos considerados para la selección del modelo final	120
Tabla 65	Resultados por clase del modelo seleccionado	121
Tabla 66	Comparación del modelo seleccionado con un trabajo relacionado	127
Tabla 67	Matriz codificada de respuestas tipo Likert	145
Tabla 68	Publicación de la aplicación móvil en Google Play Store	146
Tabla 69	Repositorio de código y cuadernos de experimentación	147
Tabla 70	Descripción general de los cuadernos de experimentación	147
Tabla 71	Publicación del conjunto de datos en Zenodo	148

RESUMEN

El presente trabajo de integración curricular tuvo como objetivo desarrollar una aplicación móvil para apoyar la detección de enfermedades foliares en cultivos de papa mediante imágenes digitales e inteligencia artificial. La investigación se orientó a brindar una herramienta tecnológica de apoyo al diagnóstico agrícola, considerando que la identificación visual de enfermedades depende en gran medida de la experiencia del productor y de la disponibilidad de asistencia técnica. Para el desarrollo de la aplicación móvil CultivoScan se aplicó la metodología Scrum, mientras que el componente de inteligencia artificial se estructuró mediante CRISP-DM. Se conformó un conjunto de datos a partir de imágenes públicas y fotografías tomadas en campo, organizadas en las clases *Alternaria*, *Lancha*, *Septoria* y *Hoja sana*. Cada imagen fue anotada con cajas delimitadoras, orientadas a la detección de objetos a nivel de hoja. En la fase experimental se probaron distintas arquitecturas de detección, entre ellas variantes YOLO y RT-DETR; la mejor configuración correspondió al modelo YOLO-26-N con ajuste BoxGain y una resolución de entrada de 1024 píxeles. Dicho modelo obtuvo un mAP50 de 98.89 %, un mAP50-95 de 98.05 %, una precisión de 96.99 % y un recall de 95.76 % en el conjunto de prueba. La aplicación incorporó el modelo seleccionado para analizar imágenes y presentar el diagnóstico generado. La evaluación con usuarios también mostró una valoración favorable, con una media de 4.77 sobre 5 en eficacia percibida y 4.78 sobre 5 en eficiencia percibida. Estos resultados muestran que la integración de una aplicación móvil con un modelo de detección de objetos puede funcionar como una solución tecnológica accesible para apoyar la toma de decisiones agrícolas en el diagnóstico de enfermedades en hojas de papa.

Palabras clave: aplicación móvil, inteligencia artificial, detección de objetos, enfermedades foliares, cultivo de papa, YOLO, agricultura de precisión.

ABSTRACT

This curricular integration project aimed to develop a mobile application to support the detection of foliar diseases in potato crops using digital images and artificial intelligence. The research sought to provide a technological tool to support agricultural diagnosis, considering that the visual identification of diseases largely depends on the farmer's experience and the availability of technical assistance. For the development of the CultivoScan mobile application, the Scrum methodology was applied, while the artificial intelligence component was structured using CRISP-DM. A dataset was constructed from public images and field-collected photographs, organized into the classes Early blight, Late blight, Septoria leaf spot, and Healthy leaf. The images were annotated with bounding boxes for leaf-level object detection tasks. During the experimental phase, different detection architectures were evaluated, including YOLO and RT-DETR variants, with the YOLO-26-N model using BoxGain adjustment and an input resolution of 1024 pixels identified as the best configuration. This model achieved an mAP50 of 98.89 %, an mAP50-95 of 98.05 %, a precision of 96.99 %, and a recall of 95.76 % on the test set. The application integrated the selected model to enable image analysis and visualization of the generated diagnosis. In addition, the user evaluation showed a positive perception, with an average score of 4.77 out of 5 for perceived effectiveness and 4.78 out of 5 for perceived efficiency. It is concluded that integrating a mobile application with an object detection model constitutes an accessible technological solution to support agricultural decision-making in the diagnosis of potato leaf diseases.

Keywords: mobile application, artificial intelligence, object detection, potato leaf diseases, potato crop, YOLO, precision agriculture.

CAPÍTULO I

INTRODUCCIÓN

1.1 Problema de investigación

En la provincia del Carchi, los cultivos de papa enfrentan una problemática persistente debido a enfermedades y plagas que afectan directamente su desarrollo. Patógenos como hongos, bacterias y virus provocan una disminución significativa en la calidad y cantidad de la producción. Además, las condiciones climáticas variables de la región favorecen la propagación de estos agentes, intensificando su impacto [1].

Estudios realizados en provincias como Cotopaxi han evidenciado la presencia de enfermedades como el mildiu y la rozadura [2], las cuales, según observaciones locales, también afectan a cultivos en Carchi, generando un escenario de alta vulnerabilidad agrícola. A esto se suma el aumento sostenido de los costos de producción, ya que los agricultores deben aplicar agroquímicos con mayor frecuencia, reduciendo así su rentabilidad. Según un diagnóstico sobre la producción de papa en la zona 1 del Ecuador, las pérdidas pueden oscilar entre el 7 % y el 30 %, dependiendo del tamaño del productor, lo que evidencia una afectación considerable a la sostenibilidad económica del sector [3].

Frente a este escenario de creciente afectación productiva y económica, los agricultores en la provincia del Carchi carecen de métodos precisos para detectar enfermedades en los cultivos de papa, ya que el diagnóstico se basa principalmente en la observación visual empírica. Esta práctica, limitada y propensa al error, dificulta la identificación certera de los patógenos presentes en las distintas fases del cultivo, lo que retrasa la toma de decisiones fundamentales. A esta problemática se suma la escasa formación técnica del personal agrícola y la falta de acceso a herramientas modernas de diagnóstico, lo que profundiza la incertidumbre en el campo. Como

resultado, la enfermedad avanza sin ser identificada con claridad, debilitando progresivamente el sistema productivo, reduciendo su eficiencia y colocando en situación de vulnerabilidad a los agricultores de la región.

A continuación, en la Figura 1 se representa el árbol causal del problema de investigación, considerando sus principales causas, efectos y el problema central identificado.



Figura 1.
Árbol causal

Fuente: Elaboración propia.

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar una aplicación móvil para detectar enfermedades en cultivos de papa mediante imágenes digitales e inteligencia artificial para mejorar la producción agrícola.

1.2.2 Objetivos Específicos

- Definir los fundamentos teóricos sobre el uso de procesamiento de imágenes e inteligencia artificial en la detección de enfermedades agrícolas, estableciendo su impacto en la productividad de los cultivos.
- Desarrollar una aplicación móvil utilizando el modelo YOLO para detectar y clasificar enfermedades en cultivos de papa, integrando procesamiento de imágenes y técnicas de segmentación.
- Evaluar la precisión y eficiencia del sistema implementado mediante pruebas de validación en campo, aplicando la norma ISO/IEC 25010 para comparar su rendimiento con métodos tradicionales, utilizando métricas de calidad del software específicas.

1.3 Alcance

Este proyecto tiene como objetivo desarrollar una aplicación móvil para detectar y monitorear enfermedades en cultivos de papa, utilizando inteligencia artificial y procesamiento de imágenes en tiempo real. El sistema utilizará dispositivos móviles para capturar fotografías a nivel del suelo, sin incluir el uso de drones, enfocándose en zonas afectadas para optimizar decisiones agrícolas [4].

El modelo de inteligencia artificial se basará en la arquitectura YOLO, optimizada para la detección rápida y precisa de objetos en imágenes del espectro visible (RGB). El sistema se centrará exclusivamente en la identificación de enfermedades con manifestaciones visuales en hojas de papa, principalmente aquellas causadas por hongos como son “Tizón tardío”, “Tizón temprano” y “Septoria”. El proyecto no contempla la detección de bacterias ni virus sin expresión visual clara, lo que permite delimitar el alcance a diagnósticos inmediatos y mejorar la efectividad en la gestión temprana de enfermedades, contribuyendo así a la reducción de pérdidas productivas y al uso eficiente de los recursos [5], [6].

Se utilizará Flutter para desarrollar una interfaz multiplataforma, intuitiva y accesible, enfocada en responder a las necesidades de los agricultores que trabajan en zonas rurales. El procesamiento de imágenes y el entrenamiento del modelo de inteligencia artificial se llevarán a cabo mediante TensorFlow, con apoyo de la infraestructura de Google Cloud, lo cual permitirá trabajar con imágenes obtenidas directamente en campo. Para ello, se elaborará un dataset propio compuesto por imágenes de hojas de papa afectadas por hongos, capturadas bajo condiciones reales de campo, que servirá como base para las fases de entrenamiento y validación del modelo. El proyecto se desarrollará bajo la metodología Scrum, con el fin de organizar el trabajo en ciclos iterativos y mantener una construcción colaborativa del prototipo. Esta metodología también permitirá ajustar el desarrollo cuando surjan cambios durante la implementación.

El alcance contempla el diseño y validación de un prototipo funcional bajo condiciones controladas, con la posibilidad de escalar su uso a otros cultivos, regiones y contextos agrícolas reales. Además, se considerarán criterios de calidad del software definidos en la norma ISO/IEC 25010, enfocándose en la eficacia del sistema en el contexto del usuario final [7].

En la Figura 2 se resume la arquitectura funcional del sistema, desde la captura de imágenes en campo hasta el procesamiento inteligente y la gestión de resultados en la aplicación móvil.

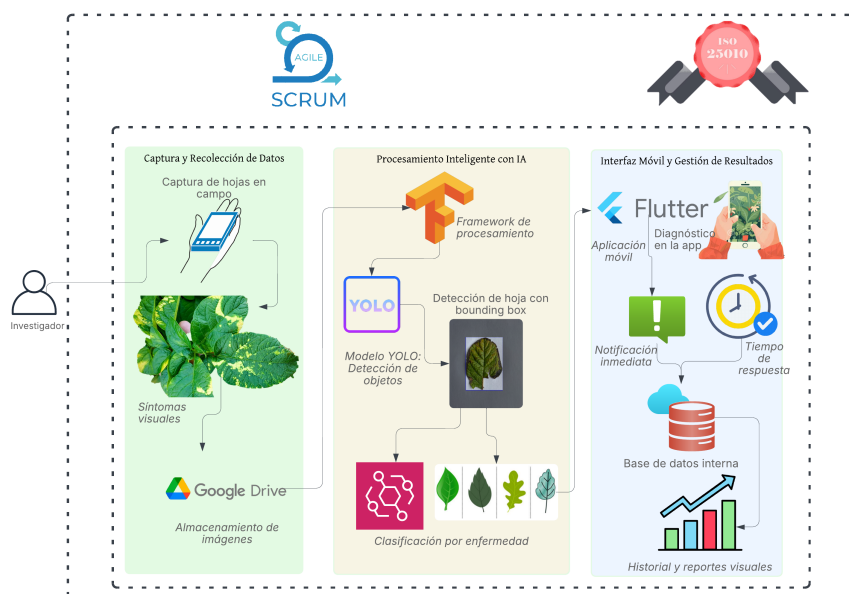


Figura 2.
Arquitectura funcional del sistema

Fuente: Elaboración propia.

1.4 Metodología

El presente trabajo de investigación es de tipo aplicado, con enfoque tecnológico y alcance experimental, ya que se orienta al desarrollo de una aplicación móvil para la detección de enfermedades en cultivos de papa mediante imágenes digitales e inteligencia artificial [8]. Se aplicará la metodología ágil Scrum, lo que permitirá una gestión iterativa, colaborativa y flexible del proceso. Para el cumplimiento de las actividades relacionadas con los objetivos del proyecto, se desarrollarán las siguientes actividades principales:

1. **Actividad 1:** Se realizará una investigación documental y técnica sobre el uso de inteligencia artificial y procesamiento de imágenes en el sector agrícola, con el fin de sustentar el diseño del modelo. Se consultarán fuentes científicas de libre acceso mediante el gestor Mendeley y se aplicará la metodología CRISP-DM, que guiará las fases de comprensión del negocio, preparación de datos, modelado, evaluación y validación.
2. **Actividad 2:** Desarrollar una solución móvil con la metodología Scrum, implementando el modelo YOLOv8 o superior, según la disponibilidad tecnológica. El entrenamiento se basará en el dataset público PlantVillage, usando únicamente las clases de hojas de papa afectadas por hongos (Potato__Early_blight, Potato__Late_blight y Potato__healthy), complementado con imágenes recolectadas en campo. Luego se construirá la aplicación con Flutter y arquitectura reactiva, permitiendo capturar imágenes, detectar enfermedades y almacenar resultados [9].
3. **Actividad 3:** Evaluar el prototipo mediante pruebas funcionales, encuestas y criterios de calidad en uso de la norma ISO/IEC 25010, con énfasis en la eficacia. Esta validación permitirá conocer la utilidad del sistema desde la experiencia del usuario final.

1.5 Justificación

En el Carchi, las enfermedades en cultivos de papa pueden generar pérdidas importantes cuando no se detectan a tiempo. Frente a esta situación, la propuesta plantea una herramienta móvil que ayude al agricultor a reconocer señales tempranas en las hojas y tomar decisiones con mayor rapidez. Además, al estar pensada para zonas rurales, busca acercar tecnología accesible a productores que no siempre cuentan con asistencia técnica permanente.

1.5.1 Justificación Tecnológica

La implementación del sistema para detectar enfermedades en cultivos de papa mediante inteligencia artificial constituye una solución innovadora para la agricultura del Carchi. Utiliza imágenes capturadas con dispositivos móviles, procesadas con TensorFlow y servicios en la nube, permitiendo diagnósticos rápidos, automáticos y accesibles en zonas rurales [10].

1.5.2 Justificación Social

En varias zonas rurales del Carchi, los pequeños agricultores toman decisiones sobre sus cultivos con recursos técnicos limitados. Esta situación puede dificultar la identificación oportuna de enfermedades y aumentar el riesgo de pérdidas en la producción. Frente a ello, la aplicación plantea un apoyo accesible para acompañar el manejo del cultivo de papa y favorecer prácticas agrícolas más sostenibles.

1.5.3 Justificación Académica

El desarrollo de una aplicación basada en la norma ISO/IEC 25010 y gestionada mediante la metodología ágil Scrum aportará al fortalecimiento de habilidades técnicas y prácticas relacionadas con la evaluación de calidad de software, generando una experiencia de aprendizaje significativa para estudiantes e investigadores.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes Generales

2.1.1 Problemática de las enfermedades en cultivos de papa

En Ecuador, el cultivo de la papa es una actividad agrícola de gran importancia que representa la principal fuente de ingresos para un gran número de agricultores que habitan en las zonas rurales de la región Sierra, donde esta práctica productiva se enfrenta constantemente a diversas amenazas sanitarias, entre ellas el tizón tardío, el tizón temprano y una variedad de virus que afectan tanto a las hojas como a los tallos, siendo estos patógenos los responsables de comprometer la calidad del cultivo y causar una considerable disminución en el rendimiento por hectárea, situación que inevitablemente conlleva a pérdidas económicas significativas para los productores [1].

En este contexto, la mayoría de los pequeños agricultores carecen de apoyo técnico especializado y se ven obligados a identificar las enfermedades presentes en sus cultivos a través de la simple observación visual y la experiencia transmitida a través de generaciones, procedimiento limitado que muchas veces conduce a errores debido a la similitud de los síntomas de diversas infecciones durante sus primeras etapas de desarrollo, lo que genera consecuencias negativas como la aplicación incorrecta de agroquímicos o la implementación tardía de tratamientos fitosanitarios, favoreciendo así la propagación acelerada de la enfermedad dentro de los cultivos [2].

En respuesta a esta problemática, se ha elaborado una tabla que sistematiza las principales enfermedades que afectan al cultivo de la papa y sus respectivos signos característicos, poniendo

de manifiesto la dificultad del diagnóstico precoz basado únicamente en la inspección visual. Estos elementos avalan la necesidad de incorporar herramientas tecnológicas avanzadas que faciliten la detección automatizada a través de aplicaciones móviles potenciadas por algoritmos de inteligencia artificial, con el fin de mejorar la precisión diagnóstica y optimizar la gestión sanitaria de estos cultivos estratégicos.

En la Tabla 1 se establece la base de enfermedades seleccionadas para orientar el proceso de detección visual propuesto en la investigación.

Tabla 1.

Enfermedades foliares consideradas para la detección visual en hojas de papa

Enfermedad	Agente causal	Síntomas visibles en hojas
Tizón tardío / Lancha	<i>Phytophthora infestans</i>	Manchas oscuras e irregulares, de aspecto húmedo; avanzan con rapidez en alta humedad.
Tizón temprano / Alternaria	<i>Alternaria solani</i>	Manchas circulares o angulares con anillos concéntricos; suelen iniciar en hojas inferiores.
Septoria	<i>Septoria lycopersici</i> / <i>Septoria spp.</i>	Manchas pequeñas, marrones o grisáceas, con bordes oscuros y clorosis alrededor.

Fuente: Adaptado de [11].

2.1.2 Principales enfermedades que afectan al cultivo de papa

El cultivo de la papa, característico de la región andina, enfrenta una alta susceptibilidad a múltiples enfermedades que afectan negativamente su productividad y valor comercial. Estas enfermedades se clasifican de acuerdo a la naturaleza del agente patógeno en hongos, bacterias y virus [2], cada una con la capacidad de afectar diferentes partes de la planta como tubérculos, raíces, tallos o follaje, causando consecuencias de diversa magnitud sobre el cultivo, aunque en el contexto ecuatoriano, las enfermedades que generan mayor impacto en los sistemas productivos locales suelen originarse por la acción de hongos, cuya acción suele ser causada por hongos, bacterias y virus. En el contexto ecuatoriano, sin embargo, las enfermedades que mayor impacto generan en los sistemas productivos locales suelen ser originadas por la acción de hongos, cuya

rápida diseminación se ve favorecida por las condiciones climáticas de humedad y temperaturas templadas que prevalecen en varias zonas agrícolas del país.

Dado que los sistemas de detección computacional dependen de características observables en la superficie foliar [10], esta decisión responde a criterios tanto técnicos como metodológicos. El tizón tardío, el tizón temprano y la septoria presentan signos visibles en el follaje que pueden diferenciarse durante la inspección de las hojas. Por esta razón, estas enfermedades resultan adecuadas para el desarrollo de herramientas inteligentes enfocadas en apoyar el diagnóstico.

Teniendo en cuenta la variedad de patologías existentes, la presente investigación restringe su alcance al estudio de enfermedades fúngicas que se manifiestan visualmente en las hojas. Dentro de este enfoque, es pertinente señalar que existen otras enfermedades de relevancia agronómica como la sarna común o la marchitez bacteriana, que no forman parte de este estudio por no mostrar síntomas visibles en el follaje o requerir métodos de detección más sofisticados, por lo que la investigación se centra en un conjunto representativo de afecciones cuya identificación a partir de imágenes es factible, permitiendo así establecer un marco práctico y realista acorde con las capacidades actuales que ofrece la inteligencia artificial en el campo del análisis visual de plantas.

2.1.3 Implementación y adopción de la agricultura de precisión en Ecuador

En la práctica, la agricultura de precisión integra herramientas como plataformas digitales, sensores avanzados, sistemas GNSS y modelos de IA para vigilar en todo momento el estado del suelo y la vegetación. De esta manera, se pueden tomar decisiones de riego y fertilización de forma puntual, lo cual impulsa mayores rendimientos, minimiza desperdicios y optimiza el uso de insumos [12]. En el caso de la papa, estas tecnologías resultan especialmente valiosas, pues permiten detectar con antelación plagas y enfermedades, así como aplicar nutrientes y productos fitosanitarios únicamente donde se necesitan.

En el contexto ecuatoriano, la incorporación de la agricultura de precisión ha mostrado un

ritmo de adopción limitado y desigual, siendo más evidente en las grandes zonas agrícolas de la región costera, mientras que en la Sierra, donde predomina la agricultura de subsistencia y familiar, la implementación de estas tecnologías es escasa debido a la falta de acceso a internet confiable, la falta de dispositivos móviles adecuados y el insuficiente conocimiento técnico necesario para aprovechar las herramientas digitales, aunque existen iniciativas promovidas por universidades, organizaciones no gubernamentales y entidades estatales que han impulsado el uso de tecnologías como drones, aplicaciones móviles y estaciones meteorológicas para apoyar la toma de decisiones agrícolas, aunque estas experiencias siguen siendo puntuales y carecen de un carácter sistemático [7].

Los pequeños agricultores de las regiones agrícolas de Ecuador pueden utilizar aplicaciones móviles que procesan imágenes de sus cultivos en el propio dispositivo para identificar plagas y enfermedades de forma temprana sin necesidad de equipos costosos ni conocimientos técnicos avanzados. Se elimina la brecha digital, se democratiza la innovación y todos los agricultores tienen acceso a información precisa para modificar las prácticas de riego, fertilización y gestión de plagas con recursos mínimos al simplificar el acceso a esta tecnología. En la Figura 3 se resume el ciclo de la agricultura de precisión.

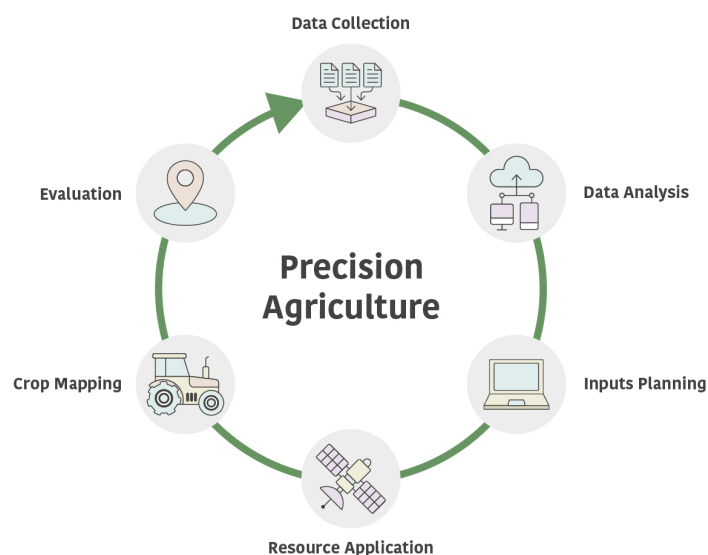


Figura 3.

Ciclo de la agricultura de precisión

Fuente: Adaptado de [13].

2.1.4 Aplicaciones móviles en el sector agropecuario

El desarrollo de las tecnologías móviles ha generado profundas transformaciones en diferentes sectores productivos, siendo la agricultura uno de los ámbitos que más cambios ha experimentado derivados de la incorporación de estas herramientas, que han permitido a los agricultores una gestión más eficiente de sus cultivos mediante el acceso a información en tiempo real, la monitorización de variables ambientales y la recepción de alertas o recomendaciones ajustadas a sus necesidades, configurando así una práctica agrícola más eficiente, precisa y sostenible, especialmente valiosa en escenarios donde los recursos son limitados y las decisiones deben tomarse con agilidad [14].

Las aplicaciones móviles diseñadas para el sector agrícola presentan una diversidad funcional que va desde la monitorización del clima, la gestión del riego y el control de plagas y enfermedades hasta la comercialización de productos y la prestación de asistencia técnica, integrando en algunos casos el uso de sensores externos o algoritmos de inteligencia artificial, mientras que otras soluciones destacan por su funcionamiento offline, lo que supone una ventaja en entornos rurales con conectividad limitada, siendo ejemplos destacados Plantix, Agrobases y Cropsafe, plataformas que han demostrado la viabilidad de ofrecer soluciones tecnológicas prácticas y escalables orientadas a pequeños y medianos productores [15].

En Ecuador, la adopción de estas tecnologías móviles en la agricultura se encuentra en una etapa incipiente, aunque ya se han realizado esfuerzos relevantes por parte de entidades como el Ministerio de Agricultura y Ganadería y varios centros de investigación que han impulsado plataformas de asistencia agrícola y programas de capacitación digital para productores. No obstante, la mayoría de estas herramientas carecen de adaptaciones específicas para ciertos cultivos y no consideran las particularidades locales, por lo que el desarrollo de una aplicación móvil orientada a la detección de enfermedades en el cultivo de papa mediante el uso de imágenes digitales es una alternativa concreta, contextualizada y con un alto potencial de impacto para los pequeños productores de la región Sierra.

En la Tabla 2 se contrastan aplicaciones móviles usadas como referencia para el sistema propuesto.

Tabla 2.

Comparativa de aplicaciones móviles agropecuarias orientadas al diagnóstico de enfermedades

Aplicación	Función principal	Enfermedades tratadas	Uso de IA	Modo offline	Cultivos principales	Público objetivo
Plantix	Diagnóstico de enfermedades vía imagen	Tizón tardío, marchitez, roya	Sí	Sí	Papa, tomate, maíz, arroz	Agricultores pequeños y medianos
AgroAI	Reconocimiento de enfermedades en hojas	General para cultivos comunes	Sí	Parcial	Papa, trigo, arroz	Productores con conectividad básica
Cropsafe	Monitoreo de cultivos vía satélite e IA	Enfermedades visuales generales	Sí	No	Papa, caña, cereales	Productores medianos y técnicos
AgriDoc	Asistencia técnica y diagnóstico remoto	Variadas	No	Sí	Multicultivo	Pequeños productores rurales
Plant Disease Detector	Diagnóstico por fotos + resultados inmediatos	Hongos, bacterias, virus	Sí	Sí	Papa, tomate, algodón	Agricultores sin conocimientos técnicos

Fuente: Adaptado de [15].

2.2 Fundamentos Teóricos

2.2.1 Inteligencia Artificial y su aplicación en la agricultura

La Inteligencia Artificial (IA) se entiende como una disciplina de la informática enfocada en la creación de sistemas capaces de llevar a cabo actividades que tradicionalmente requieren habilidades cognitivas humanas, como el reconocimiento de patrones, el aprendizaje autónomo y la toma de decisiones, basado en la implementación mediante algoritmos y estructuras matemáticas que capacitan a las máquinas para aprender a partir de grandes volúmenes de datos, el machine learning y el deep learning son algunas de las ramas más relevantes de la actualidad, que han transformado disciplinas como la visión por computador, el procesamiento del lenguaje natural y la automatización de procesos inteligentes.

En el ámbito agrícola, la IA ha mostrado un potencial considerable para la optimización de

procesos y la toma de decisiones basadas en datos precisos, con aplicaciones como la detección automática de enfermedades de los cultivos, la monitorización del estado fisiológico de las plantas, la predicción del rendimiento, la gestión eficiente del riego y el control integrado de plagas, sistemas que funcionan mediante el uso de imágenes, sensores y bases de datos que alimentan modelos de entrenamiento capaces de operar en tiempo real con márgenes de error reducidos, facilitando así una respuesta ágil y fundamentada frente a las variaciones del entorno. Esta integración tecnológica impulsa el desarrollo de prácticas más sostenibles y seguras, permitiendo la adopción de medidas preventivas y eficaces contra las amenazas fitosanitarias [16].

En este estudio se plantea avanzar en la digitalización del sector agrario mediante el diseño de una aplicación móvil potenciada por inteligencia artificial. Dicha herramienta estará enfocada en identificar de forma automática las enfermedades fúngicas que afectan las hojas de los cultivos de papa, empleando algoritmos de visión por computadora para analizar las imágenes que capture el propio usuario. De esta manera, se ofrecerá un diagnóstico rápido y contextualizado a las condiciones de producción en Ecuador, con una solución asequible y autónoma que pueda ser utilizada incluso en zonas con recursos limitados [17].

2.2.2 Visión por computadora en el diagnóstico vegetal

La visión por computador representa una subdisciplina dentro de la inteligencia artificial orientada a dotar a los sistemas de la capacidad de interpretar y analizar imágenes con el fin de extraer información relevante de forma automatizada, siendo una tecnología que ha encontrado aplicaciones exitosas en el ámbito agrícola para tareas como el recuento de frutos, la monitorización del crecimiento de las plantas y, particularmente, en el diagnóstico de enfermedades vegetales, proceso basado en la captura de imágenes de hojas o frutos afectados, el posterior procesamiento mediante algoritmos previamente entrenados y la generación de un diagnóstico sostenible; este procedimiento permite examinar las características visibles de las plantas, reconocer señales asociadas a posibles afecciones y emitir una respuesta diagnóstica sustentada en patrones visuales asociados a manchas, alteraciones cromáticas o texturas irregulares [4].

En el diagnóstico fitosanitario de cultivos, la visión por computador permite la detección de enfermedades foliares a partir de fotografías tomadas directamente en campo sin requerir el uso de equipos especializados, ya que la tecnología se apoya en redes neuronales entrenadas con conjuntos de imágenes clasificadas según el tipo de enfermedad, lo que permite alcanzar altos niveles de precisión en la identificación de síntomas, y en el caso concreto del cultivo de la papa, este tipo de análisis visual automatizado contribuye a una respuesta temprana ante patologías como el tizón tardío, minimizando el margen de error asociado al diagnóstico humano y fortaleciendo la toma de decisiones en la gestión sanitaria de los cultivos [8]. En la Figura 4 se sintetiza el flujo aplicado para entrenar y evaluar modelos de diagnóstico vegetal.

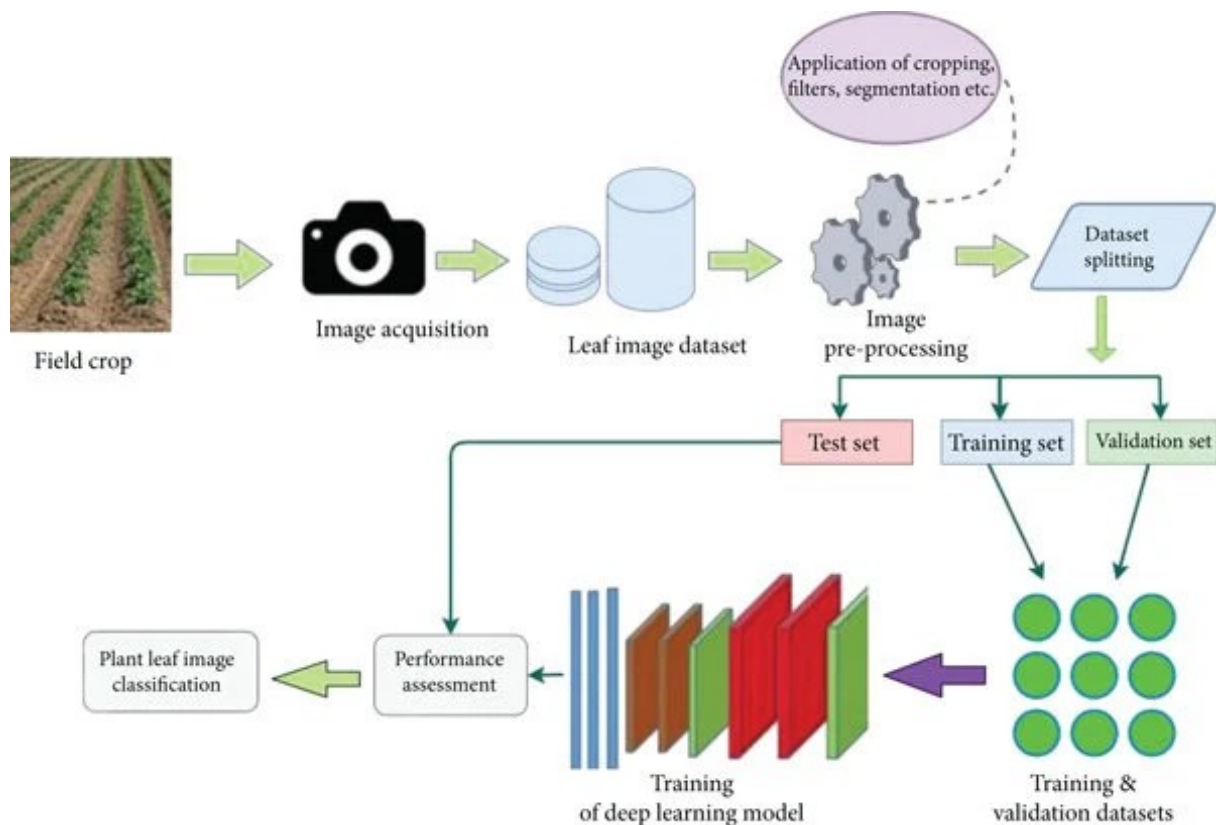


Figura 4.
Procesamiento de imágenes para diagnóstico vegetal
Fuente: Adaptado de [18].

2.2.3 Redes Neuronales Convolucionales (CNN)

En el campo del aprendizaje profundo, las redes neuronales convolucionales (CNN) se han convertido en la opción predilecta para trabajar con datos de tipo imagen. A diferencia de las

redes tradicionales, su diseño incorpora capas de convolución que aplican filtros sobre cada fotograma, lo que les permite reconocer de forma automática rasgos como bordes, texturas o formas sin necesidad de seleccionar manualmente los atributos. Gracias a esta extracción jerárquica de características, las CNN facilitan la identificación de patrones visuales complejos a partir de datos crudos [19].

Una CNN típica consta de un conjunto de capas especializadas que realizan funciones específicas, entre ellas capas convolucionales que aplican filtros para detectar patrones visuales, capas de activación que introducen funciones no lineales, capas de agrupación que reducen la dimensionalidad preservando la información esencial y capas totalmente conectadas que permiten la clasificación o regresión en función del objetivo fijado por el modelo, una estructura que ha demostrado una eficacia extraordinaria en tareas como el reconocimiento facial, la clasificación automática de objetos y, en el ámbito agrícola, la detección precisa de enfermedades que afectan al follaje de las plantas.

En el marco de esta investigación, se utilizan CNNs como base para el desarrollo de un sistema orientado a la identificación de enfermedades presentes en hojas de patata a través del análisis de imágenes digitales capturadas por el usuario, un proceso en el que el modelo es entrenado con un conjunto de imágenes previamente etiquetadas que le permiten aprender a reconocer patrones distintivos asociados a patologías específicas como el tizón tardío y el tizón temprano [20], por lo que su capacidad de generalización y adaptación a condiciones reales convierte a las CNNs en una herramienta robusta y eficiente para el diagnóstico visual automatizado, especialmente en aplicaciones móviles destinadas al uso directo por parte de los agricultores en sus propios cultivos. En la Figura 5 se muestra el flujo de una CNN aplicado a la extracción de características y clasificación de imágenes.

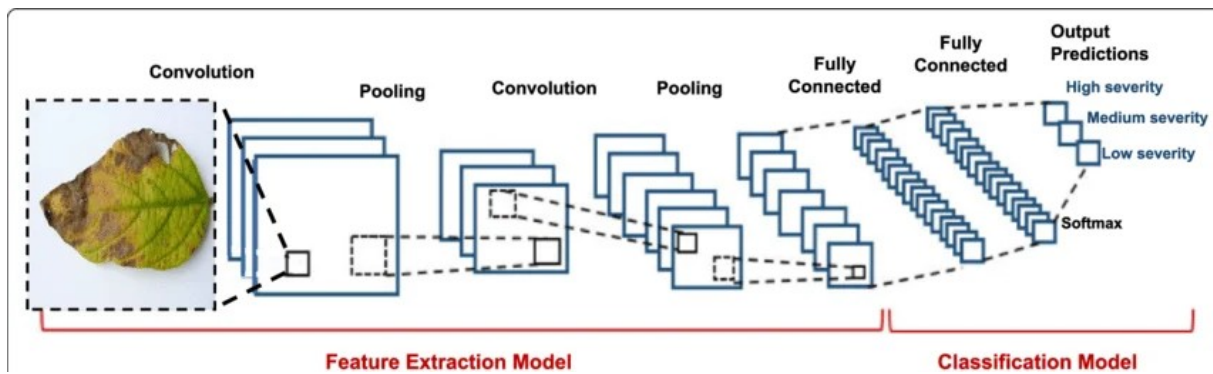


Figura 5.
Arquitectura CNN para clasificación de enfermedades foliares
Fuente: Adaptado de [21].

2.2.4 Modelos de detección en tiempo real (YOLO)

Los modelos de detección en tiempo real representan un componente fundamental en el campo de la visión por computador, ya que permiten no solo identificar sino también localizar múltiples objetos presentes en una imagen con altos niveles de velocidad y precisión, siendo la arquitectura YOLO (You Only Look Once) uno de los enfoques más representativos en este campo debido a que procesa la imagen completa en una sola pasada, eliminando la necesidad de dividir el proceso en etapas separadas de propuesta de regiones y clasificación, lo que se consigue particionando la imagen en una malla donde cada celda es responsable de predecir las cajas delimitadoras junto con las probabilidades correspondientes a cada clase, estrategia que reduce significativamente los tiempos de inferencia y facilita su implementación en dispositivos con recursos limitados como los teléfonos móviles [22].

Dentro del presente trabajo de investigación, se adopta la arquitectura YOLO como base tecnológica para el desarrollo de un sistema orientado a la detección de enfermedades fúngicas en hojas de papa a través del análisis de imágenes capturadas directamente en campo, aprovechando su capacidad de operación en tiempo real y la flexibilidad que ofrece para ser entrenado con conjuntos de datos adaptados a las condiciones específicas del entorno agrícola ecuatoriano [10], mientras que versiones más recientes como YOLOv5 y YOLOv8 han incorporado mejoras sustanciales a través de bloques convolucionales optimizados y una mayor compatibilidad con

hardware móvil, características que consolidan su valor como herramienta práctica y accesible para pequeños y medianos productores agrícolas.

2.2.5 Evaluación del rendimiento de modelos de IA (precisión, recall, F1-score, ISO 25010)

Para conocer la eficacia de un modelo de IA con datos reales, se emplean indicadores de clasificación específicos. En la detección de enfermedades a partir de imágenes, los más habituales son la precisión, la sensibilidad (recall) y la puntuación F1. Estas medidas resultan imprescindibles cuando los conjuntos de datos están desequilibrados, pues no solo evalúan cuán exacto es el modelo al marcar casos positivos, sino también su capacidad para recuperarlos correctamente [23].

Para evaluar el rendimiento del modelo, utilizamos la matriz de confusión, que organiza los resultados en verdaderos y falsos positivos, junto con verdaderos y falsos negativos. La sensibilidad es crucial para la detección de enfermedades foliares porque muestra cuántos casos reales es capaz de identificar el sistema. Sin embargo, mantener un alto nivel de precisión también es fundamental para evitar alarmas innecesarias que podrían impedir el trabajo en los cultivos.

Además de estos indicadores, realizaremos pruebas de campo en parcelas reales, comparando las predicciones automáticas con la inspección visual de expertos. Para ello, aplicaremos los criterios de la norma ISO/IEC 25010 como eficiencia, facilidad de uso, fiabilidad y adaptabilidad con el fin de comprobar no solo la exactitud del modelo, sino también su capacidad de integrarse y funcionar de manera constante en condiciones agrícolas reales.

2.3 Herramientas tecnológicas revisadas en la literatura y su integración al proyecto

2.3.1 YOLOv8: modelo para detección de enfermedades

YOLOv8 es un modelo de Ultralytics pensado para identificar objetos al instante, combinando rapidez y precisión. Su diseño ligero admite distintos formatos de anotación y reduce las

exigencias de hardware, por lo que puede ejecutarse incluso en dispositivos con recursos limitados. Gracias a esta flexibilidad, resulta especialmente útil en aplicaciones móviles orientadas al agro [22].

En la Figura 6 se observa la organización general del modelo YOLOv8.

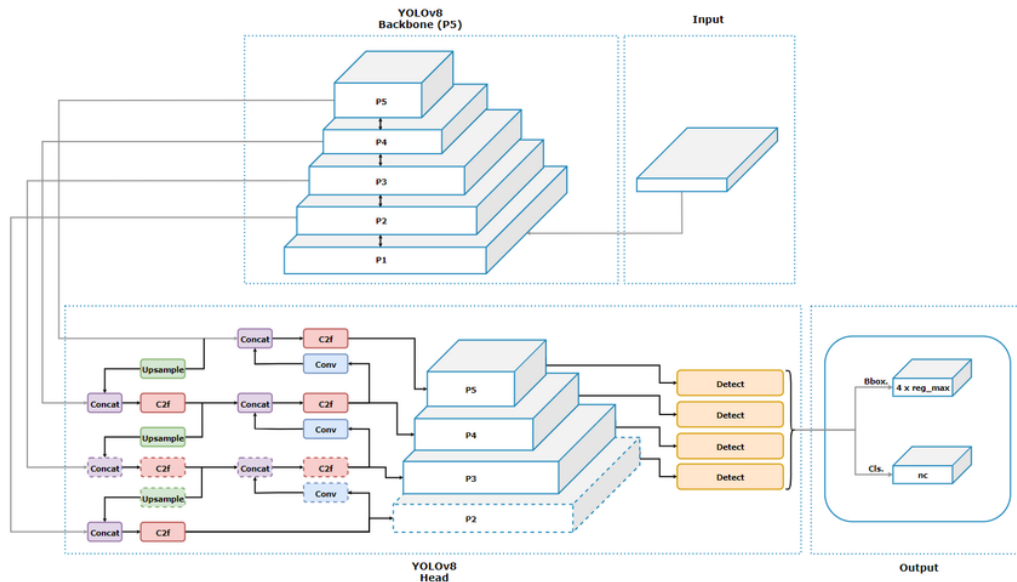


Figura 6. Diseño de los componentes de YOLOv8 para detección en cultivos. Fuente: Adaptado de [24].

En el sector agrícola, YOLOv8 se ha utilizado en estudios relacionados con detección de plagas, clasificación de frutos y reconocimiento de enfermedades en hojas. Estos trabajos reportan un desempeño estable y preciso del modelo, incluso ante variaciones de iluminación o textura en las plantas, por lo que representa una alternativa confiable para el monitoreo automático de cultivos [23].

2.3.2 TensorFlow como entorno para diseñar y ejecutar redes neuronales

Google lanzó TensorFlow como una plataforma de código abierto para agilizar el desarrollo de modelos de IA: agrupa en un solo marco las fases de diseño, entrenamiento y despliegue. Al estructurar los cálculos en grafos de datos, facilita tanto la conexión ordenada de operaciones como la construcción de redes neuronales complejas. Por otro lado, su compatibilidad nativa

con aceleradores de hardware y la extensa colección de librerías y complementos garantizan un rendimiento óptimo, ya sea en potentes servidores o en dispositivos móviles. En la Figura 7 se identifican los recursos que ofrece TensorFlow durante el ciclo de desarrollo de modelos.



Figura 7.
Funcionalidades principales de TensorFlow para modelos de IA
Fuente: Adaptado de [25].

TensorFlow es un entorno adecuado para el desarrollo de aplicaciones que requieren análisis visual en tiempo real. Diversas investigaciones han corroborado su efectividad en la creación de sistemas inteligentes para la detección de enfermedades en cultivos, lo que valida su aplicabilidad en contextos agrícolas que presentan restricciones tecnológicas [26]. En el campo específico de la visión por computador, TensorFlow pone a disposición librerías especializadas como TensorFlow Vision y TensorFlow Lite que permiten el procesamiento eficiente de imágenes, la manipulación de conjuntos de datos visuales y la optimización de modelos para su implementación en dispositivos móviles y entornos con recursos computacionales limitados.

2.3.3 Creación de interfaces móviles con Flutter

Para la capa visual de la aplicación utilizamos Flutter, un kit de desarrollo de software (SDK) de código abierto de Google que permite crear una única base de código y desplegarla tanto en Android como en iOS. Las animaciones fluidas y el rendimiento casi nativo de Flutter son posibles gracias al lenguaje de programación Dart y a su motor de renderizado integrado. Además, su enfoque basado en widgets reutilizables simplifica el desarrollo y mejora la experiencia del usuario en cualquier dispositivo al ensamblar rápida y fácilmente pantallas que se adaptan a varias resoluciones.

En el desarrollo de aplicaciones móviles que incorporan capacidades de visión por computador e inteligencia artificial, Flutter ha demostrado ser una opción eficiente debido a su facilidad de integración con librerías externas como TensorFlow Lite o modelos exportados en formatos compatibles, sumado a la disponibilidad de paquetes como camera, image_picker o flutter_tflite que permiten la captura y procesamiento de imágenes de forma local, así como la visualización de resultados en tiempo real, características que son esenciales en aplicaciones agrícolas orientadas al diagnóstico visual en campo, por lo que Flutter se presenta como una alternativa viable y escalable para el desarrollo de interfaces modernas, intuitivas y adecuadas a las condiciones técnicas de los usuarios en entornos rurales [27].

2.3.4 Google Cloud para almacenamiento y acceso remoto

Google Cloud Platform (GCP) agrupa diversos servicios en la nube que simplifican el despliegue, el almacenamiento y la ampliación de aplicaciones con seguridad y eficiencia. Entre sus componentes más útiles para nuestros fines se encuentran Cloud Storage, Firebase, Cloud Functions y AI Platform, los cuales facilitan tanto el desarrollo del servidor como la integración de soluciones de inteligencia artificial y el tratamiento de datos. Gracias a estas herramientas, es posible crear sistemas distribuidos que funcionan en tiempo real, mantienen alta disponibilidad y resisten fallos todo ello imprescindible para aplicaciones móviles que dependen del acceso

remoto y del procesamiento en la nube [28].

Google Cloud Platform ofrece herramientas útiles para proyectos de agricultura digital que trabajan con análisis de imágenes y diagnóstico de patologías vegetales. Cloud Storage permite guardar colecciones amplias de imágenes, Firebase gestiona el acceso de los usuarios y Firestore mantiene sincronizados los resultados en tiempo real. Por estas características, GCP funciona como un entorno flexible y seguro para aplicaciones móviles en zonas rurales, donde la protección de datos y la accesibilidad son aspectos importantes. Además, los modelos entrenados con TensorFlow pueden ejecutarse en la nube o en dispositivos con conectividad intermitente.

2.3.5 Integración de Scrum y CRISP-DM para proyectos de IA

Para este proyecto hemos combinado la flexibilidad de Scrum con la estructura de CRISP-DM: cada iteración (sprint) nos permite definir objetivos claros, entregar prototipos de la aplicación y recoger comentarios de los usuarios, mientras que CRISP-DM nos guía paso a paso desde la comprensión del problema y los datos hasta la validación y el despliegue del modelo. De esta forma, avanzamos de manera ágil en el desarrollo de la interfaz y, al mismo tiempo, seguimos un proceso metódico para preparar la información, entrenar el modelo, evaluar sus resultados y finalmente ponerlo en producción, asegurando así un flujo de trabajo ordenado y adaptable a las necesidades del proyecto.

Scrum divide el trabajo en ciclos breves que incluyen la definición de objetivos, entregas parciales y retroalimentación continua, lo que facilita mejorar la interfaz móvil y añadir servicios complementarios. Por su parte, CRISP-DM estructura el flujo de trabajo en seis etapas: comprensión del negocio, preparación de datos, modelado, evaluación, validación y despliegue. Al combinar ambos enfoques, garantizamos un proceso que cubre todo el recorrido, desde la recolección de imágenes hasta la comprobación final del sistema [29].

En la Figura 8 se esquematiza la integración metodológica aplicada en el proyecto.

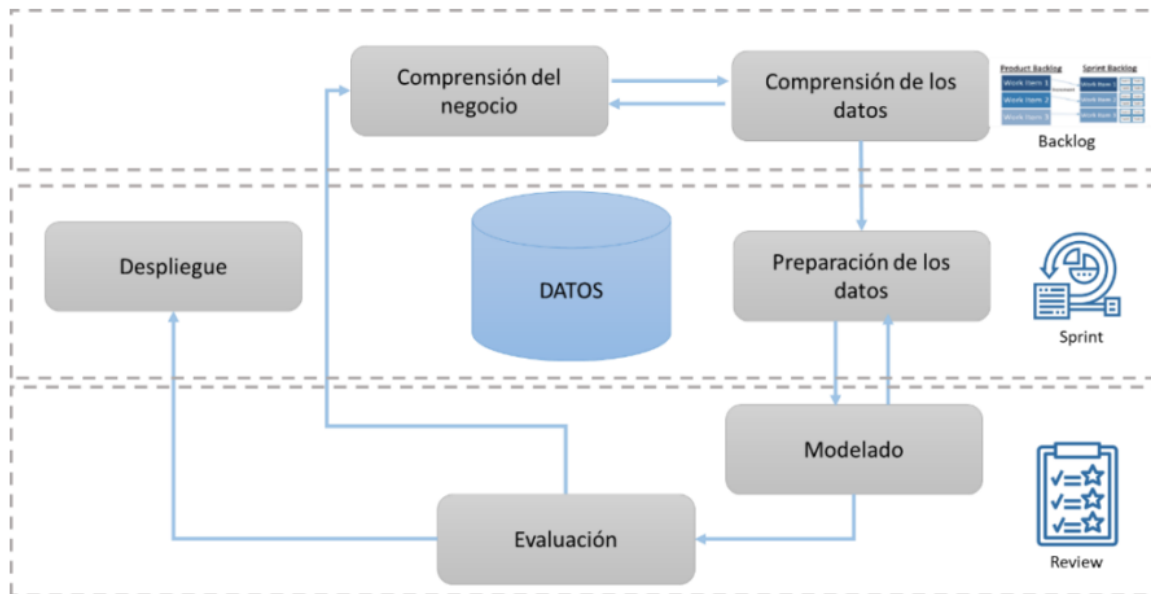


Figura 8.

Relación entre Scrum y CRISP-DM en proyectos de inteligencia artificial

Fuente: Adaptado de [30].

La combinación de estas metodologías asegura un desarrollo integral y controlado en el que cada ciclo de formación, validación y despliegue del modelo de visión por computador se alinea con la entrega progresiva de funcionalidades de la aplicación, configurando un enfoque que favorece no sólo la evolución constante del software, sino también la trazabilidad de las decisiones técnicas, la implementación de mejoras continuas y la capacidad de adaptación a eventuales ajustes requeridos durante la fase de validación en campo.

2.4 Trabajos Relacionados

2.4.1 Revisión de tesis y artículos científicos

En los últimos años, diferentes investigaciones han puesto de relieve el uso de la inteligencia artificial para detectar enfermedades en cultivos de papa mediante el análisis de imágenes de las hojas. Por ejemplo, se han desarrollado aplicaciones móviles que, aun en zonas rurales con conectividad limitada, permiten procesar los datos directamente en el dispositivo, gracias

a modelos ligeros como MobileNet. Estas herramientas incorporan además interfaces sencillas que facilitan su manejo por parte de los agricultores y promueven su adopción en contextos con recursos tecnológicos escasos [10], [31].

Asimismo, la literatura especializada evidencia que las redes neuronales convolucionales profundas siguen siendo la principal opción para clasificar patologías foliares, y que, al combinarse con técnicas de aprendizaje mixto (híbridas), logran precisiones superiores al 95 % en tareas de identificación y categorización de daños [32], [33]. Más allá de reconocer la presencia de la enfermedad, estos enfoques permiten estimar el grado de afectación de la hoja, aportando información valiosa para decisiones fitosanitarias más informadas.

De forma más reciente, se han incorporado al estudio modelos avanzados como los Vision Transformers (ViT) o arquitecturas que mezclan CNN y ViT, con el fin de captar tanto los detalles locales como el contexto global de las imágenes. Gracias a ello, se ha logrado mejorar el rendimiento general de los sistemas de diagnóstico automatizado [34], [35]. Por otra parte, para aumentar la transparencia del proceso, se aplican técnicas de interpretabilidad por ejemplo Grad-CAM, que resaltan las zonas de la imagen que el modelo considera decisivas para su predicción [36].

Además, algunos estudios han demostrado que es posible entrenar modelos robustos incluso con conjuntos de datos limitados siempre que se utilicen imágenes de alta resolución obtenidas directamente en condiciones de campo, un hallazgo relevante para escenarios agrícolas reales en los que la recopilación de grandes volúmenes de datos puede resultar compleja y costosa [37]. En conjunto, estos trabajos evidencian no sólo la viabilidad de aplicar la inteligencia artificial en el sector agrícola, sino también la oportunidad de mejorar las capacidades de diagnóstico mediante aplicaciones móviles especializadas que integren tecnologías de vanguardia y se adapten a las necesidades de los pequeños y medianos productores.

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1 Metodología general del proyecto

El proyecto se desarrolló mediante dos metodologías complementarias: Scrum [38] para la aplicación móvil y CRISP-DM [29] para el modelo de inteligencia artificial. Esta combinación permitió organizar el desarrollo del software y, al mismo tiempo, estructurar el proceso de tratamiento de datos, entrenamiento y evaluación del modelo de detección.

La propuesta trabaja con una aplicación móvil para detectar enfermedades en hojas de papa usando visión por computadora. Es una herramienta funcional. Desde ese enfoque, el estudio tiene carácter aplicado y tecnológico. También mantiene un enfoque cuantitativo y experimental, porque el modelo fue probado con métricas utilizadas en detección de objetos.

En la Tabla 3 se organiza la aplicación de cada metodología dentro de los componentes del proyecto.

Tabla 3.

Metodologías utilizadas en el desarrollo del proyecto

Componente	Aplicación metodológica
Aplicación móvil	Scrum para organizar el prototipo
Modelo de IA	CRISP-DM para datos y evaluación
Integración del sistema	Trabajo combinado entre ambas metodologías

Fuente: Elaboración propia.

3.2 Metodología Scrum para el desarrollo de la aplicación móvil

3.2.1 Aplicación de Scrum en el desarrollo de CultivoScan

Para el desarrollo de la aplicación móvil *CultivoScan* se aplicó Scrum como marco de trabajo ágil, con el fin de organizar las actividades en fases e iteraciones. Esto permitió planificar los requerimientos, priorizar funcionalidades y revisar los avances conforme se implementaban los módulos del sistema.

El desarrollo tomó como referencia las fases Pre-Juego, Juego y Post-Juego propuestas para Scrum [39]. En el Pre-Juego se establecieron los roles del proyecto, los requerimientos, las historias de usuario, el Product Backlog, la arquitectura inicial y el prototipo de interfaces. La fase de Juego concentró la construcción de los módulos mediante sprints, organizando el avance por entregas. El Post-Juego reunió las pruebas funcionales, correcciones finales, publicación en Google Play Store y preparación de la evaluación con usuarios.

3.2.2 Definición del equipo Scrum

Para el desarrollo de la aplicación móvil se definieron los roles principales del equipo Scrum, tomando en cuenta la participación de cada actor en la planificación, revisión e implementación del sistema. Esta asignación permitió organizar las responsabilidades del proyecto y mantener un seguimiento progresivo de los avances durante cada iteración.

En la Tabla 4 se presenta la conformación del equipo Scrum definido para el proyecto.

Tabla 4.

Equipo Scrum del proyecto

Rol	Integrante / actor	Responsabilidad
Product Owner	Ing. Marco Remigio PUSDÁ Chulde, PhD.	Revisar y orientar los avances del producto.
Scrum Master	Diego Ernesto Cuaycal Tirira	Organizar actividades y dar seguimiento al desarrollo.

Rol	Integrante / actor	Responsabilidad
Equipo de desarrollo	Diego Ernesto Cuaycal Tirira	Diseñar, programar e integrar los módulos de la aplicación.
Stakeholders	Ing. Iván Danilo García Santillán, PhD.; usuarios de la aplicación; representante técnico del MAGAP Tulcán	Aportar observaciones y retroalimentación.

Fuente: Elaboración propia.

3.2.3 Fase de Pre-Juego: planificación inicial

Esta fase permitió organizar los elementos base para el desarrollo de la aplicación móvil. En ella se definieron los roles, requerimientos, historias de usuario, Product Backlog, arquitectura inicial y prototipo de interfaces, con el fin de establecer una guía clara antes de iniciar la construcción por sprints.

3.2.3.1 Definición de nomenclatura

La planificación necesitó una forma clara de reconocer cada artefacto del desarrollo. Para eso se usó una nomenclatura específica. Con esta codificación fue más sencillo mantener el orden y seguir la relación entre requerimientos, historias de usuario y sprints.

En la Tabla 5 se detallan los códigos empleados para identificar los artefactos de planificación.

Tabla 5.

Nomenclatura utilizada para la planificación Scrum

Código	Descripción
RF	Requerimiento funcional del sistema
RNF	Requerimiento no funcional del sistema
HU	Historia de usuario
SP	Sprint de desarrollo

Fuente: Elaboración propia.

3.2.3.2 Levantamiento de requerimientos

Los requerimientos se definieron a partir de las necesidades del contexto agrícola, principalmente el diagnóstico visual de enfermedades en hojas de papa y el registro de información de parcelas.

También se incluyeron funcionalidades de apoyo para el seguimiento del cultivo, como calendario agrícola, consulta climática, precio de mercado, historial de diagnósticos y monitoreo de casos críticos.

3.2.3.3 Requerimientos funcionales

Los requerimientos funcionales precisan las acciones que el usuario puede ejecutar dentro de la aplicación móvil. En CultivoScan, ese alcance quedó definido en tareas concretas: autenticación, gestión de parcelas, registro de actividades agrícolas, consulta de información complementaria y diagnóstico de enfermedades mediante imágenes.

En la Tabla 6 se especifican las funciones principales que conforman el alcance operativo de CultivoScan.

Tabla 6.
Requerimientos funcionales

Código	Descripción
RF01	Permitir la visualización de pantallas de bienvenida, registro, inicio de sesión, recuperación de contraseña e ingreso con cuenta de Google.
RF02	Permitir la creación, edición, duplicación, eliminación, búsqueda y visualización de parcelas.
RF03	Registrar actividades agrícolas en un calendario con fecha, prioridad y recordatorios.
RF04	Consultar el pronóstico climático semanal según la ubicación del usuario.
RF05	Consultar el precio de mercado de la papa mediante información actualizada.
RF06	Permitir la captura o selección de imágenes para el diagnóstico con inteligencia artificial.
RF07	Mostrar el resultado del diagnóstico, incluyendo enfermedad detectada, zona afectada y recomendaciones.
RF08	Guardar y consultar el historial de diagnósticos asociados a las parcelas.

Código	Descripción
RF09	Presentar casos críticos en el módulo de monitoreo, asociados a las parcelas y sus recomendaciones.
RF10	Permitir la configuración del usuario, incluyendo edición de perfil, preferencias, notificaciones e información de la aplicación.

Fuente: Elaboración propia.

3.2.3.4 Requerimientos no funcionales

Los requerimientos no funcionales definen las condiciones de calidad que debe cumplir la aplicación móvil, más allá de sus funciones principales. Estos se relacionan con aspectos como usabilidad, seguridad, persistencia de datos, disponibilidad y ejecución local del diagnóstico.

En la Tabla 7 se describen las condiciones de calidad definidas para el funcionamiento de CultivoScan.

Tabla 7.

Requerimientos no funcionales

Código	Descripción
RNF01	La aplicación debe presentar una interfaz intuitiva para usuarios vinculados al sector agrícola.
RNF02	El acceso al sistema debe realizarse mediante autenticación segura con correo electrónico o cuenta de Google.
RNF03	Los datos de usuarios, parcelas y diagnósticos deben almacenarse de forma persistente en Firebase Firestore.
RNF04	El diagnóstico con inteligencia artificial debe ejecutarse localmente en el dispositivo mediante TensorFlow Lite, sin depender de un servidor externo.
RNF05	La aplicación debe consumir servicios externos para consultar clima y precio de mercado de la papa.
RNF06	La aplicación debe estar disponible para dispositivos Android mediante Google Play Store.

Fuente: Elaboración propia.

3.2.3.5 Historias de usuario

Las historias de usuario permitieron describir las funcionalidades de la aplicación desde la perspectiva del usuario final. Cada historia se estructuró considerando el actor involucrado, la

necesidad identificada, la prioridad, la estimación de esfuerzo y los criterios de validación.

En la Tabla 8 se define la estructura utilizada para documentar cada historia de usuario.

Tabla 8.

Formato de historias de usuario

Historia de usuario	
Número	HUXX
Usuario	Actor que utiliza la funcionalidad
Nombre de la historia	Nombre breve de la funcionalidad
Prioridad	Alta / Media / Baja
Estimación	S / M / L
Descripción	Necesidad expresada desde la perspectiva del usuario
Validación	Criterios para comprobar el cumplimiento de la historia

Fuente: Elaboración propia.

Técnica de estimación: Para estimar el esfuerzo de las historias de usuario, se utilizó la técnica *T-Shirt Sizing* [40], la cual permite clasificar la complejidad de una funcionalidad mediante tallas. Esta técnica es útil en etapas iniciales de planificación, ya que no exige definir tiempos exactos desde el inicio, sino una estimación general del esfuerzo requerido.

Para clasificar el esfuerzo de las funcionalidades se trabajó con tallas S, M y L, asociadas a niveles de complejidad baja, media y alta. Esta referencia permitió ordenar las historias de usuario antes de ubicarlas en los sprints correspondientes, según el esfuerzo previsto para su desarrollo.

En la Tabla 9 se establecen los rangos utilizados para clasificar la complejidad de las funcionalidades.

Tabla 9.

Estimación de esfuerzo

Estimación de esfuerzo	Talla
5 – 10 horas	S
10 – 20 horas	M
20 – 30 horas	L

Fuente: Elaboración propia.

A continuación, se muestran las historias de usuario definidas a partir de los requerimientos funcionales de la aplicación móvil. Cada historia recoge una necesidad específica del usuario. También se vincula con una funcionalidad desarrollada durante los sprints.

En la Tabla 10 se describe el acceso inicial del usuario a la aplicación.

Tabla 10.

Historia de usuario HU01

Historia de usuario	
Número: HU01	Usuario: Usuario de la aplicación
Nombre de la historia: Bienvenida, registro e inicio de sesión	
Prioridad: Alta	Estimación: M (10 – 20 horas)
Descripción: Como usuario de la aplicación, necesito visualizar las pantallas de bienvenida, registrarme e iniciar sesión mediante correo electrónico, contraseña o cuenta de Google, para acceder a las funcionalidades principales del sistema.	
Validación:	
<ul style="list-style-type: none"> ▪ Verificar la visualización de las pantallas de bienvenida al ingresar por primera vez. ▪ Comprobar el inicio de sesión mediante correo y cuenta de Google. ▪ Validar la recuperación de contraseña mediante correo electrónico. 	

Fuente: Elaboración propia.

En la Tabla 11 se describe la administración de parcelas dentro de la aplicación.

Tabla 11.

Historia de usuario HU02

Historia de usuario	
Número: HU02	Usuario: Usuario de la aplicación
Nombre de la historia: Gestión de parcelas	
Prioridad: Alta	Estimación: L (20 – 30 horas)
Descripción: Como usuario de la aplicación, necesito registrar y administrar mis parcelas de cultivo para almacenar información agrícola relevante, como nombre, variedad de papa, ubicación, dimensiones, fecha de siembra y notas adicionales.	

Historia de usuario (continuación)

Validación:

- Verificar la creación de una parcela con los campos obligatorios.
 - Comprobar la edición, duplicación y eliminación de parcelas registradas.
 - Validar que la información de cada parcela se almacene correctamente en Firebase Firestore.
 - Confirmar la búsqueda y visualización de parcelas desde la aplicación.
-

Fuente: Elaboración propia.

En la Tabla 12 se describe la planificación de actividades agrícolas asociadas a cada parcela.

Tabla 12.

Historia de usuario HU03

Historia de usuario

Número: HU03

Usuario: Usuario de la aplicación

Nombre de la historia: Registro de actividades por parcela

Prioridad: Media

Estimación: M (10 – 20 horas)

Descripción: Como usuario de la aplicación, necesito registrar actividades dentro de una parcela para organizar tareas relacionadas con el cultivo, asignar una fecha programada y revisar su estado desde una vista de calendario o lista.

Validación:

- Verificar el registro de actividades con título, descripción, fecha programada y prioridad.
 - Comprobar que las actividades se visualicen en la fecha correspondiente dentro del calendario.
 - Validar la visualización de actividades en las secciones pendientes, completadas y todas.
 - Confirmar que el usuario pueda cambiar el estado de una actividad registrada.
-

Fuente: Elaboración propia.

En la Tabla 13 se describe la consulta climática integrada como apoyo a la gestión del cultivo.

Tabla 13.

Historia de usuario HU04

Historia de usuario	
Número: HU04	Usuario: Usuario de la aplicación
Nombre de la historia: Consulta del pronóstico climático	
Prioridad: Media	Estimación: M (10 – 20 horas)
Descripción: Como usuario de la aplicación, necesito consultar el pronóstico climático semanal según mi ubicación, para conocer condiciones del clima, alertas y recomendaciones de apoyo al cultivo.	
Validación:	
<ul style="list-style-type: none">▪ Verificar el uso de la ubicación actual del dispositivo.▪ Comprobar la consulta del pronóstico mediante Open-Meteo.▪ Validar la visualización del pronóstico semanal.▪ Confirmar la presentación de alertas y recomendaciones.	

Fuente: Elaboración propia.

En la Tabla 14 se describe la consulta del valor comercial de la papa desde la aplicación.

Tabla 14.

Historia de usuario HU05

Historia de usuario	
Número: HU05	Usuario: Usuario de la aplicación
Nombre de la historia: Consulta del precio de mercado de la papa	
Prioridad: Media	Estimación: M (10 – 20 horas)
Descripción: Como usuario de la aplicación, necesito consultar el precio actualizado de la papa para conocer su valor referencial en el mercado y contar con información de apoyo para la toma de decisiones relacionadas con la comercialización del cultivo.	
Validación:	
<ul style="list-style-type: none">▪ Verificar que la aplicación consulte el precio de mercado desde el servicio implementado.▪ Comprobar que la información mostrada corresponda al precio actualizado de la papa.▪ Validar que el módulo presente los datos de forma clara para el usuario.▪ Confirmar que el servicio funcione mediante el backend desarrollado con Express.js y Firebase Cloud Functions.	

Fuente: Elaboración propia.

En la Tabla 15 se describe la selección de imágenes para iniciar el proceso de diagnóstico.

Tabla 15.

Historia de usuario HU06

Historia de usuario	
Número: HU06	Usuario: Usuario de la aplicación
Nombre de la historia: Captura y selección de imágenes	
Prioridad: Alta	Estimación: M (10 – 20 horas)
Descripción: Como usuario de la aplicación, necesito tomar una fotografía o seleccionar una imagen desde la galería del dispositivo para utilizarla en el proceso de diagnóstico de enfermedades en hojas de papa.	
Validación:	
<ul style="list-style-type: none">▪ Verificar que el usuario pueda seleccionar la parcela antes de iniciar el diagnóstico.▪ Comprobar el acceso a la cámara del dispositivo para tomar una fotografía.▪ Validar la selección de imágenes desde la galería.▪ Confirmar que la imagen seleccionada pueda visualizarse antes de continuar con el análisis.	

Fuente: Elaboración propia.

En la Tabla 16 se describe el análisis inteligente de imágenes para detectar enfermedades en hojas de papa.

Tabla 16.

Historia de usuario HU07

Historia de usuario	
Número: HU07	Usuario: Usuario de la aplicación
Nombre de la historia: Diagnóstico con inteligencia artificial	
Prioridad: Alta	Estimación: L (20 – 30 horas)
Descripción: Como usuario de la aplicación, necesito analizar una imagen de una hoja de papa para identificar posibles enfermedades mediante el modelo de inteligencia artificial integrado en el dispositivo.	
Validación:	
<ul style="list-style-type: none">▪ Verificar que el modelo TensorFlow Lite procese la imagen localmente.▪ Comprobar la detección de la enfermedad en la hoja analizada.▪ Validar que se muestre la zona afectada en la imagen.▪ Confirmar la presentación de recomendaciones según el resultado obtenido.	

Fuente: Elaboración propia.

En la Tabla 17 se describe el seguimiento de diagnósticos realizados en las parcelas registradas.

Tabla 17.

Historia de usuario HU08

Historia de usuario	
Número: HU08	Usuario: Usuario de la aplicación
Nombre de la historia: Historial de diagnósticos	
Prioridad: Alta	Estimación: M (10 – 20 horas)
Descripción: Como usuario de la aplicación, necesito consultar los diagnósticos realizados para revisar los resultados obtenidos y dar seguimiento al estado de mis parcelas.	
Validación:	
<ul style="list-style-type: none">▪ Verificar que los diagnósticos se guarden asociados a la parcela correspondiente.▪ Comprobar la visualización del historial de diagnósticos realizados.▪ Validar la clasificación de registros sanos, enfermos y recientes.▪ Confirmar que cada diagnóstico muestre la información principal del resultado.	

Fuente: Elaboración propia.

En la Tabla 18 se describe la revisión de casos que requieren atención prioritaria en las parcelas.

Tabla 18.

Historia de usuario HU09

Historia de usuario	
Número: HU09	Usuario: Usuario de la aplicación
Nombre de la historia: Monitoreo de casos críticos	
Prioridad: Media	Estimación: M (10 – 20 horas)
Descripción: Como usuario de la aplicación, necesito visualizar los diagnósticos considerados críticos para identificar parcelas que requieren atención y revisar las recomendaciones asociadas.	

Validación:

- Verificar que el módulo muestre los diagnósticos con mayor nivel de riesgo.
 - Comprobar que cada caso crítico se relacione con su parcela correspondiente.
 - Validar la visualización de recomendaciones según el diagnóstico obtenido.
 - Confirmar que la información del monitoreo se actualice con los diagnósticos registrados.
-

Fuente: Elaboración propia.

En la Tabla 19 se describe la gestión de opciones generales del perfil y la aplicación.

Tabla 19.

Historia de usuario HU10

Historia de usuario	
Número: HU10	Usuario: Usuario de la aplicación
Nombre de la historia: Configuración de usuario	
Prioridad: Baja	Estimación: S (5 – 10 horas)
Descripción: Como usuario de la aplicación, necesito acceder a las opciones de configuración para editar mi perfil, revisar información de la aplicación y gestionar preferencias generales.	
Validación:	
<ul style="list-style-type: none">▪ Verificar la edición del nombre del usuario.▪ Comprobar el acceso a notificaciones, políticas de privacidad y condiciones de uso.▪ Validar la visualización de información de la aplicación y colaboradores.▪ Confirmar el acceso a opciones de contacto, calificación y opción para compartir la aplicación.	

Fuente: Elaboración propia.

3.2.3.6 Diagrama general de casos de uso

Con base en los requerimientos funcionales y las historias de usuario definidas, se elaboró el diagrama general de casos de uso de la aplicación móvil CultivoScan. Su incorporación permite representar el alcance funcional del sistema desde una perspectiva general, mostrando

la interacción principal entre el usuario y las funcionalidades contempladas durante la fase de planificación.

Con esta representación se consolida una visión inicial del comportamiento funcional de la aplicación, vinculando los requerimientos definidos con las historias de usuario y la posterior organización del Product Backlog. La relación mantiene la coherencia. Conecta la planificación del sistema con las funcionalidades que posteriormente fueron desarrolladas durante los sprints.

En la Figura 9 se visualiza la interacción general entre el usuario y los módulos del sistema.



Figura 9.
Diagrama general de casos de uso de la aplicación móvil CultivoScan
Fuente: Elaboración propia.

El diagrama sintetiza el alcance funcional definido para CultivoScan y respalda la organización de las funcionalidades priorizadas en la fase de planificación.

3.2.3.7 Product Backlog

Las historias de usuario definidas previamente fueron organizadas según su prioridad, estimación y sprint de desarrollo. Esta organización permitió establecer una guía de trabajo para la construcción incremental de los módulos de la aplicación móvil, considerando primero las funcionalidades esenciales para el funcionamiento del sistema.

Este listado también facilitó la relación entre los requerimientos funcionales, las historias de usuario y las actividades planificadas para cada iteración. El orden quedó claro. Permitted identificar qué funcionalidades debían implementarse en cada sprint y cómo aportaban al cumplimiento de los objetivos de la aplicación.

En la Tabla 20 se organiza la planificación inicial de las historias de usuario por sprint.

Tabla 20.
Product Backlog de la aplicación móvil

ID	Historia de usuario	Prioridad	Estimación	Sprint
HU01	Bienvenida, registro e inicio de sesión	Alta	M	SP01
HU02	Gestión de parcelas	Alta	L	SP02
HU03	Registro de actividades por parcela	Media	M	SP03
HU04	Consulta del pronóstico climático	Media	M	SP04
HU05	Consulta del precio de mercado de la papa	Media	M	SP04
HU06	Captura y selección de imágenes	Alta	M	SP05
HU07	Diagnóstico con inteligencia artificial	Alta	L	SP05
HU08	Historial de diagnósticos	Alta	M	SP06
HU09	Monitoreo de casos críticos	Media	M	SP06
HU10	Configuración de usuario	Baja	S	SP06

Fuente: Elaboración propia.

3.2.3.8 Prototipado de interfaces en Figma

Antes de iniciar la implementación de la aplicación móvil, se elaboró un prototipo de interfaces en Figma [41] con el propósito de definir la estructura visual, la distribución de los componentes y el flujo de navegación del sistema. Esta etapa permitió organizar de manera previa las pantallas principales y establecer una guía visual para el posterior desarrollo en Flutter.

El prototipo permitió representar las funcionalidades principales de la aplicación, como el acceso de usuarios, la gestión de parcelas, la captura de imágenes, la visualización del diagnóstico y la configuración general del sistema. Además, facilitó la revisión temprana del diseño, permitiendo identificar ajustes necesarios antes de avanzar con la programación de las interfaces.

Las pantallas diseñadas incluyeron la pantalla de inicio, registro, acceso, recuperación de contraseña, pantalla principal, perfil, captura de imagen, resultado del diagnóstico, historial, gestión de parcelas y configuración. Durante el desarrollo, algunas interfaces fueron modificadas de acuerdo con las necesidades funcionales identificadas en la implementación, por lo que el prototipo sirvió como una base inicial y no como una versión definitiva del diseño; en la Figura 10 se presenta la base utilizada para orientar el diseño inicial de la aplicación.

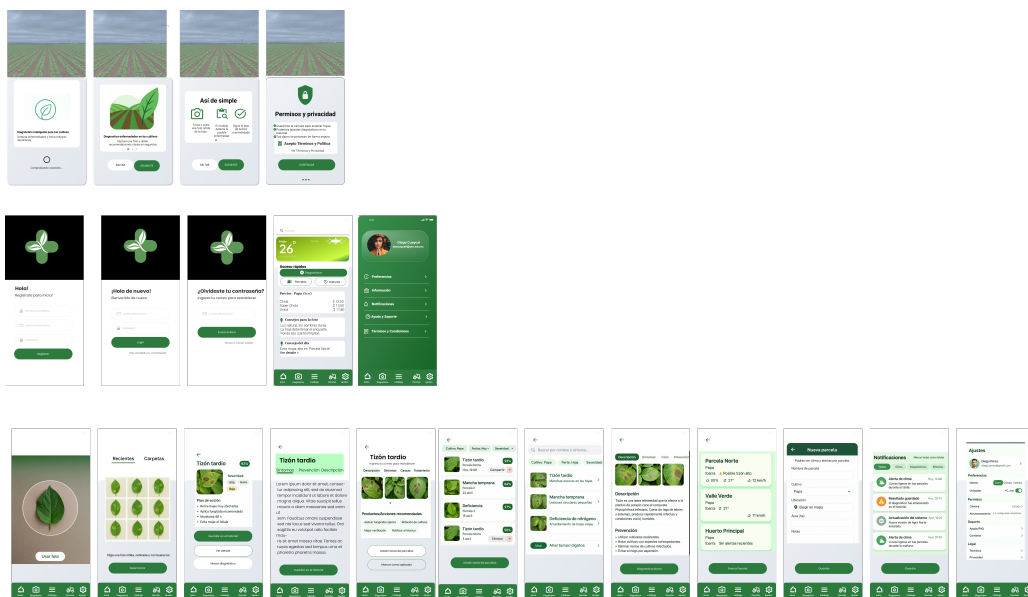


Figura 10.
Prototipo general de interfaces diseñado en Figma

Fuente: Elaboración propia.

3.2.3.9 Arquitectura de la aplicación móvil

La arquitectura de la aplicación móvil se estructuró por capas, con el propósito de separar la interfaz de usuario, la gestión de estado, la lógica de dominio, el acceso a datos y el procesamiento local del modelo de inteligencia artificial. Esta organización permitió mantener una estructura modular y facilitar la integración de los servicios utilizados por la aplicación.

En la capa de presentación se ubicaron los módulos principales desarrollados en Flutter [27], como autenticación, pantalla principal, diagnóstico, parcelas, alertas y configuración. Para la gestión de estado se empleó Riverpod, permitiendo controlar la comunicación entre las interfaces y los casos de uso de la aplicación.

La capa de datos integró los servicios necesarios para el funcionamiento del sistema, entre ellos Firebase Authentication para el acceso de usuarios, Firebase Firestore para el almacenamiento de información [42], Firebase Cloud Functions con Express.js para la consulta del precio de la papa y la API externa Open-Meteo para el pronóstico climático. Además, el diagnóstico con inteligencia artificial fue incorporado directamente en la aplicación mediante TensorFlow Lite [27], lo que permite ejecutar el modelo en el dispositivo sin depender de un servidor externo para el procesamiento de imágenes.

En la Figura 11 se presenta la arquitectura por capas de CultivoScan, integrando la interfaz móvil, la gestión de estado, el dominio, los servicios de datos, el motor de IA local y los servicios externos.

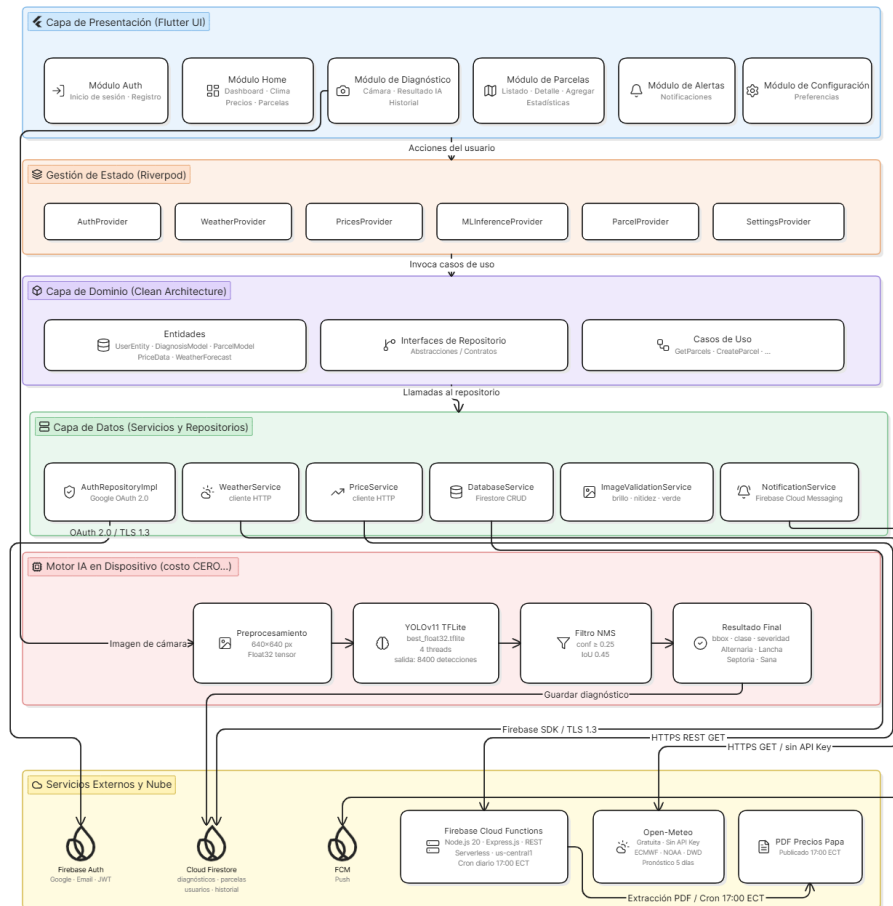


Figura 11.
Arquitectura de la aplicación móvil
Fuente: Elaboración propia.

3.2.3.10 Estructura general del proyecto

La aplicación móvil se organizó mediante una estructura modular, con el propósito de separar las funcionalidades principales y facilitar el mantenimiento del código. El código fuente principal se ubicó dentro del directorio *lib*, donde se organizaron los módulos de navegación, servicios compartidos, autenticación, inicio, captura de imágenes, diagnóstico con inteligencia artificial, gestión de parcelas, alertas, configuración y tutorial inicial.

El sistema incluyó varios recursos de apoyo para poder ejecutarse correctamente. Entre ellos estuvieron el modelo de inteligencia artificial en formato TensorFlow Lite, las reglas de se-

guridad de Firestore, las dependencias registradas en pubspec.yaml y el backend usado para consultar el precio de la papa. Con esta organización se mantuvo separada la aplicación móvil de los servicios externos y de los recursos utilizados durante la ejecución.

En la Figura 12 se observa la organización modular del código fuente y los recursos del proyecto.

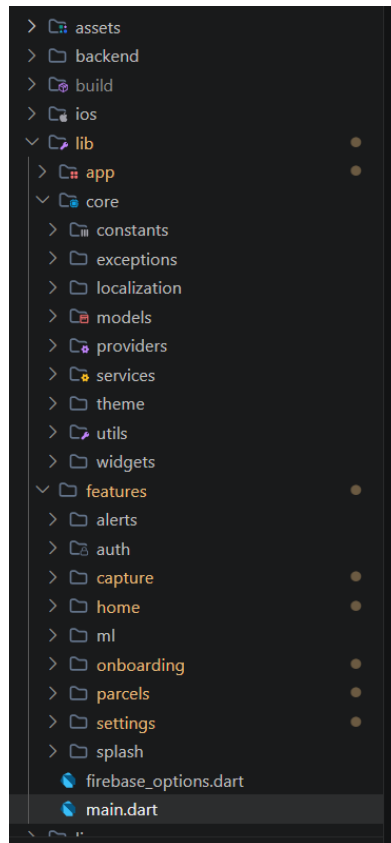


Figura 12.
Estructura general del proyecto Flutter

Fuente: Elaboración propia.

3.2.4 Fase de Juego: desarrollo por sprints

En esta fase se realizó la implementación de la aplicación móvil mediante iteraciones de desarrollo. Cada sprint agrupó un conjunto de historias de usuario relacionadas con funcionalidades específicas, permitiendo construir el sistema de manera progresiva.

En cada iteración se trabajó con un objetivo de sprint, actividades previstas, incremento cons-

truido y revisión de criterios de aceptación. Así se controló el avance de la aplicación. También fue posible ajustar partes del desarrollo a medida que se terminaban los módulos principales.

3.2.4.1 Planificación general de sprints

La planificación general permitió distribuir las historias de usuario en diferentes sprints, considerando la prioridad, complejidad y dependencia entre funcionalidades. Esta organización facilitó el desarrollo incremental de la aplicación móvil.

En la Tabla 21 se presenta la distribución del trabajo por sprint y su relación con las historias de usuario.

Tabla 21.
Distribución general de sprints del proyecto

Sprint	Trabajo planificado	Historias vinculadas
SP00	Preparación del entorno y base arquitectónica	Configuración inicial, estructura del proyecto y servicios principales
SP01	Pantallas de bienvenida, autenticación y acceso	HU01
SP02	Administración de parcelas agrícolas	HU02
SP03	Registro de actividades por parcela	HU03
SP04	Consulta de información agrícola	HU04, HU05
SP05	Captura de imágenes y diagnóstico con IA	HU06, HU07
SP06	Historial, monitoreo y opciones de configuración	HU08, HU09, HU10

Fuente: Elaboración propia.

Para asegurar la calidad de los incrementos desarrollados durante cada sprint, se definió un criterio de terminado aplicable a las historias de usuario implementadas. La verificación fue necesaria. Esta definición permitió comprobar que cada funcionalidad cumpliera criterios mínimos antes de considerarse finalizada dentro del proceso Scrum.

En la Tabla 22 se establecen los criterios para validar la finalización de cada incremento.

Tabla 22.

Definición de terminado aplicada durante los sprints

Criterio	Descripción
Funcionalidad implementada	La historia de usuario fue desarrollada conforme al alcance definido para el sprint correspondiente.
Validación de datos	Los formularios, entradas o acciones del usuario fueron revisados para evitar registros incompletos o inconsistentes.
Persistencia y procesamiento correcto	La información se guardó, consultó o procesó de acuerdo con la funcionalidad implementada.
Navegación e interfaz revisadas	La pantalla conservó el diseño general de la aplicación y permitió desplazarse sin dificultad entre sus opciones.
Criterios de aceptación verificados	La funcionalidad fue contrastada con los criterios definidos en la historia de usuario correspondiente.

Fuente: Elaboración propia.

3.2.4.2 Sprint 0: configuración del entorno y servicios base

Esta iteración permitió preparar las condiciones técnicas necesarias para iniciar el desarrollo de la aplicación móvil. Durante esta etapa se configuró el proyecto en Flutter, se añadieron las dependencias principales y se estableció la conexión con los servicios de Firebase utilizados en la aplicación.

También se configuraron los servicios base para la autenticación, el almacenamiento de datos, el consumo de información externa y la organización inicial del código. Además, se prepararon los recursos necesarios para integrar posteriormente el modelo de inteligencia artificial en formato *TensorFlow Lite*.

En la Tabla 23 se detallan las actividades para preparar el entorno y los servicios base.

Tabla 23.

Sprint Backlog – Sprint 0

Actividad	Descripción
Configuración del entorno	Creación del proyecto Flutter y configuración inicial para dispositivos Android.
Dependencias del proyecto	Incorporación de paquetes necesarios para Firebase, Riverpod, GoRouter, consumo HTTP y TensorFlow Lite.

Actividad	Descripción
Servicios Firebase	Configuración de Firebase Authentication, Firestore y reglas iniciales de seguridad.
Servicios complementarios	Preparación de la conexión con Open-Meteo y el backend de precios mediante Firebase Cloud Functions.
Recursos del modelo IA	Organización del directorio para almacenar el modelo <i>.tflite</i> dentro de los recursos de la aplicación.

Fuente: Elaboración propia.

3.2.4.3 Sprint 1: Bienvenida, autenticación y acceso de usuarios

La primera iteración funcional estuvo orientada al desarrollo del flujo inicial de acceso al sistema. En esta etapa se implementaron las pantallas de bienvenida, registro de cuenta, inicio de sesión, recuperación de contraseña y acceso mediante cuenta de Google.

Para este módulo se utilizó Firebase Authentication, lo que permitió gestionar la autenticación de usuarios mediante correo electrónico y contraseña, además del acceso con Google. Esta funcionalidad permitió proteger las funciones principales de la aplicación y relacionar la información registrada con cada cuenta de usuario.

En la Tabla 24 se detalla la actividad para implementar el acceso inicial al sistema.

Tabla 24.
Sprint Backlog – Sprint 1

Historia	Nombre	Actividad desarrollada
HU01	Bienvenida, registro e inicio de sesión	Implementación de pantallas de bienvenida, registro, inicio de sesión, recuperación de contraseña y acceso con cuenta de Google.

Fuente: Elaboración propia.

Incremento desarrollado: Como resultado de esta iteración se obtuvo el flujo inicial de acceso a la aplicación. El usuario puede visualizar las pantallas de bienvenida, crear una cuenta, iniciar sesión mediante correo electrónico y contraseña, recuperar el acceso mediante el correo registrado e ingresar con una cuenta de Google. Además, se incorporaron validaciones en los campos de registro e inicio de sesión.

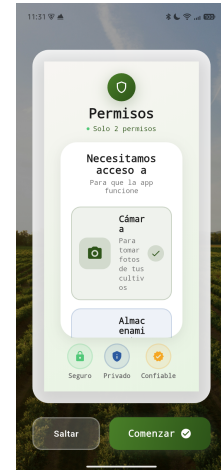
Como parte del flujo inicial, se implementaron pantallas de bienvenida para presentar al usuario el propósito general del sistema antes de acceder a las funciones principales. Estas pantallas se muestran únicamente durante el primer uso de la aplicación, como se observa en la Figura 13.



(a) Bienvenida inicial



(b) Descripción del sistema



(c) Acceso a la aplicación

Figura 13.

Pantallas de bienvenida de la aplicación móvil

Fuente: Elaboración propia.

Posteriormente, se desarrollaron las interfaces de autenticación, las cuales permiten registrar una cuenta, iniciar sesión, recuperar la contraseña y acceder mediante una cuenta de Google. Estas pantallas forman parte del flujo de ingreso al sistema.

En la Figura 14 se presenta el flujo de autenticación implementado para el acceso al sistema.



(a) Inicio de sesión

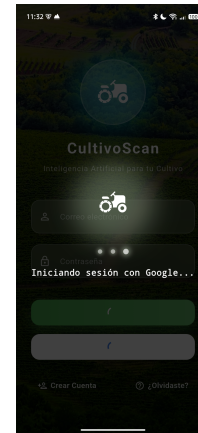


(b) Registro de usuario

Parte 1: Pantallas de acceso



(c) Recuperación de contraseña



(d) Acceso con cuenta de Google

Figura 14.

Pantallas de autenticación de la aplicación móvil

Fuente: Elaboración propia.

Revisión del sprint: Se verificó que el usuario pudiera visualizar las pantallas de bienvenida, registrarse, iniciar sesión, recuperar su contraseña y acceder mediante una cuenta de Google. También se comprobó que, después de una autenticación correcta, el usuario pueda ingresar a la pantalla principal.

En la Tabla 25 se verifica el cumplimiento de los criterios definidos para el acceso al sistema.

Tabla 25.

Revisión de criterios de aceptación – Sprint 1

Criterio de aceptación	Cumple
El usuario puede visualizar las pantallas de bienvenida iniciales.	Sí
El usuario puede registrarse con nombre, correo y contraseña.	Sí
El usuario puede iniciar sesión con correo electrónico y contraseña.	Sí
El usuario puede acceder mediante una cuenta de Google.	Sí
El usuario puede recuperar su contraseña mediante correo electrónico.	Sí
Los campos obligatorios son validados antes de continuar.	Sí

Fuente: Elaboración propia.

3.2.4.4 Sprint 2: Gestión de parcelas

La segunda iteración funcional estuvo orientada al desarrollo del módulo de parcelas. En esta etapa se implementaron las funciones necesarias para registrar, consultar y administrar la información de las parcelas dentro de la aplicación móvil.

Este módulo permitió al usuario organizar los datos principales de cada parcela, como nombre, variedad de papa, ubicación, dimensiones, fecha de siembra y notas adicionales. Además, cada registro quedó preparado para asociarse con los diagnósticos realizados posteriormente, facilitando el seguimiento individual de cada cultivo.

En la Tabla 26 se detalla la actividad para implementar la gestión de parcelas.

Tabla 26.

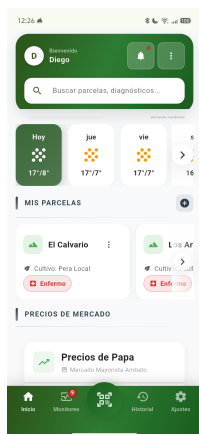
Sprint Backlog – Sprint 2

Historia	Nombre	Actividad desarrollada
HU02	Gestión de parcelas	Implementación del módulo de parcelas, incluyendo registro, consulta, actualización, duplicación, eliminación y búsqueda de información.

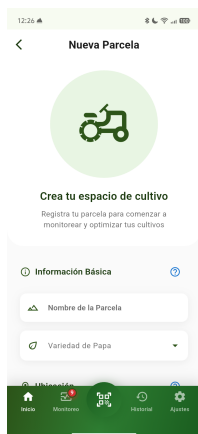
Fuente: Elaboración propia.

Incremento desarrollado: Como resultado de esta iteración se obtuvo el módulo de parcelas funcional. Se permitió al usuario registrar nuevos cultivos, revisar la información de cada parcela, actualizar datos existentes, duplicar registros, eliminar información y realizar búsquedas dentro del listado. También se incorporó una vista de detalle con información general, ubicación, estadísticas y diagnósticos recientes asociados.

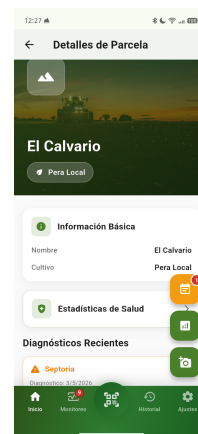
Como evidencia del módulo desarrollado, se presentan las pantallas principales de gestión de parcelas. Estas interfaces muestran el listado de registros, el formulario de creación, la vista de detalle y las acciones disponibles para editar, duplicar o eliminar una parcela; en la Figura 15 se presentan las interfaces implementadas para administrar parcelas dentro de la aplicación.



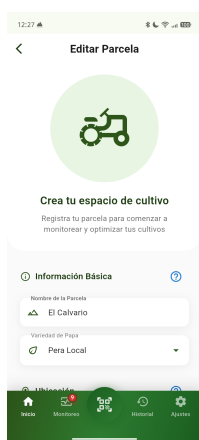
(a) Listado de parcelas



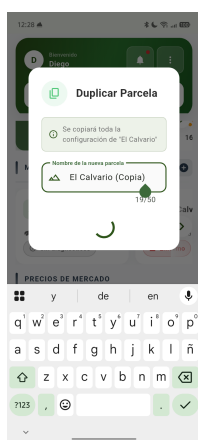
(b) Registro de parcela



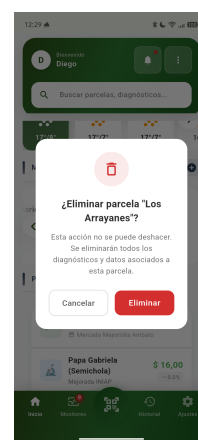
(c) Detalle de parcela



(d) Edición de parcela



(e) Duplicación de parcela



(f) Eliminación de parcela

Figura 15.

Pantallas del módulo de gestión de parcelas

Fuente: Elaboración propia.

Revisión del sprint: Se verificó que el usuario pudiera crear, consultar, editar, duplicar, eliminar y buscar parcelas dentro de la aplicación. También se comprobó que la información registrada se almacenara correctamente y pudiera visualizarse desde la vista de detalle.

En la Tabla 27 se verifica el cumplimiento de los criterios para el módulo de parcelas.

Tabla 27.

Revisión de criterios de aceptación – Sprint 2

Criterio de aceptación	Cumple
El usuario puede crear una nueva parcela con información básica del cultivo.	Sí
El usuario puede visualizar el listado de parcelas registradas.	Sí

Criterio de aceptación	Cumple
El usuario puede consultar el detalle de una parcela seleccionada.	Sí
El usuario puede editar la información de una parcela existente.	Sí
El usuario puede duplicar una parcela registrada.	Sí
El usuario puede eliminar una parcela cuando sea necesario.	Sí
El usuario puede buscar parcelas dentro del listado.	Sí

Fuente: Elaboración propia.

3.2.4.5 Sprint 3: Registro de actividades por parcela

La tercera iteración funcional se orientó al desarrollo del módulo de actividades asociadas a cada parcela. En esta etapa se implementó una vista de calendario que permite planificar tareas relacionadas con el manejo del cultivo y llevar un control de las actividades registradas.

Este módulo permitió crear actividades con información como tipo de tarea, título, descripción, fecha programada y prioridad. También se incorporaron secciones para consultar actividades pendientes, completadas y generales, facilitando el seguimiento de las labores registradas dentro de cada parcela.

En la Tabla 28 se detalla la actividad para implementar el registro de actividades por parcela.

Tabla 28.

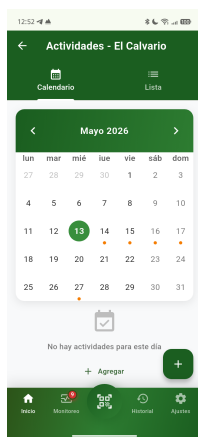
Sprint Backlog – Sprint 3

Historia	Nombre	Actividad desarrollada
HU03	Registro de actividades por parcela	Implementación del módulo de actividades, incluyendo calendario, registro de tareas, clasificación por estado y gestión de prioridades.

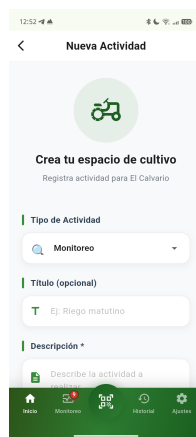
Fuente: Elaboración propia.

Incremento desarrollado: Como resultado de esta iteración se obtuvo el módulo de actividades por parcela. Se permitió al usuario registrar tareas, asignar una fecha programada, establecer una prioridad y consultar las actividades desde una vista de calendario. Además, se incorporó la visualización de actividades pendientes, completadas y generales, junto con una vista de detalle para revisar la información de cada actividad registrada.

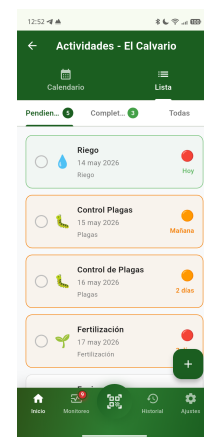
Como evidencia del módulo desarrollado, se presentan las pantallas principales del registro de actividades por parcela. Estas interfaces muestran el calendario general, el formulario de creación, las secciones de actividades pendientes, completadas y generales, así como el detalle de una actividad seleccionada; en la Figura 16 se presentan las interfaces para planificar actividades por parcela.



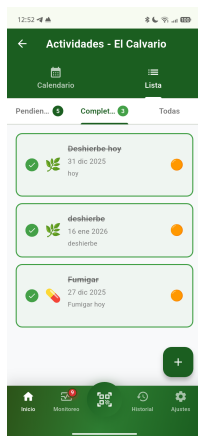
(a) Calendario general



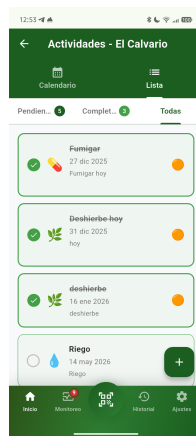
(b) Registro de actividad



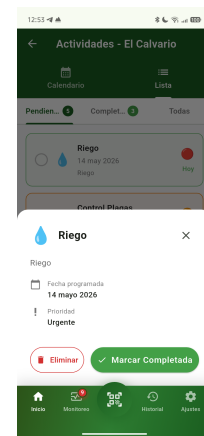
(c) Actividades pendientes



(d) Actividades completadas



(e) Todas las actividades



(f) Detalle de actividad

Figura 16.

Pantallas del módulo de actividades por parcela

Fuente: Elaboración propia.

Revisión del sprint: Se verificó que el usuario pudiera registrar actividades dentro de una parcela, asignar fecha y prioridad, visualizar las tareas en el calendario y consultar su estado. También se verificó la clasificación de actividades pendientes, completadas y generales. Se revisaron las acciones disponibles desde el detalle de cada actividad.

En la Tabla 29 se verifica el cumplimiento de los criterios para el módulo de actividades.

Tabla 29.

Revisión de criterios de aceptación – Sprint 3

Criterio de aceptación	Cumple
El usuario puede registrar actividades dentro de una parcela.	Sí
El usuario puede ingresar título, descripción, fecha programada y prioridad.	Sí
El usuario puede seleccionar el tipo de actividad agrícola.	Sí
Las actividades se visualizan en la fecha correspondiente dentro del calendario.	Sí
El usuario puede consultar actividades pendientes, completadas y generales.	Sí
El usuario puede revisar el detalle de una actividad registrada.	Sí
El usuario puede cambiar el estado o eliminar una actividad registrada.	Sí

Fuente: Elaboración propia.

3.2.4.6 Sprint 4: Servicios de información agrícola

La cuarta iteración funcional se orientó a la integración de servicios informativos para apoyar el seguimiento del cultivo. En esta etapa se implementaron los módulos de pronóstico climático y consulta del precio de mercado de la papa.

El pronóstico climático permite visualizar información semanal de acuerdo con la ubicación de la parcela seleccionada, incluyendo el detalle diario de temperatura, amplitud térmica y recomendaciones agrícolas. Por otra parte, el módulo de precios permite consultar valores referenciales de distintas variedades de papa, tomando como fuente la información del Mercado Mayorista de Ambato.

En la Tabla 30 se detallan las actividades para integrar los servicios de clima y precios.

Tabla 30.

Sprint Backlog – Sprint 4

Historia	Nombre	Actividad desarrollada
HU04	Consulta del pronóstico climático	Integración del servicio climático para mostrar el pronóstico semanal, detalle diario y recomendaciones agrícolas.
HU05	Consulta del precio de mercado de la papa	Implementación del módulo de precios para visualizar valores actualizados de variedades de papa.

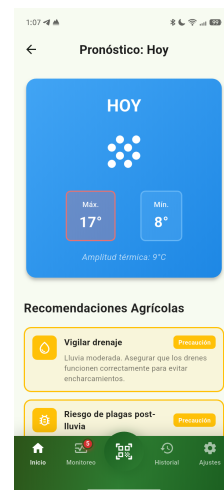
Fuente: Elaboración propia.

Incremento desarrollado: Como resultado de esta iteración se incorporaron servicios de información agrícola dentro de la aplicación. Se permitió al usuario consultar el pronóstico climático desde la pantalla principal, revisar el detalle del día seleccionado y visualizar recomendaciones relacionadas con el manejo del cultivo. Además, se integró la consulta del precio de mercado de la papa como información complementaria para la toma de decisiones.

Como evidencia del módulo climático, se presentan las pantallas correspondientes a la consulta del pronóstico semanal y al detalle diario de las condiciones meteorológicas. Esta funcionalidad permite al usuario revisar información climática asociada al seguimiento del cultivo y visualizar recomendaciones agrícolas dentro de la aplicación; en la Figura 17 se presentan las interfaces para consultar el clima y sus recomendaciones.



(a) Pronóstico semanal

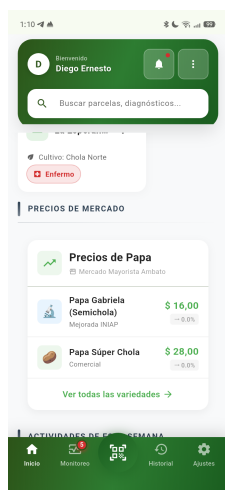


(b) Detalle y recomendaciones

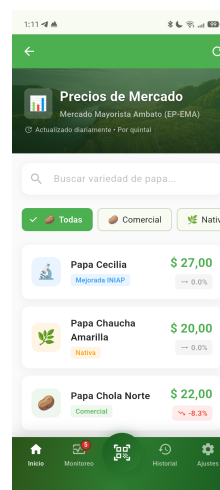
Figura 17.
Pantallas del módulo de pronóstico climático

Fuente: Elaboración propia.

Como evidencia del módulo de precios, se presentan las pantallas correspondientes a la consulta del valor referencial de la papa. Esta funcionalidad permite visualizar información del Mercado Mayorista de Ambato, incluyendo variedades, categorías y precios por quintal dentro de la aplicación; en la Figura 18 se muestran las interfaces para consultar valores referenciales de la papa.



(a) Resumen de precios



(b) Listado de variedades

Figura 18.

Pantallas del módulo de precio de mercado de la papa

Fuente: Elaboración propia.

Revisión del sprint: Se verificó que el usuario pudiera consultar el pronóstico climático asociado a una parcela y revisar el detalle diario con recomendaciones agrícolas. Además, se comprobó la visualización del precio de mercado de la papa mediante el servicio integrado en la aplicación; en la Tabla 31 se verifica el cumplimiento de los criterios para los servicios agrícolas.

Tabla 31.

Revisión de criterios de aceptación – Sprint 4

Criterio de aceptación	Cumple
El usuario puede seleccionar una parcela para consultar información climática.	Sí
La aplicación muestra el pronóstico climático semanal.	Sí
El usuario puede revisar el detalle climático de un día seleccionado.	Sí
La aplicación presenta recomendaciones agrícolas asociadas al clima.	Sí
El usuario puede consultar el precio de mercado de la papa.	Sí
La aplicación muestra variedades, categorías y precios referenciales de papa.	Sí

Fuente: Elaboración propia.

3.2.4.7 Sprint 5: Captura de imágenes y diagnóstico con inteligencia artificial

La quinta iteración funcional se orientó al desarrollo del flujo de diagnóstico mediante imágenes. En esta etapa se implementó la selección de parcela, la captura o carga de fotografías, la

validación de la imagen y la visualización del resultado generado por el modelo de inteligencia artificial.

El modelo TensorFlow Lite permite realizar el diagnóstico desde la propia aplicación, procesando la imagen directamente en el dispositivo. Todo ocurre localmente. El resultado se muestra mediante una salida visual que presenta la enfermedad detectada, la zona afectada y las recomendaciones generales asociadas.

En la Tabla 32 se detallan las actividades para la captura y el diagnóstico con IA.

Tabla 32.

Sprint Backlog – Sprint 5

Historia	Nombre	Actividad desarrollada
HU06	Captura y selección de imágenes	Implementación del flujo para seleccionar una parcela, tomar una fotografía o cargar una imagen desde la galería.
HU07	Diagnóstico con inteligencia artificial	Integración del modelo <i>TensorFlow Lite</i> para procesar la imagen y mostrar el resultado del diagnóstico dentro de la aplicación.

Fuente: Elaboración propia.

El diagnóstico con inteligencia artificial inicia cuando el usuario selecciona una parcela y carga la imagen que será analizada. Este recorrido se documentó en un diagrama de secuencia, donde intervienen la aplicación móvil, el modelo TensorFlow Lite y Firebase Firestore. La representación muestra el paso de la imagen por la inferencia local, el postprocesamiento visual, el registro del diagnóstico y la entrega del resultado final al usuario.

En la Figura 19 se representa el flujo completo seguido para procesar, visualizar, guardar y presentar el diagnóstico generado por inteligencia artificial dentro de la aplicación móvil.

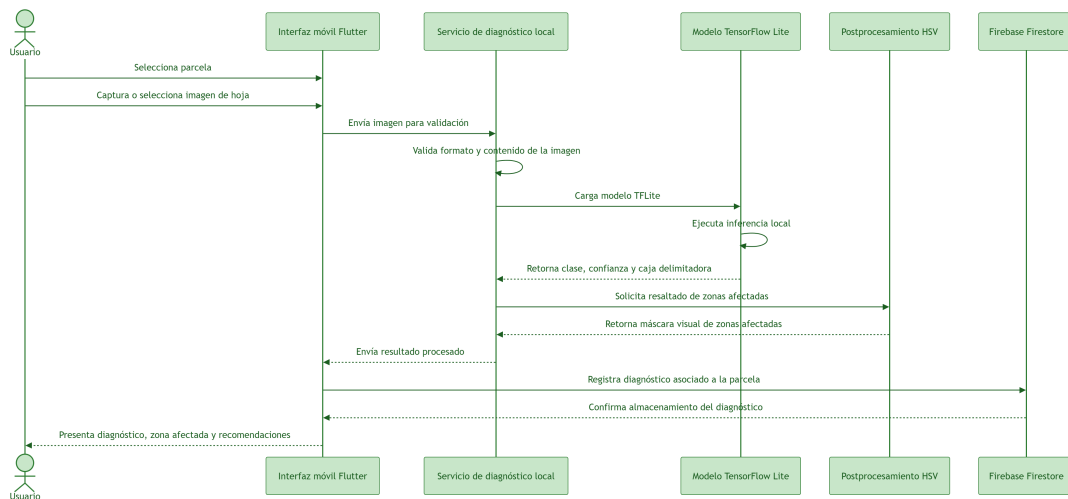


Figura 19.
Diagrama de secuencia del diagnóstico con inteligencia artificial

Fuente: Elaboración propia.

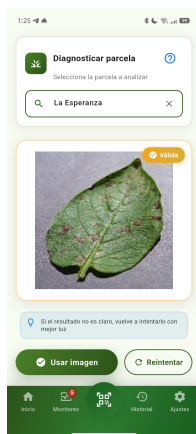
Incremento desarrollado: Como resultado de esta iteración se obtuvo el módulo de diagnóstico con inteligencia artificial. Se permitió al usuario seleccionar una parcela, cargar una imagen desde la galería o tomar una fotografía, validar la imagen antes del análisis y procesarla mediante el modelo integrado en la aplicación. Finalmente, el sistema presenta el resultado del diagnóstico, la zona afectada y las recomendaciones generales vinculadas a la enfermedad detectada.

Como evidencia del módulo desarrollado, se muestran las pantallas principales del flujo de diagnóstico. Estas interfaces presentan la selección de parcela, la carga y validación de la imagen, el procesamiento con inteligencia artificial, el resultado obtenido, las recomendaciones generales y los consejos para capturar una fotografía adecuada.

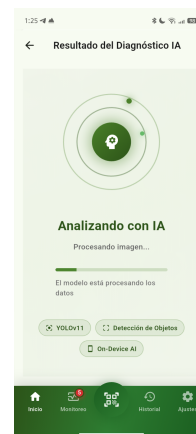
En la Figura 20 se presentan las interfaces del flujo de análisis y resultado del diagnóstico. Este recorrido incluye la selección de parcela, la validación de imagen, el procesamiento con IA y la entrega de recomendaciones al usuario.



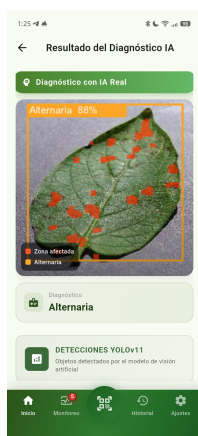
(a) Selección de parcela



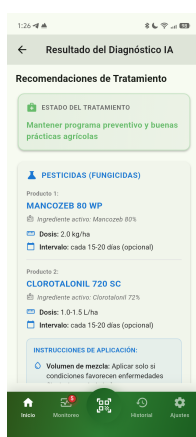
(b) Validación de imagen



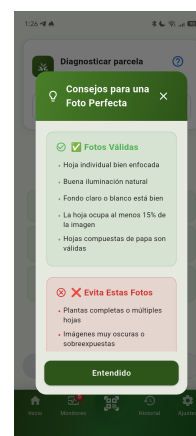
(c) Procesamiento con IA



(d) Resultado del diagnóstico



(e) Recomendaciones de tratamiento



(f) Consejos de captura

Figura 20.
Pantallas del módulo de diagnóstico con inteligencia artificial

Fuente: Elaboración propia.

Para la visualización del diagnóstico se combinaron dos procesos. El modelo YOLO-26-N con ajuste BoxGain, seleccionado por su mejor rendimiento experimental, permitió identificar la región donde se detectó la enfermedad, representada mediante un recuadro sobre la imagen analizada. De forma complementaria, se aplicó un procesamiento basado en el espacio de color HSV [43] para resaltar las zonas afectadas dentro de la hoja, diferenciando los píxeles asociados a lesiones de aquellos correspondientes al tejido sano o al fondo.

Este procesamiento permitió mejorar la interpretación visual del resultado, ya que el usuario no solo observa la clase detectada por el modelo, sino también una marcación aproximada

de las áreas con síntomas visibles. De esta manera, el diagnóstico se presenta de forma más comprensible dentro de la aplicación móvil.

Revisión del sprint: Se comprobó que el usuario pudiera seleccionar una parcela, capturar o cargar una imagen, validar la fotografía y ejecutar el diagnóstico mediante el modelo integrado. También se comprobó que la aplicación mostrara el resultado obtenido, la zona afectada, la clase detectada y las recomendaciones generales asociadas.

En la Tabla 33 se verifica el cumplimiento de los criterios para el diagnóstico con IA.

Tabla 33.

Revisión de criterios de aceptación – Sprint 5

Criterio de aceptación	Cumple
El usuario puede seleccionar una parcela antes de realizar el diagnóstico.	Sí
El usuario puede tomar una fotografía desde la cámara del dispositivo.	Sí
El usuario puede seleccionar una imagen desde la galería.	Sí
La aplicación permite validar la imagen antes de ejecutar el análisis.	Sí
El modelo de inteligencia artificial procesa la imagen dentro de la aplicación.	Sí
El usuario puede visualizar la enfermedad detectada y la zona afectada.	Sí
La aplicación muestra recomendaciones de tratamiento asociadas al diagnóstico.	Sí
El sistema presenta consejos para capturar una imagen adecuada.	Sí

Fuente: Elaboración propia.

3.2.4.8 Sprint 6: Historial, monitoreo y configuración

La sexta iteración funcional se orientó al desarrollo de los módulos complementarios para la consulta, seguimiento y configuración de la aplicación móvil. En esta etapa se implementaron el historial de diagnósticos, el módulo de monitoreo de situaciones críticas y las opciones generales de configuración del usuario.

El historial permite revisar los diagnósticos realizados previamente, consultar el detalle de cada análisis y filtrar los registros según su estado. Por otra parte, el módulo de monitoreo muestra situaciones urgentes asociadas a las parcelas, facilitando la identificación de casos que requieren atención. Finalmente, el apartado de configuración permite editar el perfil del usuario y acceder a información general de la aplicación.

En la Tabla 34 se detallan las actividades para historial, monitoreo y configuración.

Tabla 34.

Sprint Backlog – Sprint 6

Historia	Nombre	Actividad desarrollada
HU08	Historial de diagnósticos	Implementación del listado de diagnósticos, filtros de consulta y detalle de cada análisis registrado.
HU09	Monitoreo de casos críticos	Desarrollo del módulo de monitoreo para visualizar situaciones urgentes asociadas a las parcelas.
HU10	Configuración de usuario	Implementación de ajustes, edición de perfil e información general de la aplicación.

Fuente: Elaboración propia.

Incremento desarrollado: Como resultado de esta iteración se obtuvo el conjunto de módulos finales de consulta y configuración. Se permitió al usuario revisar el historial de diagnósticos, acceder al detalle de cada análisis, identificar situaciones críticas en el módulo de monitoreo y administrar opciones relacionadas con su cuenta. Además, se incorporaron secciones informativas como colaboradores, políticas de privacidad, condiciones de uso, contacto, calificación y versión de la aplicación.

Como evidencia del módulo desarrollado, se presentan las pantallas correspondientes al historial de diagnósticos, detalle del análisis, monitoreo de casos críticos y configuración del usuario. Estas interfaces permiten consultar resultados previos, identificar parcelas que requieren atención y acceder a las opciones generales de la aplicación.

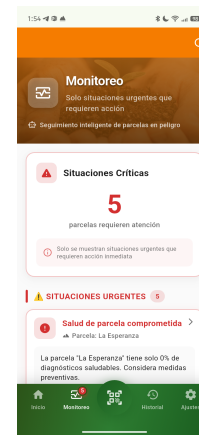
En la Figura 21 se presentan las interfaces para consultar diagnósticos, monitorear casos críticos y configurar la aplicación. Estas pantallas consolidan el seguimiento de resultados previos, la identificación de parcelas con atención prioritaria y la administración de opciones generales del usuario.



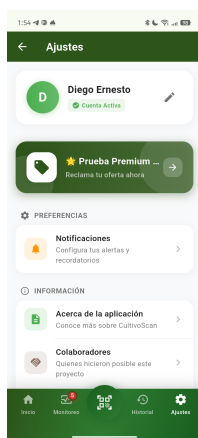
(a) Historial de diagnósticos



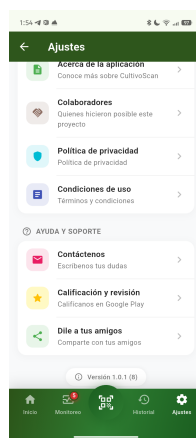
(b) Detalle del diagnóstico



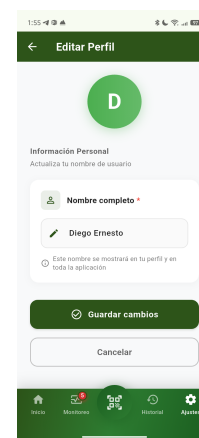
(c) Monitoreo de casos críticos



(d) Ajustes de usuario



(e) Opciones informativas



(f) Edición de perfil

Figura 21.

Pantallas de historial, monitoreo y configuración

Fuente: Elaboración propia.

Revisión del sprint: Se verificó que el usuario pudiera consultar el historial de diagnósticos, revisar el detalle de cada análisis y filtrar los registros según su estado. También se comprobó el funcionamiento del monitoreo de situaciones críticas y el acceso a las opciones de configuración, edición de perfil e información general de la aplicación.

En la Tabla 35 se verifica el cumplimiento de los criterios para historial, monitoreo y configuración.

Tabla 35.

Revisión de criterios de aceptación – Sprint 6

Criterio de aceptación	Cumple
El usuario puede consultar el historial de diagnósticos realizados.	Sí
El usuario puede filtrar diagnósticos según su estado.	Sí
El usuario puede revisar el detalle de un diagnóstico seleccionado.	Sí
La aplicación muestra situaciones críticas asociadas a las parcelas.	Sí
El usuario puede acceder al apartado de configuración.	Sí
El usuario puede editar la información de su perfil.	Sí
La aplicación permite acceder a información general, políticas, contacto y opciones de soporte.	Sí

Fuente: Elaboración propia.

Retrospectiva Scrum general: Al finalizar los sprints de desarrollo, se realizó una revisión general del proceso Scrum aplicado en la construcción de la aplicación móvil. Esta revisión permitió identificar los avances alcanzados, las dificultades encontradas y los ajustes incorporados durante la implementación progresiva de los módulos, especialmente en la integración de servicios externos, autenticación de usuarios, flujo de diagnóstico con inteligencia artificial, validación de imágenes y presentación de resultados.

La retrospectiva dejó ajustes puntuales en la aplicación: navegación más clara, estabilidad funcional y una presentación visual más ordenada. Además, se revisó que los incrementos desarrollados mantuvieran relación con las historias de usuario definidas durante la planificación.

3.2.4.9 Matriz de trazabilidad Scrum

Después de completar los sprints de desarrollo, se consolidó una matriz de trazabilidad Scrum para relacionar los requerimientos funcionales con las historias de usuario, las iteraciones implementadas, las evidencias generadas y las pruebas funcionales realizadas. Esta matriz permitió verificar la correspondencia entre la planificación inicial y los incrementos desarrollados durante la construcción de la aplicación móvil.

La matriz permitió revisar el recorrido de cada requerimiento dentro de la fase Scrum. En ella

se enlazaron las historias de usuario, el sprint asignado, las evidencias visuales y las pruebas funcionales aplicadas durante el desarrollo; en la Tabla 36 se relacionan los requerimientos, historias, sprints, evidencias y pruebas funcionales.

Tabla 36.

Matriz de trazabilidad Scrum de la aplicación móvil

RF	HU	Sprint	Evidencia	Prueba funcional	Estado
RF01	HU01	SP01	Pantallas de bienvenida y autenticación	Registro, inicio de sesión, recuperación de contraseña y acceso con Google.	Cumplido
RF02	HU02	SP02	Pantallas de gestión de parcelas	Creación, consulta, edición, duplicación, eliminación y búsqueda de parcelas.	Cumplido
RF03	HU03	SP03	Pantallas de actividades agrícolas	Creación de actividades, revisión en calendario y actualización de estado.	Cumplido
RF04	HU04	SP04	Pantallas de consulta climática	Revisión del pronóstico semanal y visualización del detalle por día.	Cumplido
RF05	HU05	SP04	Pantallas de precios de mercado	Visualización de precios referenciales de papa desde el servicio integrado.	Cumplido
RF06	HU06	SP05	Pantallas del flujo de diagnóstico	Captura o selección de imagen para iniciar el diagnóstico.	Cumplido
RF07	HU07	SP05	Pantallas del diagnóstico con IA	Procesamiento de la imagen y visualización del diagnóstico generado.	Cumplido
RF08	HU08	SP06	Pantallas de historial de diagnósticos	Consulta de diagnósticos realizados y visualización del detalle.	Cumplido
RF09	HU09	SP06	Pantallas de monitoreo	Consulta visual de casos críticos vinculados con las parcelas.	Cumplido
RF10	HU10	SP06	Pantallas de configuración	Edición de perfil y acceso a opciones informativas de la aplicación.	Cumplido

Fuente: Elaboración propia.

3.2.5 Fase de Post-Juego: pruebas, ajustes y publicación

La fase de Post-Juego correspondió a la etapa final del desarrollo de la aplicación móvil. En esta fase se realizaron pruebas funcionales, correcciones de errores, ajustes visuales y preparación de la versión final para su publicación y evaluación con usuarios.

En esta etapa se revisaron los módulos creados durante los sprints. También se validó el paso entre pantallas. Con esto se comprobó que las funciones principales de la aplicación trabajaran de forma correcta.

3.2.5.1 Pruebas funcionales de la aplicación móvil

Las pruebas funcionales se realizaron con el propósito de verificar el comportamiento de los módulos implementados durante los sprints. Estas pruebas permitieron comprobar que las funcionalidades principales respondieran correctamente a las acciones del usuario y que la información se registre, consulte y visualice de forma adecuada dentro de la aplicación.

Durante esta etapa se revisaron los módulos de autenticación, gestión de parcelas, actividades, pronóstico climático, precios de mercado, diagnóstico con inteligencia artificial, historial, monitoreo y configuración. La validación se realizó mediante la interacción directa con la aplicación móvil instalada en un dispositivo Android; en la Tabla 37 se resumen las pruebas aplicadas a los módulos principales de la aplicación.

Tabla 37.

Pruebas funcionales de la aplicación móvil

Módulo	Prueba realizada	Resultado
Autenticación	Registro, inicio de sesión, recuperación de contraseña y acceso con cuenta de Google.	Correcto
Parcelas	Creación, edición, duplicado, eliminación, búsqueda y revisión de parcelas.	Correcto
Actividades	Registro de actividades, vista en calendario y actualización de estado.	Correcto
Clima	Pronóstico semanal, detalle diario y recomendaciones agrícolas asociadas.	Correcto
Precios	Consulta de valores referenciales de papa desde el servicio disponible.	Correcto
Diagnóstico IA	Captura o carga de imagen, validación, procesamiento y salida del resultado.	Correcto
Historial	Revisión de diagnósticos guardados y apertura del detalle de análisis.	Correcto
Monitoreo	Revisión de casos críticos relacionados con las parcelas registradas.	Correcto

Configuración	Actualización de perfil y consulta de información general de la aplicación.	Correcto
---------------	---	----------

Fuente: Elaboración propia.

3.2.5.2 Corrección de errores y ajustes finales

Después de realizar las pruebas funcionales, se identificaron errores y ajustes necesarios en distintos módulos de la aplicación móvil. Estas correcciones permitieron mejorar la estabilidad del sistema, la autenticación de usuarios, la actualización de información externa, el procesamiento de imágenes y la persistencia de los diagnósticos realizados.

Se corrigieron fallos en funciones clave del sistema: inicio de sesión con cuenta de Google, actualización de precios de mercado, selección de imágenes para diagnóstico, validación de fotografías, guardado de resultados en el historial y notificaciones. También se ajustaron elementos visuales y de navegación, con el fin de mantener una experiencia de uso más clara y consistente. En la Tabla 38 se resumen las correcciones para mejorar la estabilidad de la aplicación.

Tabla 38.

Correcciones y ajustes realizados en la aplicación móvil

Módulo	Ajuste realizado	Estado
Autenticación	Corrección del acceso mediante cuenta de Google para usuarios en producción.	Corregido
Precios de mercado	Corrección en la actualización de precios de la papa desde el servicio integrado.	Corregido
Captura de imágenes	Selección de imágenes desde galería con flujo.	Corregido
Validación de imagen	Validación mejorada de fotografías para comprobar si corresponden a una hoja adecuada.	Corregido
Historial	Revisión del guardado de diagnósticos y acceso a los registros consultados.	Corregido
Notificaciones	Ajuste de avisos y recordatorios para su recepción dentro de la aplicación.	Corregido
Interfaz y navegación	Corrección de detalles visuales y del paso entre pantallas de la aplicación.	Corregido

Fuente: Elaboración propia.

3.2.5.3 Publicación en Google Play Store

Al cierre del desarrollo, la aplicación móvil quedó publicada en Google Play Store para dispositivos Android. Su disponibilidad se limitó a usuarios de Ecuador. El acceso fue gratuito. Con esto, la instalación se realizó desde un canal oficial de distribución.

Esta actividad permitió disponer de una versión accesible para la comunidad agrícola del país, facilitando su instalación, actualización y evaluación en un entorno real de uso. Además, representa una evidencia del nivel de madurez alcanzado por la aplicación, al estar disponible como una solución tecnológica orientada al apoyo del sector agrícola, específicamente en el seguimiento y diagnóstico de enfermedades en cultivos de papa. En la Figura 22 se evidencia la publicación de CultivoScan dentro de Google Play Store.

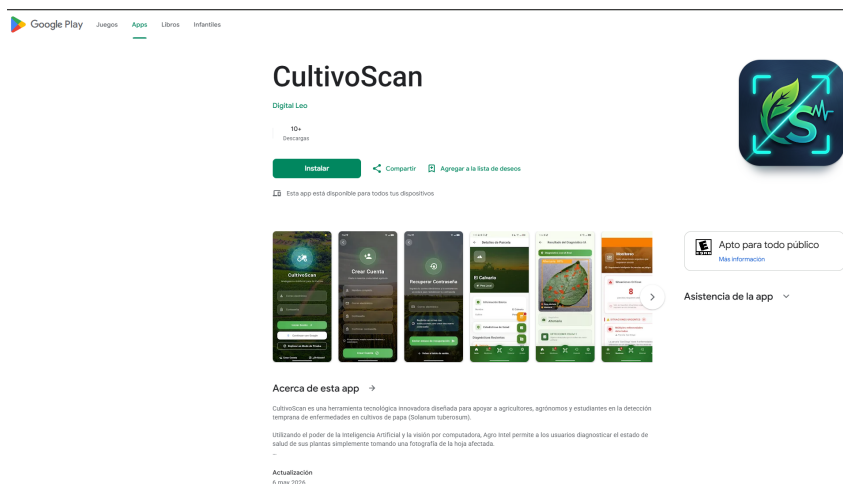


Figura 22.

Publicación de la aplicación móvil mediante Google Play Store.

Fuente: Elaboración propia.

3.2.5.4 Preparación para evaluación con usuarios

Una vez publicada la aplicación móvil y cerradas las correcciones finales, se preparó la evaluación con usuarios para conocer su percepción sobre el funcionamiento, utilidad y facilidad de uso del sistema. Se usó la versión disponible en Google Play Store. Esto permitió que los participantes instalaran e interactuaran con la aplicación en un entorno real.

La evaluación se realizó mediante una encuesta aplicada a usuarios que utilizaron la aplicación móvil, con el propósito de recopilar información relacionada con la experiencia de uso, comprensión de las funcionalidades, presentación de la información y percepción general del sistema. Para esta etapa se consideraron aspectos vinculados con la navegación, la claridad de las pantallas, la facilidad para realizar diagnósticos y la utilidad de la información presentada.

Los resultados obtenidos en esta evaluación no se presentan en este apartado, debido a que forman parte del análisis de resultados de la investigación. Por ello, la información recopilada mediante la encuesta será desarrollada posteriormente en el capítulo correspondiente.

3.3 Metodología CRISP-DM para el desarrollo del modelo de inteligencia artificial

Para el desarrollo del modelo de inteligencia artificial se aplicó la metodología CRISP-DM [44], con el propósito de organizar las etapas relacionadas con la comprensión del problema, adquisición de datos, preparación del conjunto de imágenes, entrenamiento, evaluación y despliegue del modelo. Su aplicación fue adecuada para esta investigación, debido a que el componente de inteligencia artificial requirió un proceso sistemático desde la construcción del dataset hasta su integración en la aplicación móvil.

CRISP-DM sirvió como base metodológica para construir el modelo orientado a detectar enfermedades en hojas de papa mediante imágenes digitales. El proceso no se limitó al uso de un dataset existente. Para el entrenamiento se preparó un conjunto híbrido, formado por imágenes públicas y fotografías propias recolectadas en campo, con el fin de contar con información visual más completa para las clases consideradas en el diagnóstico.

El trabajo del modelo se organizó con base en seis fases: comprensión del negocio, comprensión y adquisición de datos, preparación de datos, modelado, evaluación y despliegue. Bajo esa estructura se registraron decisiones técnicas como la selección de enfermedades, la recolección de imágenes propias, el uso de fondo blanco en la captura, la anotación en Roboflow, la preparación del dataset, el entrenamiento de distintas arquitecturas y la integración del modelo en la

aplicación móvil; en la Tabla 39 se sintetiza la aplicación de estas fases dentro del desarrollo del modelo de inteligencia artificial.

Tabla 39.

Fases de CRISP-DM aplicadas al modelo de inteligencia artificial

Fase	Aplicación en la investigación
Comprensión del negocio	Identificación de la necesidad de apoyar el diagnóstico de enfermedades fúngicas en hojas de papa mediante inteligencia artificial.
Comprensión y adquisición de los datos	Revisión de imágenes disponibles, toma de fotografías propias en campo y delimitación de las clases usadas en el diagnóstico.
Preparación de los datos	Trabajo del dataset en Roboflow: limpieza, etiquetado, división, preprocesamiento y aumento de las imágenes.
Modelado	Pruebas de arquitecturas para detección de objetos en un equipo local con aceleración por GPU.
Evaluación	Análisis del rendimiento de los modelos mediante métricas de detección e interpretación visual de las regiones atendidas por el modelo.
Despliegue	Exportación del modelo seleccionado a TensorFlow Lite e integración dentro de la aplicación móvil para inferencia local.

Fuente: Elaboración propia.

3.3.1 Aplicación de CRISP-DM en el desarrollo del modelo

CRISP-DM sirvió como base metodológica para ordenar el desarrollo del modelo de detección en este estudio. Bajo esa guía se trabajó la construcción del dataset, la preparación de imágenes, el entrenamiento de arquitecturas de detección, la evaluación con métricas de rendimiento y la elección del modelo final para integrarlo en la aplicación móvil. El proceso quedó más controlado. Esta adaptación ayudó a conectar cada fase de inteligencia artificial con las decisiones técnicas de la investigación, desde la definición de las clases de diagnóstico hasta la exportación del modelo compatible con TensorFlow Lite para ejecutarlo en dispositivos Android.

3.3.2 Fase 1: Comprensión del negocio

La primera fase de CRISP-DM permitió definir la necesidad que orientó el desarrollo del modelo de inteligencia artificial. En el contexto agrícola, el diagnóstico de enfermedades en hojas

de papa suele realizarse mediante observación visual, lo cual puede depender de la experiencia del productor o de la disponibilidad de asistencia técnica especializada. Frente a esta situación, se planteó el desarrollo de un modelo capaz de apoyar la identificación de enfermedades fúngicas en hojas de papa a partir de imágenes digitales.

El modelo se orientó a la detección de cuatro clases: *Alternaria*, *Lancha*, *Septoria* y Hoja sana [45]. La inclusión de estas categorías permitió cubrir las enfermedades fúngicas foliares consideradas dentro del alcance del estudio, además de incorporar una clase de referencia correspondiente a hojas sin síntomas visibles. Desde esta fase se estableció que el modelo debía ser compatible con un entorno móvil, por lo que se consideró tanto su rendimiento durante el entrenamiento como su posibilidad de ser exportado, integrado y ejecutado directamente en un dispositivo Android.

3.3.3 Fase 2: Comprensión y adquisición de los datos

La segunda fase de CRISP-DM se enfocó en la revisión, adquisición y organización inicial de las imágenes utilizadas para el entrenamiento del modelo de inteligencia artificial. Para esta investigación se construyó un conjunto de datos híbrido, conformado por imágenes públicas provenientes de PlantVillage [46] y fotografías propias recolectadas en campo, con el propósito de combinar condiciones controladas con capturas obtenidas en entornos agrícolas reales.

La adquisición de imágenes propias permitió complementar las clases disponibles en fuentes públicas y representar mejor el contexto agrícola local. En particular, la clase *Septoria* fue construida completamente con fotografías propias debido a la limitada disponibilidad de conjuntos públicos de libre acceso para esta enfermedad. El conjunto de datos final quedó conformado por cuatro clases: *Alternaria*, *Lancha*, *Septoria* y Hoja sana, seleccionadas de acuerdo con el alcance de la investigación.

3.3.3.1 Fuentes del conjunto de datos

El conjunto de datos se conformó mediante la integración de imágenes provenientes de PlantVillage y fotografías recolectadas en campo. PlantVillage aportó imágenes con condiciones controladas para las clases *Alternaria*, *Lancha* y *Hoja sana*, mientras que la recolección propia incorporó capturas obtenidas en parcelas agrícolas locales, principalmente para completar la clase *Septoria* y enriquecer la variabilidad visual del dataset.

Esta combinación permitió disponer de imágenes con diferentes condiciones de captura, fondos, iluminación y características visuales de las hojas, lo cual contribuyó a construir un conjunto de datos más adecuado para el entrenamiento del modelo de detección. De esta manera, las fuentes utilizadas no solo ampliaron la cantidad de imágenes disponibles, sino que también fortalecieron la representación de las clases consideradas dentro del alcance del estudio.

3.3.3.2 Recolección de imágenes propias en campo

Con el propósito de complementar el conjunto de datos público y representar condiciones agrícolas locales, se visitaron parcelas de cultivo de papa ubicadas principalmente en la provincia del Carchi. Entre los sectores considerados se incluyó la parroquia Santa Martha de Cuba y zonas agrícolas cercanas, donde se identificaron hojas con síntomas visibles mediante el apoyo de productores y personas de la localidad.

Durante las visitas se seleccionaron hojas con indicios visuales de enfermedad, las cuales fueron retiradas de forma individual y colocadas sobre una cartulina blanca para realizar la captura fotográfica. El uso del fondo blanco permitió trabajar bajo una condición visual controlada, aislar la hoja del entorno, mejorar la visibilidad de los síntomas foliares y reducir la interferencia de elementos externos durante la captura. Además, esta condición complementó las imágenes públicas del conjunto PlantVillage, aportando una variación adicional de fondo respecto a las imágenes existentes y fortaleciendo la diversidad visual del conjunto de datos. La hoja se ubicó extendida, procurando que los síntomas fueran visibles en la superficie superior. Para la captura

se utilizó un dispositivo móvil iPhone 11, manteniendo la cámara en posición perpendicular, con una distancia aproximada entre 20 y 30 cm y bajo iluminación natural.

Las imágenes obtenidas fueron transferidas como archivos para conservar su calidad original y posteriormente organizadas en Google Drive. Este repositorio fue compartido con profesionales del área agronómica de la provincia del Carchi, quienes apoyaron en la revisión y clasificación de las hojas según la enfermedad observada. Este proceso fortaleció la validez técnica del conjunto de datos propio; en la Figura 23 se muestra evidencia de la recolección realizada en parcelas agrícolas de la provincia del Carchi.



(a) Parcela agrícola visitada



(b) Hoja con síntomas visibles en campo

Figura 23.

Recolección de imágenes en parcelas agrícolas de la provincia del Carchi

Fuente: Elaboración propia.

En la Tabla 40 se resumen los parámetros durante la recolección de imágenes propias.

Tabla 40.

Parámetros considerados durante la recolección de imágenes propias

Parámetro	Descripción
Ubicación	Provincia del Carchi.
Sectores visitados	Santa Martha de Cuba y zonas agrícolas cercanas.
Parcelas visitadas	Aproximadamente 15 parcelas.
Fases de recolección	Julio-agosto de 2025 y marzo-abril de 2026.
Dispositivo	iPhone 11.
Superficie fotografiada	Lado superior de la hoja.
Fondo	Cartulina blanca.
Ángulo de captura	Aproximadamente 90 grados.

Parámetro	Descripción (continuación)
Distancia aproximada	Entre 20 y 30 cm.
Iluminación	Natural.
Organización	Transferencia como archivo y almacenamiento en Google Drive.
Revisión técnica	Apoyo de profesionales del área agronómica.

Fuente: Elaboración propia.

3.3.3.3 Justificación de la recolección de la clase Septoria

Dentro del conjunto de datos, la clase Septoria presentó una condición particular, debido a que no se identificaron suficientes imágenes públicas de libre acceso para ser utilizadas directamente en la investigación. A diferencia de Alternaria, Lancha y Hoja sana, que contaban con imágenes disponibles en PlantVillage, esta enfermedad requirió una recolección específica en campo para completar el conjunto de enfermedades foliares consideradas en el estudio.

Para esta clase se recolectaron 548 imágenes propias, las cuales fueron revisadas y clasificadas con apoyo técnico del área agronómica. Este proceso permitió construir una categoría independiente dentro del dataset, ampliar el alcance del modelo y fortalecer su representatividad frente a síntomas que no estaban cubiertos por las fuentes públicas utilizadas, considerando que incrementar el volumen de muestras en clases minoritarias equilibra el conjunto de datos y mejora significativamente la precisión del modelo [47].

3.3.3.4 Clases consideradas para el diagnóstico

Para el desarrollo del modelo se definieron cuatro clases de salida: Alternaria (*Alternaria solani*), Lancha o Tizón tardío (*Phytophthora infestans*), Septoria (*Septoria sp.*) y Hoja sana. Estas categorías patológicas foliares fueron seleccionadas de acuerdo con el alcance de la investigación, ya que representan las afecciones más comunes y representativas para el entrenamiento de sistemas de visión artificial enfocados en el diagnóstico de cultivos de papa [34].

Las clases se definieron según los rasgos visibles del follaje. Algunas muestran manchas.

Otras presentan lesiones concéntricas, variaciones de color o ausencia de síntomas. La categoría Hoja sana ayudó a separar el tejido afectado del tejido sin señales de enfermedad. Con las cuatro categorías también fue más ordenado construir el conjunto de datos, anotar en Roboflow y entrenar la red neuronal de forma supervisada; en la Tabla 41 se resumen las clases consideradas para el diagnóstico del modelo.

Tabla 41.

Clases consideradas para el diagnóstico del modelo

Clase	Descripción
Alternaria	Enfermedad foliar asociada a manchas oscuras, generalmente de forma irregular o concéntrica, que afectan el tejido de la hoja.
Lancha	Afectación foliar que puede generar lesiones oscuras, avance rápido del daño y deterioro visible del follaje en condiciones favorables de humedad.
Septoria	Enfermedad caracterizada por manchas pequeñas o lesiones visibles en la hoja, generalmente con zonas claras o grisáceas y bordes más oscuros.
Hoja sana	Hoja sin presencia visible de manchas, lesiones o cambios de coloración asociados a enfermedades. Se utilizó como clase de referencia.

Fuente: Elaboración propia.

3.3.3.5 Distribución del conjunto de datos

El conjunto de datos final quedó conformado por 3060 imágenes distribuidas en cuatro clases. Del total de imágenes, 2152 pertenecieron a PlantVillage y 908 fueron fotografías propias tomadas en campo. Así quedó identificado el aporte de cada fuente. También se mantuvo un control directo sobre el origen de las imágenes usadas durante el entrenamiento.

La distribución del dataset no fue uniforme entre clases. Alternaria y Lancha agruparon más imágenes, mientras que Hoja sana y Septoria tuvieron una representación menor. Este comportamiento se tomó en cuenta al preparar, dividir, anotar y aumentar los datos, buscando que el conjunto resultante fuera útil para el entrenamiento supervisado del modelo de detección; en la Tabla 42 se presenta el conteo de imágenes por clase y procedencia.

Tabla 42.

Conteo de imágenes por clase y procedencia

Categoría	Campo propio	PlantVillage	Total
Alternaria	57	1000	1057
Lancha	72	1000	1072
Hoja sana	231	152	383
Septoria	548	0	548
Total	908	2152	3060

Fuente: Elaboración propia.

Con esta distribución quedó lista la base inicial para preparar el conjunto de datos. Las imágenes se ordenaron por clase y por fuente. Así se tuvo mayor claridad sobre su procedencia y se avanzó con más control en la depuración, anotación, división y aumento de datos dentro de Roboflow.

3.3.4 Fase 3: Preparación de los datos

Una vez definido el conjunto de imágenes, se realizó la preparación del dataset para el entrenamiento del modelo de detección. Esta fase permitió organizar las imágenes por clase, revisar su calidad y establecer una estructura compatible con las herramientas utilizadas durante la anotación y el entrenamiento.

El proceso se desarrolló principalmente en Roboflow, debido a que esta plataforma permitió centralizar la carga de imágenes, la revisión de clases, el etiquetado, la división del conjunto de datos, el preprocesamiento y el aumento de imágenes [48]. Antes del entrenamiento, se verificó que las imágenes correspondieran a la clase asignada, mantuvieran una calidad visual adecuada y que las transformaciones aplicadas no alteraran la interpretación de los síntomas presentes en las hojas.

3.3.4.1 Organización y depuración de imágenes

Las imágenes recolectadas en campo y las obtenidas de PlantVillage fueron organizadas de acuerdo con la clase correspondiente: Alternaria, Lancha, Septoria y Hoja sana. Esta clasificación inicial permitió separar las muestras antes de cargarlas en Roboflow y mantener una estructura clara para las siguientes etapas del proceso.

Durante la revisión se descartaron fotografías borrosas, repetidas, mal enfocadas, con baja calidad visual o en las que la hoja y los síntomas no se observaban con claridad. Esta depuración permitió mejorar la calidad inicial del dataset y reducir la posibilidad de incorporar imágenes incorrectas o poco representativas que pudieran afectar el aprendizaje del modelo [49].

3.3.4.2 Anotación de imágenes y estrategia de etiquetado en Roboflow

Después de organizar y depurar el conjunto de imágenes, se utilizó Roboflow para realizar el proceso de anotación. Esta plataforma permitió cargar las imágenes por clase, revisar las etiquetas disponibles y generar las cajas delimitadoras necesarias para el entrenamiento del modelo de detección de objetos.

El etiquetado se trabajó tomando la hoja como unidad central del análisis. Cada muestra fue marcada con una caja rectangular que cubría la hoja completa. Después se asignó la clase según su apariencia visual: Alternaria, Lancha, Septoria u Hoja sana. Este criterio se mantuvo alineado con la aplicación móvil, donde el usuario carga o captura una imagen de una hoja para recibir un diagnóstico general.

Inicialmente se revisó la posibilidad de etiquetar lesiones individuales; sin embargo, este enfoque presentó limitaciones debido a la variabilidad, tamaño reducido y dificultad de delimitación de ciertas manchas, especialmente en Septoria. Por esta razón, el modelo no fue entrenado para segmentar lesiones individuales, sino para localizar la hoja y reconocer su estado general [50], [51]. La identificación de posibles zonas afectadas fue abordada posteriormente

mediante un postprocesamiento visual complementario dentro de la aplicación.



Figura 24.
Ejemplo de anotación de hoja en Roboflow

Fuente: Elaboración propia.

La Figura 24 muestra un ejemplo del proceso de anotación realizado en Roboflow, donde la hoja fue delimitada mediante una caja y asociada a la clase correspondiente para el entrenamiento del modelo de detección.

3.3.4.3 División del conjunto de datos

Después del proceso de anotación, el conjunto de datos fue dividido en tres subconjuntos: entrenamiento, validación y prueba. Esta división permitió separar las imágenes utilizadas para el aprendizaje del modelo de aquellas destinadas al seguimiento del entrenamiento y a la evaluación final.

La partición inicial se realizó en Roboflow bajo una distribución 80/10/10 a partir de las 3060 imágenes originales. Posteriormente, se aplicó aumento de datos sobre el subconjunto de entrenamiento, generando una versión final de 12852 imágenes. Los conjuntos de validación y prueba se mantuvieron sin aumentación, con el propósito de evaluar el modelo sobre imágenes no transformadas [52]; en la Tabla 43 se presenta la división final del conjunto de datos utilizado en el entrenamiento.

Tabla 43.

División final del conjunto de datos utilizado en el entrenamiento

Subconjunto	Cantidad de imágenes	Uso
Entrenamiento	12240	Ajuste de los parámetros del modelo.
Validación	306	Seguimiento del comportamiento durante el entrenamiento.
Prueba	306	Evaluación final del desempeño del modelo.
Total	12852	Dataset final preparado.

Fuente: Elaboración propia.

Esta organización permitió conservar conjuntos de validación y prueba independientes del proceso de aumento de datos. De esta manera, el modelo fue entrenado con un mayor volumen de imágenes, mientras que su rendimiento pudo ser evaluado con datos originales no transformados.

3.3.4.4 Preprocesamiento aplicado al conjunto de datos

Antes de generar la versión final del dataset, se aplicaron técnicas de preprocesamiento en Roboflow con el propósito de estandarizar las imágenes y preparar el conjunto de datos para el entrenamiento. Este proceso permitió reducir diferencias relacionadas con la orientación, el tamaño y el contraste de las imágenes.

Las transformaciones aplicadas fueron seleccionadas para mantener la proporción de la hoja y conservar la información visual de los síntomas. De esta manera, las imágenes quedaron ajustadas a una resolución uniforme, sin deformar el contenido principal de la fotografía; en la Tabla 44 se presentan las técnicas de preprocesamiento aplicadas en Roboflow.

Tabla 44.

Técnicas de preprocesamiento aplicadas en Roboflow

Técnica	Descripción
Autoorientación	Corrección automática de la orientación de las imágenes.
Cambio de tamaño	Ajuste de las imágenes a 1024×1024 píxeles con bordes blancos.
Ajuste automático del contraste	Aplicación de ecualización adaptativa para mejorar la visibilidad de detalles en la hoja.

Fuente: Elaboración propia.

Estas técnicas permitieron obtener imágenes con una estructura más uniforme antes del entrenamiento. Además, ayudaron a mantener la coherencia entre las fotografías propias y las imágenes públicas utilizadas dentro del conjunto de datos. Con ello, se redujeron variaciones innecesarias que pudieran afectar el aprendizaje del modelo durante la fase de entrenamiento [53].

3.3.4.5 Aumento de datos

Con el fin de ampliar la variabilidad del conjunto de entrenamiento, se aplicaron transformaciones sobre las imágenes originales manteniendo su clase correspondiente. Este procedimiento permitió generar nuevas versiones de las hojas sin modificar la información visual principal de los síntomas presentes en cada imagen. Con ello, se buscó mejorar la capacidad del modelo para reconocer variaciones de orientación, iluminación y pequeños cambios visuales durante el entrenamiento.

El aumento de datos se realizó en Roboflow únicamente sobre el subconjunto de entrenamiento, respetando la separación previamente definida para validación y prueba. Esta decisión permitió incrementar el volumen de datos utilizados para el aprendizaje del modelo, sin alterar los subconjuntos destinados al seguimiento y evaluación del rendimiento; en la Tabla 45 se presentan las técnicas de aumento de datos aplicadas en Roboflow.

Tabla 45.

Técnicas de aumento de datos aplicadas en Roboflow

Técnica	Configuración aplicada
Volteo	Horizontal y vertical.
Rotación	Entre -15° y $+15^\circ$.
Brillo	Entre -15% y $+15\%$.
Difuminado	Hasta 1.5 píxeles.
Ruido	Hasta el 1% de los píxeles.
Multiplicador	Generación de una versión aumentada 5x.

Fuente: Elaboración propia.

Como resultado de este proceso, el conjunto de datos pasó de 3060 imágenes originales a una

versión final de 12852 imágenes. Esta versión aumentada permitió fortalecer el entrenamiento del modelo y mejorar su capacidad de generalización frente a variaciones de orientación, iluminación y alteraciones visuales menores presentes en las hojas [54].

3.3.5 Fase 4: Modelado

Con el dataset ya listo, empezó el entrenamiento de los modelos de detección de objetos. En esta fase se compararon distintas arquitecturas. El objetivo fue claro: identificar la condición de una hoja de papa a partir de imágenes digitales.

El modelado se realizó mediante arquitecturas basadas en YOLO, debido a su capacidad para realizar detección de objetos en imágenes y su aplicabilidad en escenarios donde se requiere localizar una región específica y asociarla con una clase [55]. En este caso, la hoja fue considerada como la región principal de análisis, siguiendo la estrategia de detección a nivel de hoja definida en la fase anterior.

La experimentación buscó comparar modelos bajo distintas condiciones de entrenamiento, considerando arquitectura, resolución, tamaño de lote, número de épocas y parámetros de optimización. Con estos resultados se eligió una alternativa adecuada para su posterior despliegue dentro de la aplicación móvil.

3.3.5.1 Entorno de entrenamiento local

El entrenamiento de los modelos se realizó en un entorno local con aceleración por GPU [56], con el propósito de mantener control sobre los experimentos, los archivos generados y las configuraciones utilizadas. Esta decisión permitió ejecutar varias pruebas sin depender de servicios externos de cómputo y facilitó el seguimiento directo de los resultados obtenidos.

El entorno tuvo que configurarse con cuidado por el tamaño final del dataset y por las arquitecturas que se iban a evaluar. Se trabajó de forma local. Eso permitió modificar la resolución de

entrada, el tamaño de lote, las épocas de entrenamiento y otros parámetros según lo que requería cada experimento.

En la Tabla 46 se presenta el entorno de hardware y software utilizado para el entrenamiento. Esta información registra las condiciones técnicas en las que se ejecutaron los modelos y deja una referencia clara del entorno utilizado durante los experimentos.

Tabla 46.

Configuración local empleada en el entrenamiento de los modelos

Elemento	Detalle técnico
Hardware	
Memoria RAM	32 GB de RAM disponibles para la ejecución de los experimentos.
Procesamiento gráfico	NVIDIA GeForce RTX 4070 Ti, con 12 GB de VRAM.
Almacenamiento	Unidad principal de 1 TB y unidad secundaria de 2 TB destinada al respaldo del dataset.
Software y entorno	
Sistema operativo	Windows 11, versión de 64 bits.
Entorno de desarrollo	Jupyter Notebook utilizado desde Visual Studio Code.
Lenguaje de programación	Python 3.13.12.
Aceleración por hardware	CUDA Toolkit 11.8 junto con cuDNN.
Librerías de aprendizaje profundo	
Framework base	PyTorch 2.5.1+cu121 con ejecución sobre GPU.
Arquitecturas de detección	Ultralytics 8.4.41 para trabajar con modelos YOLO y RT-DETR.
Visión por computadora	OpenCV 4.13.0 y NumPy 2.2.6.
Análisis y monitorización	
Análisis de datos	Pandas, Matplotlib y Seaborn.
Monitorización de métricas	TensorBoard para el seguimiento de métricas durante el entrenamiento.

Fuente: Elaboración propia.

3.3.5.2 Selección de arquitecturas de detección

La selección de arquitecturas se realizó considerando que el modelo debía trabajar bajo un enfoque de detección a nivel de hoja. Por ello, no se buscó únicamente clasificar una imagen completa, sino localizar la región donde se encuentra la hoja y asociarla con una clase de diagnóstico, manteniendo coherencia con el proceso de anotación realizado en Roboflow.

Durante el modelado se probaron arquitecturas con diferentes niveles de complejidad, con el propósito de comparar su comportamiento frente al dataset construido. Esta comparación permitió analizar variaciones en capacidad de representación, tiempo de inferencia y costo computacional, considerando además la compatibilidad del modelo con TensorFlow Lite y Flutter para su integración posterior en la aplicación móvil [57]; en la Tabla 47 se presentan las arquitecturas consideradas durante el modelado.

Tabla 47.

Arquitecturas consideradas durante el modelado

Arquitectura	Propósito dentro de la evaluación
YOLO Nano	Evaluar modelos livianos con menor costo computacional y menor tiempo de inferencia.
YOLO Small	Evaluar precisión, velocidad y capacidad de detección en conjunto.
YOLO Medium	Comparar una variante de mayor representación con modelos más ligeros.
RT-DETR	Contrastar una arquitectura tipo Transformer en tareas de detección de objetos.

Fuente: Elaboración propia.

3.3.5.3 Configuración de los experimentos

Cada experimento fue organizado con un identificador interno, lo que permitió registrar la arquitectura utilizada y las condiciones aplicadas durante el entrenamiento. Esta organización fue necesaria porque se realizaron varias pruebas con diferentes configuraciones, con el propósito de analizar el comportamiento de los modelos frente al dataset preparado.

La configuración de los experimentos consideró ajustes relacionados con la resolución de entrada, el tamaño de lote, el optimizador, la tasa de aprendizaje, la regularización y el número de épocas. También se probaron estrategias asociadas a la función de pérdida y al aumento dinámico de datos durante el entrenamiento, con el propósito de mejorar la estabilidad del modelo y su capacidad de detección [58].

Con estos parámetros fue posible revisar el comportamiento de cada arquitectura en escenarios de entrenamiento diferentes. La Tabla 48 recoge los parámetros e hiperparámetros princi-

pales usados para configurar los experimentos.

Tabla 48.

Parámetros e hiperparámetros considerados en la configuración de los experimentos

Parámetro	Descripción
Arquitectura	Variantes convolucionales (YOLO) y de atención (RT-DETR) evaluadas.
Resolución de entrada	Tamaño de imagen (<i>imgsz</i>) redimensionado matemáticamente durante el entrenamiento.
Optimizador	Algoritmo utilizado para la convergencia y actualización de pesos, como AdamW o SGD.
Épocas y paciencia	Iteraciones máximas y umbral de parada temprana (<i>Early Stopping</i>) para prevenir el sobreajuste.
Tamaño de lote (<i>Batch</i>)	Cantidad de imágenes procesadas simultáneamente por iteración en la memoria VRAM.
Tasa de aprendizaje (LR)	Valor inicial (<i>lr0</i>) para controlar el tamaño de los pasos durante la optimización.
Decaimiento de pesos	Regularización matemática (<i>Weight Decay</i>) para penalizar la complejidad del modelo.
Ganancia de caja (<i>Box Gain</i>)	Modificador de la función de pérdida para priorizar la precisión geométrica de las cajas delimitadoras.
Aumento de datos dinámico	Estrategias aplicadas durante el entrenamiento, como <i>Mosaic</i> y <i>MixUp</i> , para fortalecer la robustez espacial.

Fuente: Elaboración propia.

3.3.5.4 Matriz de diseño experimental

Con el fin de organizar el proceso de modelado, se estructuró una matriz de diseño experimental que resume las configuraciones evaluadas durante el entrenamiento de los modelos de detección. Esta matriz permitió establecer una correspondencia entre cada experimento, la arquitectura utilizada, la resolución de entrada, el ajuste aplicado y el objetivo técnico de la prueba.

La organización experimental facilitó la comparación entre variantes YOLO y RT-DETR, así como el análisis del efecto de diferentes configuraciones sobre el rendimiento del modelo. De esta manera, los experimentos se plantearon como una secuencia controlada de evaluaciones orientadas a identificar la configuración con mejor desempeño para su integración posterior en la aplicación móvil [59]; en la Tabla 49 se sintetiza el diseño experimental de los modelos

evaluados durante la fase de modelado.

Tabla 49.

Síntesis del diseño experimental de los modelos evaluados

Grupo	Modelos evaluados	Propósito experimental
G01	YOLO-26-S y YOLO-11-S	Comparar arquitecturas pequeñas y variantes YOLO de referencia.
G02	YOLO-26-N	Evaluar ajustes de fondo blanco, optimizador, tasa de aprendizaje, resolución de entrada y BoxGain.
G03	YOLO-26-N con aumento de datos	Revisar el efecto de configuraciones de aumento sobre el comportamiento del modelo.
G04	YOLO-26-M	Contrastar una arquitectura de mayor capacidad con ajustes de aprendizaje y ponderación de clases.
G05	RT-DETR-L y RT-DETR-S	Comparar detectores basados en Transformer frente a las arquitecturas YOLO.
G06	Configuración seleccionada	Identificar el modelo con mejor desempeño para su integración en la aplicación móvil.

Fuente: Elaboración propia.

3.3.5.5 Entrenamiento de los modelos

Con las configuraciones definidas, se ejecutaron los entrenamientos en el entorno local preparado previamente. Cada modelo fue entrenado utilizando el conjunto de datos generado en Roboflow, considerando las imágenes aumentadas del subconjunto de entrenamiento y manteniendo los conjuntos de validación y prueba sin transformaciones adicionales.

Durante el entrenamiento se registraron los pesos generados por cada arquitectura y los archivos de salida correspondientes. Esta organización permitió conservar los resultados de cada experimento y facilitar su análisis posterior.

En algunos entrenamientos se aplicó parada temprana, con el propósito de detener el proceso cuando el modelo no presentaba mejoras relevantes. Esta estrategia permitió reducir entrenamientos innecesarios y conservar los pesos asociados al mejor desempeño alcanzado durante cada experimento [60].

La comparación detallada entre modelos se desarrolló posteriormente a partir de las métricas de evaluación definidas para detección de objetos. Los resultados obtenidos por cada arquitectura se presentan en el capítulo correspondiente.

3.3.6 Fase 5: Evaluación

En esta fase se analizó el desempeño de los modelos entrenados y se compararon las arquitecturas consideradas durante el modelado. Para ello, se emplearon métricas propias de detección de objetos, junto con tiempos de entrenamiento e inferencia, con el propósito de valorar tanto la precisión del modelo como su viabilidad para el posterior despliegue en la aplicación móvil.

El análisis consideró los resultados de validación y prueba. Los resultados de validación permitieron observar el comportamiento del modelo durante el entrenamiento, mientras que los resultados de prueba fueron utilizados como referencia principal para la comparación final entre arquitecturas. De esta manera, las métricas utilizadas para contrastar el rendimiento del modelo con otros trabajos se basaron en imágenes no utilizadas durante el ajuste de los pesos [61].

3.3.6.1 Métricas de evaluación utilizadas

Para comparar los modelos entrenados, se utilizaron métricas propias de detección de objetos. Estas métricas permitieron analizar el rendimiento de cada arquitectura desde dos aspectos principales: la localización correcta de la hoja y la asignación adecuada de la clase correspondiente.

Dentro de las métricas consideradas, mAP50 y mAP50-95 fueron especialmente importantes para valorar la calidad de las detecciones. mAP50 evalúa el desempeño del modelo con un umbral de coincidencia más flexible entre la caja predicha y la caja real. En cambio, mAP50-95 realiza una valoración más exigente, ya que considera varios umbrales de IoU y permite analizar con mayor detalle la precisión de localización de las cajas delimitadoras [62].

Además de las métricas de detección, se registraron tiempos de inferencia y entrenamiento como información complementaria. Estos valores fueron útiles para valorar no solo el rendimiento del modelo, sino también su viabilidad para ser integrado en una aplicación móvil; en la Tabla 50 se resumen las métricas utilizadas para la evaluación de modelos de detección.

Tabla 50.

Métricas utilizadas para la evaluación de modelos de detección

Métrica	Descripción
Precisión	Proporción de predicciones correctas realizadas por el modelo.
Recall	Capacidad del modelo para detectar los objetos presentes en las imágenes.
mAP50	Precisión media considerando un umbral IoU de 0.50.
mAP50-95	Precisión media calculada con umbrales IoU entre 0.50 y 0.95.
Tiempo de inferencia	Tiempo requerido para generar una predicción.
Tiempo de entrenamiento	Duración total del entrenamiento de cada experimento.

Fuente: Elaboración propia.

3.3.6.2 Protocolo de evaluación en validación y prueba

La evaluación de los modelos se organizó a partir de las particiones del conjunto de datos. El conjunto de entrenamiento se usó para ajustar los pesos de las arquitecturas evaluadas, mientras que el conjunto de validación permitió monitorear el comportamiento durante el entrenamiento y conservar las configuraciones con mejor desempeño intermedio.

El conjunto de prueba se dejó separado del entrenamiento y de la validación. Así se pudo medir el rendimiento final con imágenes que el modelo no había utilizado antes. En el Capítulo IV se presentan los resultados obtenidos en prueba como referencia principal de evaluación, y se complementan con una comparación frente a los resultados de validación para analizar la estabilidad del desempeño alcanzado [63].

3.3.6.3 Evaluación comparativa de modelos

La evaluación comparativa se organizó a partir de las métricas definidas previamente y de los registros generados durante cada experimento. Para cada arquitectura se consideraron los

resultados de detección, el tiempo de inferencia y la duración del entrenamiento, con el propósito de analizar tanto la calidad del modelo como su viabilidad técnica para la integración en la aplicación móvil [64].

El capítulo correspondiente reúne los resultados completos. Ahí queda el detalle. En esa sección se presentan las métricas obtenidas en cada experimento y se identifica la configuración que alcanzó el mejor desempeño experimental.

3.3.6.4 Criterio de selección del modelo para la aplicación móvil

La selección del modelo para la aplicación móvil se hizo considerando el rendimiento experimental obtenido en el conjunto de prueba, el tiempo de inferencia y la compatibilidad con el entorno móvil. Estos criterios fueron clave. Permitieron priorizar una configuración con desempeño adecuado en la detección y, al mismo tiempo, exportable e integrable dentro de la aplicación desarrollada en Flutter.

Con este proceso, se seleccionó la configuración con mejor equilibrio entre desempeño, eficiencia e integración técnica para su despliegue en CultivoScan. La comparación detallada de los experimentos y la justificación del modelo seleccionado se presentan en el capítulo de resultados.

3.3.7 Fase 6: Despliegue del modelo

En esta fase se preparó el modelo seleccionado para su funcionamiento dentro de la aplicación móvil. Para ello, el modelo entrenado fue convertido a un formato compatible con Flutter e incorporado como recurso interno del proyecto, lo que permitió ejecutar el diagnóstico directamente desde el dispositivo del usuario sin depender de servicios externos para analizar la imagen.

Esta decisión también permitió reducir costos operativos asociados al despliegue del compo-

nente de inteligencia artificial, ya que no fue necesario contratar servidores, servicios de inferencia en la nube ni infraestructura activa de procesamiento para mantener disponible el diagnóstico. A diferencia de una arquitectura web basada en servicios externos, el modelo permanece dentro de la aplicación móvil y se ejecuta localmente en el dispositivo del usuario. Firebase se utilizó únicamente como soporte para servicios complementarios de la aplicación, como autenticación, almacenamiento de información y gestión de datos del sistema.

El despliegue incluyó la exportación del modelo y su incorporación en la estructura del proyecto móvil. También se ejecutó la inferencia de forma local. Para mejorar la lectura del resultado, se añadió un postprocesamiento visual que marca las zonas afectadas en la hoja y muestra la clase detectada, el nivel de confianza y la caja delimitadora correspondiente.

3.3.7.1 Exportación del modelo a TensorFlow Lite

El modelo se llevó a la aplicación móvil después de convertir los pesos del entrenamiento a TensorFlow Lite. Con ese cambio se obtuvo un archivo más adecuado para ejecutarse en dispositivos móviles. También se conservó la compatibilidad con el entorno de desarrollo usado en Flutter.

El archivo final se generó en precisión de punto flotante de 32 bits y fue nombrado como `best_float32.tflite`. Esta versión se seleccionó para conservar la precisión del modelo durante la inferencia [65] y facilitar su integración posterior dentro de la estructura del proyecto móvil.

3.3.7.2 Integración del modelo en Flutter

El archivo `best_float32.tflite` fue incorporado dentro de la estructura del proyecto Flutter, específicamente en la carpeta destinada a los modelos de inteligencia artificial. Esta organización permitió mantener separado el recurso del modelo respecto al resto de módulos de la aplicación.

Para que el modelo pudiera ser utilizado durante la ejecución de la aplicación, el archivo fue declarado dentro del archivo `pubspec.yaml`. De esta manera, Flutter lo reconoce como un recurso interno del proyecto y permite su carga desde la aplicación móvil durante el diagnóstico; en la Figura 25 se muestra la ubicación del modelo *TensorFlow Lite* dentro de la estructura del proyecto Flutter.

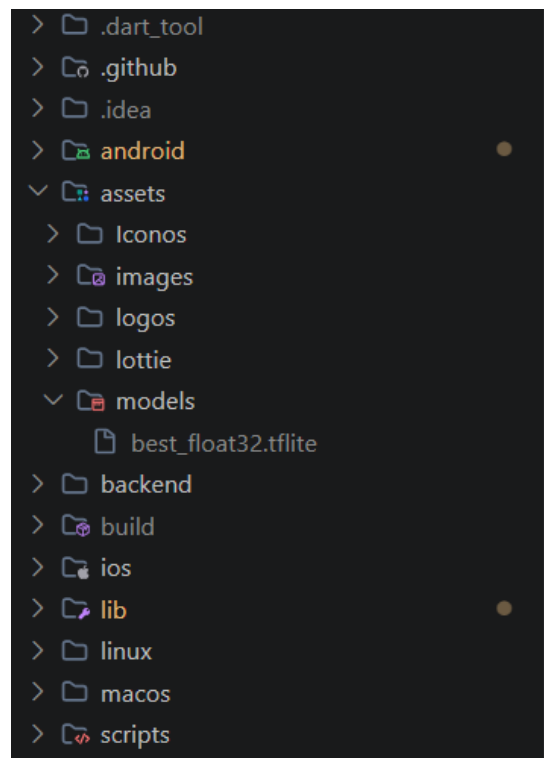


Figura 25.

Ubicación del modelo *TensorFlow Lite* en la estructura del proyecto Flutter

Fuente: Elaboración propia.

3.3.7.3 Inferencia local en el dispositivo móvil

El modelo integrado en la aplicación móvil se configuró para ejecutar la inferencia directamente en el dispositivo del usuario. Este enfoque permitió procesar la imagen de la hoja dentro de la misma aplicación, sin depender de un servidor externo para realizar el diagnóstico.

El procesamiento local ayudó a que la aplicación pudiera usarse en zonas agrícolas con conectividad limitada. La imagen se analiza dentro del dispositivo móvil. Con esto se disminuye la necesidad de enviar datos externos y se mantiene el diagnóstico en el propio equipo.

En la inferencia se obtienen tres datos: clase identificada, nivel de confianza y caja delimitadora de la zona analizada. Con esa salida, la aplicación genera una representación visual del diagnóstico, pensada para que el usuario interprete el resultado con mayor claridad.

3.3.7.4 Postprocesamiento visual de zonas afectadas

Dentro de la aplicación móvil se añadió un postprocesamiento visual a la salida del modelo YOLO-26-N con ajuste BoxGain. La idea fue facilitar la lectura del diagnóstico. Este paso no sustituyó la detección del modelo ni cambió la clase predicha; únicamente actuó como soporte gráfico para señalar posibles zonas afectadas en la hoja analizada.

El postprocesamiento se basó en el análisis de color mediante el espacio HSV, con el propósito de resaltar tonalidades asociadas a manchas, lesiones o cambios visibles en el tejido foliar [66]. A partir de este proceso, la aplicación generó una superposición visual sobre la imagen, facilitando que el usuario asocie visualmente el diagnóstico obtenido con las áreas que presentan síntomas visibles. En la Figura 26 se muestra un ejemplo de la imagen original de entrada y su respectiva salida con caja delimitadora y resaltado visual de zonas afectadas.



a) Imagen original de entrada



b) Resultado con postprocesamiento visual

Figura 26.

Ejemplo del postprocesamiento visual de zonas afectadas mediante HSV

Fuente: Elaboración propia.

CAPÍTULO IV

RESULTADOS Y ANÁLISIS

4.1 Resultados de la aplicación móvil

4.1.1 Instrumento de evaluación aplicado

La evaluación de CultivoScan se realizó mediante una encuesta estructurada en Microsoft Forms, aplicada a 25 participantes relacionados con el sector agrícola. El instrumento se enfocó en valorar la calidad en uso de la aplicación, considerando criterios de eficacia y eficiencia asociados al diagnóstico de enfermedades en hojas de papa y al apoyo brindado por la herramienta durante su uso en campo [67].

La encuesta estuvo compuesta por preguntas de caracterización, ítems con escala tipo Likert de cinco niveles y preguntas abiertas para recoger comentarios sobre utilidad, mejoras y recomendación de uso. Esta estructura permitió obtener información cuantitativa sobre la percepción de los usuarios y, al mismo tiempo, recopilar observaciones cualitativas relacionadas con la experiencia de uso de la aplicación; en la Tabla 51 se presentan los componentes del instrumento de evaluación aplicado.

Tabla 51.
Componentes del instrumento de evaluación aplicado

Apartado	Aspectos revisados
Datos generales	Rol del participante, experiencia agrícola y frecuencia de diagnóstico.
Eficacia percibida	Precisión del diagnóstico y coincidencia visual con la hoja analizada.
Eficiencia percibida	Rapidez de uso, fluidez de navegación y apoyo brindado por la aplicación.
Percepción general	Utilidad de la herramienta, sugerencias de mejora y recomendación de uso.

Fuente: Elaboración propia.

4.1.2 Criterios de interpretación de la escala de evaluación

Para el análisis de los ítems cerrados, se utilizó una escala tipo Likert de cinco niveles, asignando un valor numérico a cada opción de respuesta. Esta codificación permitió calcular la media de cada ítem y obtener una interpretación general de las dimensiones evaluadas; en la Tabla 52 se presentan los valores asignados a la escala Likert.

Tabla 52.

Valores asignados a la escala Likert

Respuesta seleccionada	Puntaje
Totalmente en desacuerdo	1
En desacuerdo	2
Ni de acuerdo ni en desacuerdo	3
De acuerdo	4
Totalmente de acuerdo	5

Fuente: Elaboración propia.

Para la presentación de los resultados se consideró como aceptación la suma de las respuestas *De acuerdo* y *Totalmente de acuerdo*. Asimismo, la media permitió interpretar el nivel de valoración obtenido en cada ítem y dimensión evaluada; en la Tabla 53 se presentan los criterios de valoración según la media obtenida.

Tabla 53.

Criterios de valoración según la media obtenida

Intervalo de media	Valoración obtenida
1.00 – 1.80	Muy bajo
1.81 – 2.60	Bajo
2.61 – 3.40	Medio
3.41 – 4.20	Alto
4.21 – 5.00	Muy alto

Fuente: Elaboración propia.

4.1.3 Relación de la evaluación con ISO/IEC 25010

La evaluación de la aplicación móvil se organizó considerando dimensiones relacionadas con la calidad en uso del sistema. Para mantener coherencia entre el instrumento aplicado y el análisis de resultados, las preguntas de la encuesta fueron asociadas a características de ISO/IEC 25010 [68] vinculadas a la eficacia, eficiencia y percepción general de uso.

Esta relación permitió interpretar los resultados obtenidos desde un enfoque de calidad del software, considerando no solo la valoración general de los usuarios, sino también la capacidad de la aplicación para apoyar el diagnóstico, responder de forma ágil durante el uso y aportar utilidad en el contexto agrícola; en la Tabla 54 se presenta la relación entre la evaluación aplicada e ISO/IEC 25010.

Tabla 54.

Relación entre la evaluación aplicada e ISO/IEC 25010

Característica ISO/IEC 25010	Dimensión evaluada	Justificación
Eficacia	Eficacia percibida	Evalúa si la aplicación apoya adecuadamente el diagnóstico visual de enfermedades en hojas de papa y si los resultados presentados son comprensibles para el usuario.
Eficiencia	Eficiencia percibida	Evalúa la rapidez, fluidez y facilidad con la que el usuario puede capturar o seleccionar una imagen, procesarla y obtener el resultado del diagnóstico.
Satisfacción y utilidad percibida	Percepción general	Recoge la aceptación del usuario, la utilidad observada durante el uso de la aplicación y las oportunidades de mejora identificadas a partir de la experiencia de evaluación.

Fuente: Elaboración propia.

4.1.4 Caracterización de los participantes

La encuesta fue respondida por 25 participantes vinculados con el sector agrícola. La muestra incluyó agricultores, ingenieros agrónomos, técnicos agrícolas, estudiantes del área agropecuaria, extensionistas y técnicos de instituciones relacionadas con el sector agrícola.

La participación de usuarios con distintos niveles de experiencia permitió obtener una valoración más amplia sobre el uso de CultivoScan. El contexto fue relevante. La mayoría de encuestados indicó realizar diagnósticos de enfermedades en campo de forma semanal, lo que aportó mayor relación con el uso real de la aplicación; en la Tabla 55 se presenta la caracterización de los participantes encuestados.

Tabla 55.

Caracterización de los participantes encuestados

Variable	Categoría	Frecuencia
Cargo o rol	Agricultor / productor agrícola	5
	Ingeniero agrónomo	5
	Estudiante del área agropecuaria	5
	Técnico agrícola	4
	Técnico de institución pública o privada	3
	Extensionista agrícola	2
	Otros	1
Experiencia agrícola	Menos de 1 año	1
	Entre 1 y 3 años	12
	Entre 4 y 7 años	11
	Más de 7 años	1
Nivel de experiencia en identificación de enfermedades	Muy baja	2
	Baja	4
	Media	14
	Alta	5
Frecuencia de diagnóstico en campo	Mensualmente	5
	Semanalmente	17
	Diariamente	3

Fuente: Elaboración propia.

La evaluación fue respondida, en su mayoría, por participantes con entre uno y siete años de experiencia agrícola. El perfil era pertinente. También predominó un nivel medio en la identificación de enfermedades y una frecuencia semanal de diagnóstico en campo, lo que permitió obtener resultados desde usuarios vinculados directamente con el problema que aborda la aplicación.

4.1.5 Resultados de eficacia percibida

En esta sección se presentan los resultados relacionados con la eficacia percibida del diagnóstico generado por CultivoScan. El análisis consideró la correspondencia entre la predicción de la aplicación, la observación visual de la hoja y las zonas afectadas identificadas durante la evaluación.

Los resultados muestran una valoración favorable en todos los ítems evaluados, con una aceptación general del 100 % y una media promedio de 4.77. Las valoraciones más altas se observaron en la diferenciación entre hojas sanas y enfermas, la confiabilidad del diagnóstico como apoyo en campo y la correspondencia visual entre las zonas señaladas y las áreas afectadas; en la Tabla 56 se presentan los resultados de eficacia percibida obtenidos en la evaluación.

Tabla 56.

Resultados de eficacia percibida

Ítem evaluado	De acuerdo	Totalmente de acuerdo	Aceptación	Media
Coincidencia del diagnóstico con la observación visual	11	14	100 %	4.56
Diferenciación entre hoja sana y hoja enferma	2	23	100 %	4.92
Correspondencia entre enfermedad detectada y afección presente	7	18	100 %	4.72
Consistencia en hojas con características similares	5	20	100 %	4.80
Claridad de la información presentada	8	17	100 %	4.68
Confiabilidad como apoyo para identificación en campo	4	21	100 %	4.84
Correspondencia de zonas señaladas con áreas afectadas	4	21	100 %	4.84
Promedio general	–	–	100 %	4.77

Fuente: Elaboración propia.

4.1.6 Resultados de eficiencia percibida

En esta sección se presentan los resultados relacionados con el comportamiento de la aplicación móvil durante el análisis de imágenes. Se consideraron aspectos como el tiempo de respuesta, la fluidez de uso y el apoyo percibido frente al diagnóstico visual tradicional.

La eficiencia percibida registró una media general de 4.78 sobre 5, valor que ubica la evaluación de los usuarios en un nivel favorable respecto al desempeño del sistema; en la Tabla 57 se presentan los resultados de eficiencia percibida obtenidos en la evaluación.

Tabla 57.

Resultados de eficiencia percibida

Ítem evaluado	De acuerdo	Totalmente de acuerdo	Aceptación	Media
Tiempo adecuado para mostrar resultados	1	24	100 %	4.96
Análisis rápido y sin interrupciones	9	16	100 %	4.64
Fluidez durante el uso en campo	5	20	100 %	4.80
Diagnóstico más rápido frente a métodos tradicionales	9	16	100 %	4.64
Optimización del tiempo en la identificación de enfermedades	3	22	100 %	4.88
Promedio general	–	–	100 %	4.78

Fuente: Elaboración propia.

El tiempo para mostrar resultados fue el ítem mejor valorado, con una media de 4.96 sobre 5. Junto con la optimización del tiempo y la fluidez durante el uso en campo, este resultado indica que los participantes percibieron la aplicación móvil como una herramienta eficiente durante el diagnóstico.

4.1.7 Percepción general y oportunidades de mejora

Las preguntas abiertas permitieron identificar observaciones complementarias sobre el uso de la aplicación móvil. Para el análisis se consideraron las respuestas más relevantes y relacio-

nadas directamente con el objetivo de la evaluación, descartando comentarios generales que no aportaron información técnica o funcional.

Las respuestas ubicaron mejoras pendientes en la experiencia de uso y en la precisión del diagnóstico en campo. La captura requiere más guía. También se identificó la necesidad de simplificar el manejo para usuarios con menor familiaridad tecnológica y ampliar progresivamente el reconocimiento de más enfermedades del cultivo; en la Tabla 58 se sintetizan las oportunidades de mejora identificadas por los usuarios.

Tabla 58.

Síntesis de oportunidades de mejora identificadas

Categoría	Síntesis de mejora
Facilidad de uso	Incorporar una interfaz más guiada para usuarios con poca experiencia tecnológica.
Captura de imagen	Agregar instrucciones visuales para tomar correctamente la fotografía de una sola hoja.
Condiciones de campo	Mejorar el desempeño del diagnóstico frente a variaciones de iluminación, fondo y calidad de imagen.
Cobertura del modelo	Ampliar el reconocimiento a más enfermedades presentes en el cultivo de papa.
Información de apoyo	Fortalecer las recomendaciones posteriores al diagnóstico para orientar mejor al usuario.

Fuente: Elaboración propia.

La experiencia de uso y la calidad en la captura de imágenes quedaron como puntos principales para próximas actualizaciones de la aplicación móvil. Aun con estas mejoras pendientes, el sistema se mantiene como apoyo al diagnóstico, junto con el criterio del usuario y las recomendaciones técnicas en campo.

4.1.8 Análisis general de la evaluación de la aplicación móvil

La evaluación de CultivoScan reflejó una percepción favorable en el contexto agrícola. La eficacia percibida registró una media general de 4.77 sobre 5 y la eficiencia percibida alcanzó 4.78 sobre 5, valores que muestran una aceptación positiva del apoyo brindado por la aplicación durante el diagnóstico de enfermedades en hojas de papa.

La valoración evidencia que la aplicación cumple su propósito como herramienta de apoyo para la identificación de enfermedades en campo. No obstante, los comentarios abiertos permitieron reconocer oportunidades de mejora relacionadas con la facilidad de uso, la guía para capturar imágenes y la ampliación del reconocimiento a más enfermedades. En conjunto, la evaluación aporta evidencia sobre la calidad en uso de la aplicación y sirve como base para continuar con el análisis del componente de inteligencia artificial desarrollado en la investigación.

4.2 Resultados del modelo de inteligencia artificial

4.2.1 Resultados generales de los experimentos

En esta sección se presentan los resultados obtenidos durante la evaluación de las arquitecturas de detección de objetos aplicadas al diagnóstico de enfermedades en hojas de papa. El análisis se organizó en dos niveles: primero, una comparación entre validación y prueba para revisar la estabilidad de los modelos representativos; segundo, una comparación general en el conjunto de prueba, considerado como referencia principal para valorar el rendimiento final de cada configuración.

El contraste entre validación y prueba se usó para comprobar si los modelos mantenían estabilidad ante imágenes no empleadas durante el ajuste. En esta revisión se incluyeron configuraciones representativas por familia de arquitectura y la configuración final YOLO-26-N con ajuste BoxGain, integrada después en la aplicación móvil; en la Tabla 59 se presenta la comparación entre resultados de validación y prueba en modelos representativos.

Tabla 59.

Comparación entre resultados de validación y prueba en modelos representativos

Arquitectura	Configuración	mAP50 Val.	mAP50 Test	mAP50-95 Val.	mAP50-95 Test
YOLO-26-S	Ajuste fino	99.42 %	97.98 %	98.83 %	97.21 %
YOLO-11-S	Modelo de comparación	99.26 %	98.55 %	98.30 %	97.60 %
YOLO-26-N	BoxGain, 1024 px / modelo integrado	99.11 %	98.89 %	98.03 %	98.05 %

Arquitectura	Configuración	mAP50 Val.	mAP50 Test	mAP50-95 Val.	mAP50-95 Test
YOLO-26-M	Línea base estable	99.38 %	98.28 %	98.10 %	97.22 %
RT-DETR-L	Transformer, versión large	97.46 %	96.32 %	96.74 %	95.76 %
RT-DETR-S	Transformer, versión small	97.47 %	96.01 %	96.01 %	94.64 %

Fuente: Elaboración propia.

Los resultados muestran que varias configuraciones alcanzaron valores altos en validación; sin embargo, el comportamiento en el conjunto de prueba permitió diferenciar mejor la capacidad de generalización. En este contexto, el modelo YOLO-26-N con ajuste BoxGain y resolución de entrada de 1024 píxeles presentó una diferencia mínima entre validación y prueba, con un mAP50 de 99.11 % en validación y 98.89 % en prueba. Además, el mAP50-95 se mantuvo estable, pasando de 98.03 % en validación a 98.05 % en prueba.

A partir de esta revisión, los resultados generales del conjunto de prueba se utilizaron como referencia principal para la comparación final de los experimentos. Este criterio permitió evaluar el rendimiento de cada configuración sobre imágenes independientes y seleccionar el modelo con mejor equilibrio entre precisión, capacidad de detección, estabilidad e integración funcional en la aplicación móvil; en la Tabla 60 se presentan los resultados generales de los experimentos en el conjunto de prueba.

Tabla 60.

Resultados generales de los experimentos en el conjunto de prueba

Arquitectura	Exp.	Configuración	mAP50	mAP50-95	Precisión	Recall	Inferencia
YOLO-26-S	E07	Ajuste fino	97.98	97.21	96.98	96.44	8.2 ms
YOLO-26-S	E05	MixUp	98.22	97.18	95.26	96.30	7.8 ms
YOLO-26-S	E08	Mosaic aumentado	98.56	97.51	97.00	97.67	7.0 ms
YOLO-11-S	Y11s	Modelo de comparación	98.55	97.60	97.60	98.00	3.7 ms
YOLO-26-N	E04	Ajuste con fondo blanco	98.71	97.63	96.14	96.67	5.3 ms
YOLO-26-N	E10	SGD con batch 16	98.46	97.44	96.68	97.98	4.17 ms
YOLO-26-N	E11	Tasa de aprendizaje baja	98.33	97.41	96.54	97.05	5.22 ms

Arquitectura	Exp.	Configuración	mAP50	mAP50-95	Precisión	Recall	Inferencia
YOLO-26-N	E12*	BoxGain, 1024 px / modelo integrado	98.89	98.05	96.99	95.76	5.06 ms
YOLO-26-N	E14	BoxGain, 800 px	98.41	74.28	95.31	95.20	4.26 ms
YOLO-26-N	E15	BoxGain, 640 px	55.21	39.37	60.51	59.39	2.68 ms
YOLO-26-N	E13	Aumento extremo	97.84	81.38	90.56	95.39	3.79 ms
YOLO-26-N	E09	Configuración de prueba final	98.00	96.79	96.82	96.04	3.6 ms
YOLO-26-M	Q1	Línea base estable	98.28	97.22	96.37	95.67	15.1 ms
YOLO-26-M	E01	Tasa de aprendizaje baja	98.38	96.52	95.02	96.90	15.4 ms
YOLO-26-M	E02	Ponderación de clases	96.28	80.67	69.74	98.36	15.8 ms
RT-DETR-L	E06	Transformer, versión large	96.32	95.76	96.76	97.71	22.5 ms
RT-DETR-S	RT-S	Transformer, versión small	96.01	94.64	97.27	95.70	21.33 ms

Fuente: Elaboración propia. El experimento resaltado y señalado con el símbolo * corresponde al mejor resultado obtenido en el conjunto de prueba y al modelo integrado en la aplicación móvil.

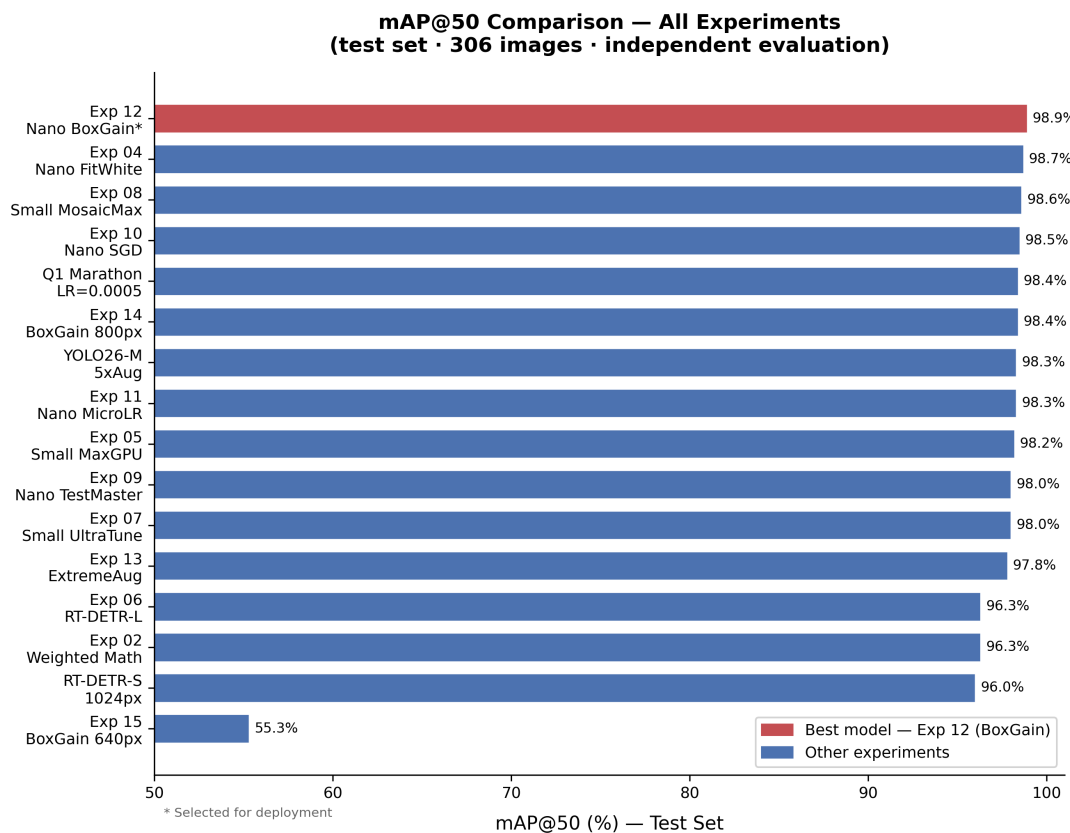


Figura 27.

Comparación del mAP50 en el conjunto de prueba para los experimentos evaluados

Fuente: Elaboración propia.

La Figura 27 complementa la Tabla 60 y permite visualizar de forma directa el comportamiento del mAP50 en los experimentos evaluados. En esta comparación, el experimento E12 destaca como la configuración con mayor rendimiento en el conjunto de prueba.

El experimento E12 quedó como configuración final del sistema, al presentar el mejor desempeño global en el conjunto de prueba y ser el modelo integrado en CultivoScan. Con ello se mantuvo la relación entre la evaluación experimental y el despliegue operativo de la aplicación móvil.

Las pruebas con reducción de resolución mostraron que el tamaño de entrada influye directamente en la calidad de detección. La configuración de 800 píxeles mantuvo un comportamiento aceptable. En cambio, la reducción a 640 píxeles provocó una caída considerable en las métricas, lo que indica que disminuir demasiado la resolución puede eliminar información visual relevante para diferenciar correctamente las enfermedades analizadas.

4.2.2 Comparación de arquitecturas de detección

La comparación por familias de modelos permitió analizar el comportamiento general de las arquitecturas YOLO y RT-DETR frente al conjunto de prueba. En la Tabla 61 se presenta el mejor resultado obtenido por cada arquitectura evaluada, considerando las métricas principales de detección y el tiempo de inferencia.

Tabla 61.

Comparación resumida de arquitecturas de detección

Arquitectura	Mejor configuración	mAP50	mAP50-95	Inferencia
YOLO-26-S	Mosaic aumentado	98.56	97.51	7.0 ms
YOLO-11-S	Modelo de comparación	98.55	97.60	3.7 ms
YOLO-26-N	BoxGain, 1024 px / modelo integrado	98.89	98.05	5.06 ms
YOLO-26-M	Tasa de aprendizaje baja	98.38	96.52	15.4 ms
RT-DETR-L	Transformer, versión large	96.32	95.76	22.5 ms
RT-DETR-S	Transformer, versión small	96.01	94.64	21.33 ms

Fuente: Elaboración propia.

La comparación favoreció a las variantes YOLO por su balance entre capacidad de detección y tiempo de respuesta. Dentro de las configuraciones evaluadas, YOLO-26-N con ajuste Box-Gain y resolución de entrada de 1024 píxeles obtuvo el mejor rendimiento global, por lo que fue integrado en la aplicación móvil para realizar el diagnóstico de manera local. RT-DETR también alcanzó resultados favorables, aunque su mayor tiempo de inferencia lo hizo menos conveniente para escenarios móviles donde se requiere una respuesta rápida.

4.2.3 Análisis del efecto de la resolución de entrada

El análisis tomó como base la configuración YOLO-26-N con BoxGain. Se probaron entradas de 1024, 800 y 640 píxeles. Con esta evaluación se revisó cómo cambia la detección de enfermedades en hojas de papa cuando se reduce la resolución de la imagen; en la Tabla 62 se presenta la comparación del desempeño según la resolución de entrada.

Tabla 62.

Comparación del desempeño según la resolución de entrada

Configuración	Resolución	mAP50	mAP50-95	Precisión	Recall
YOLO-26-N BoxGain	1024 px	98.89	98.05	96.99	95.76
YOLO-26-N BoxGain	800 px	98.41	74.28	95.31	95.20
YOLO-26-N BoxGain	640 px	55.21	39.37	60.51	59.39

Fuente: Elaboración propia.

Los resultados muestran que la resolución de 1024 píxeles permitió conservar el mejor desempeño general, especialmente en la métrica mAP50-95, que evalúa la calidad de localización con criterios más estrictos. Como se observa en la Figura 28, al reducir la entrada a 800 píxeles, el mAP50 se mantuvo alto, pero el mAP50-95 disminuyó de forma considerable, lo que indica una pérdida de precisión en la localización de las cajas delimitadoras.

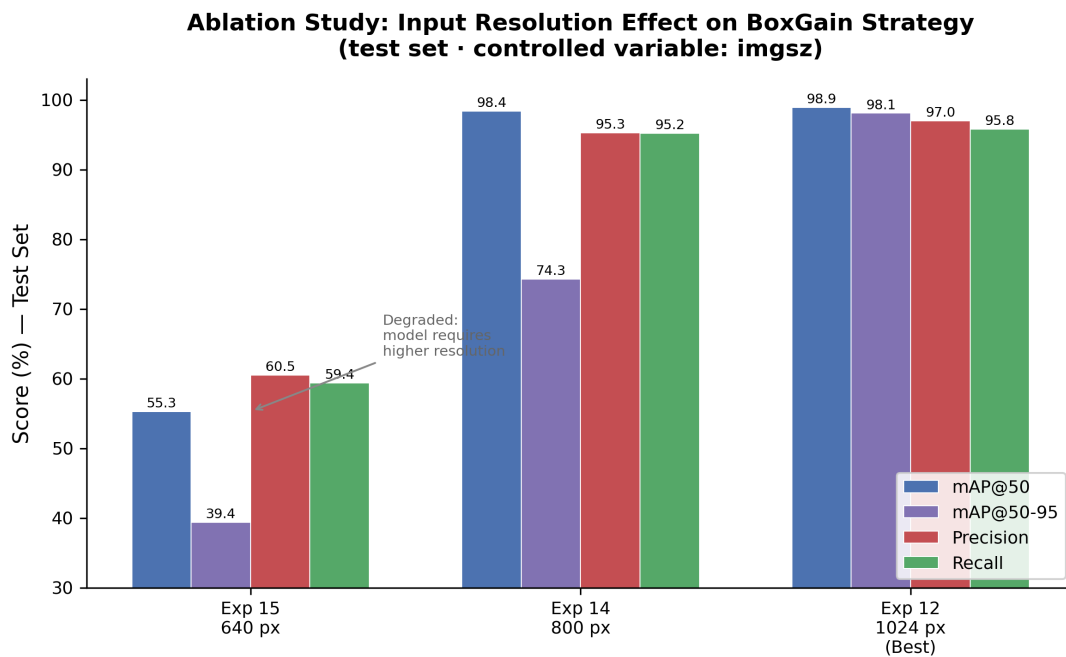


Figura 28.

Estudio de ablación del efecto de la resolución de entrada en la estrategia BoxGain

Fuente: Elaboración propia.

La configuración de 640 píxeles presentó una reducción marcada en todas las métricas evaluadas. Este comportamiento sugiere que, para el diagnóstico de enfermedades foliares, una reducción excesiva de la resolución afecta la conservación de detalles visuales como bordes, manchas y cambios de coloración, elementos necesarios para diferenciar correctamente las clases consideradas en el modelo.

4.2.4 Análisis del modelo seleccionado

A partir de la comparación experimental, se seleccionó como mejor configuración el modelo YOLO-26-N con ajuste BoxGain y resolución de entrada de 1024 píxeles. Esta configuración mostró el mejor comportamiento en el conjunto de prueba, especialmente en las métricas relacionadas con la detección y localización de objetos.

La selección de este modelo se fundamentó en su rendimiento global frente a las demás configuraciones evaluadas. Además, se consideró la estabilidad del entrenamiento y la capacidad

del modelo para conservar una localización precisa de las zonas afectadas. Este aspecto fue relevante en el análisis de enfermedades foliares, donde las lesiones pueden presentar tamaños reducidos y formas irregulares. En la Tabla 63 se reúnen las condiciones técnicas y las métricas principales del modelo seleccionado. La decisión final no dependió solo de esos valores. También se revisaron la estabilidad, la resolución de entrada, el tiempo de inferencia y la posibilidad real de integrarlo en la aplicación móvil.

Tabla 63.

Resumen técnico del modelo con mejor rendimiento experimental

Elemento	Descripción / Resultado
Arquitectura	YOLO-26-N
Experimento	E12
Configuración	BoxGain con resolución de entrada de 1024 píxeles.
Épocas configuradas	300
Épocas ejecutadas	146
Tamaño de lote (<i>batch size</i>)	4
Tasa de aprendizaje inicial	0.001
Decaimiento de pesos	0.0005
mAP50	98.89
mAP50-95	98.05
Precisión	96.99
Recall	95.76
Tiempo de inferencia	5.06 ms
Tiempo de entrenamiento	14 h 19 min 11 s
Peso del archivo final	5.23 MB
Total de parámetros	2 505 360

Fuente: Elaboración propia.

En la Tabla 64 se presentan los criterios técnicos considerados para seleccionar el modelo final.

Tabla 64.

Criterios técnicos considerados para la selección del modelo final

Criterio	Evidencia observada	Justificación técnica
Rendimiento en prueba	Mayor mAP50 y mAP50-95.	Mejor desempeño en imágenes independientes.
Estabilidad del modelo	Diferencia mínima entre validación y prueba.	Reduce indicios de sobreajuste.

Criterio	Evidencia observada	Justificación técnica
Resolución de entrada	Mejor resultado con 1024 píxeles.	Conserva detalles visuales relevantes.
Tiempo de inferencia	Inferencia de 5.06 ms.	Permite una respuesta rápida en la aplicación.
Compatibilidad móvil	Integración mediante TensorFlow Lite.	Facilita la ejecución local en Flutter.
Coherencia con el despliegue	Modelo integrado en CultivoScan.	Alinea evaluación experimental y uso real.

Fuente: Elaboración propia.

YOLO-26-N con ajuste BoxGain fue seleccionado como modelo final por su rendimiento alto, estabilidad experimental, rapidez de respuesta e integración funcional dentro de CultivoScan.

4.2.5 Resultados por clase del modelo seleccionado

En la Tabla 65 se presentan los resultados obtenidos por el modelo seleccionado en el conjunto de prueba. Se consideran las clases Alternaria, Lancha, Hoja sana y Septoria, junto con las métricas de precisión, recall, F1-score, mAP50 y mAP50-95.

Tabla 65.

Resultados por clase del modelo seleccionado

Clase	Precisión	Recall	F1-score	mAP50	mAP50-95
Alternaria	100.00 %	96.72 %	98.33 %	99.44 %	99.05 %
Lancha	99.02 %	94.83 %	96.88 %	99.22 %	96.90 %
Hoja sana	97.30 %	94.97 %	96.12 %	98.85 %	98.63 %
Septoria	91.66 %	96.36 %	93.95 %	98.08 %	97.68 %

Fuente: Elaboración propia.

Las cuatro clases evaluadas presentaron resultados altos con el modelo seleccionado. Alternaria obtuvo los valores más favorables. En Septoria, la precisión fue menor, pero se mantuvieron un recall alto y métricas mAP elevadas. En conjunto, los resultados muestran un comportamiento consistente tanto para las enfermedades como para las hojas sanas.

4.2.6 Dinámica de entrenamiento del modelo seleccionado

La dinámica de entrenamiento permitió observar la evolución del modelo seleccionado durante el proceso de aprendizaje. En la Figura 29 se presentan las curvas de pérdida y las métricas principales registradas durante las épocas de entrenamiento, considerando el comportamiento tanto en entrenamiento como en validación.

Las curvas muestran una reducción progresiva de las pérdidas de entrenamiento y un incremento sostenido de las métricas de detección durante la mayor parte del proceso. Para la evaluación se utilizó el archivo *best.pt*, correspondiente al mejor estado del modelo, con el cual se trabajó con la configuración que presentó el mejor desempeño durante el entrenamiento.

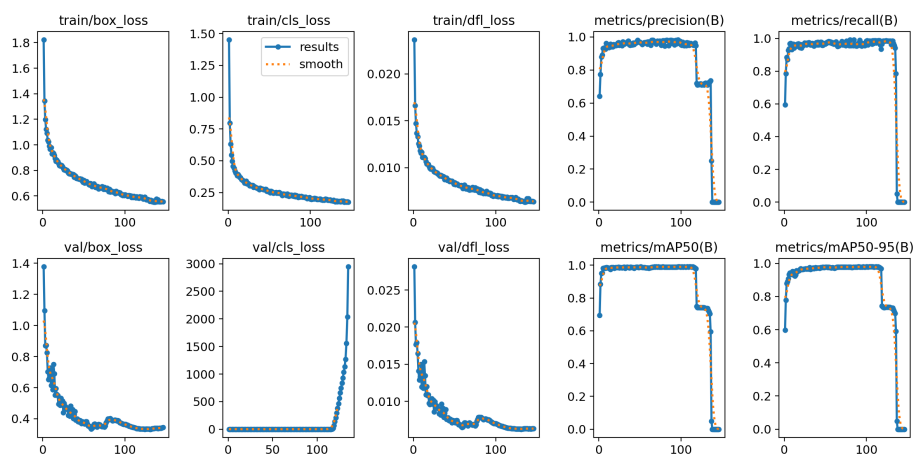


Figura 29.

Curvas de entrenamiento del modelo seleccionado

Fuente: Elaboración propia.

4.2.7 Análisis gráfico del rendimiento del modelo seleccionado

Para complementar los resultados cuantitativos del conjunto de prueba, se consideraron tres gráficas principales: la curva precisión-recall, la curva F1-confianza y la matriz de confusión normalizada. Estas representaciones permiten observar la relación entre precisión y recall, el comportamiento según el umbral de confianza y la distribución de aciertos por clase.

En la Figura 30 se observa que la curva precisión-recall evidencia un comportamiento alto y

estable en las clases evaluadas, con un mAP50 general de 0.989.

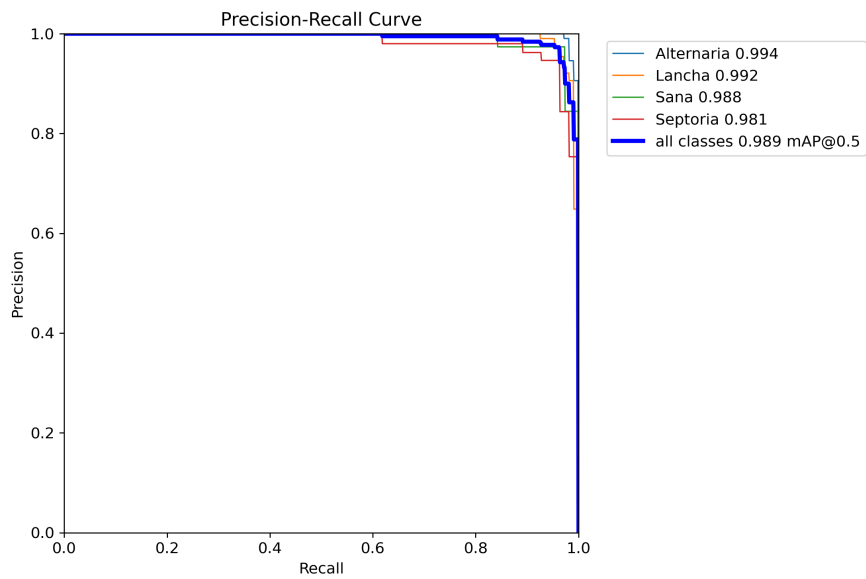


Figura 30.
Curva precisión-recall del modelo seleccionado

Fuente: Elaboración propia.

En la Figura 31 se observa que la curva F1-confianza obtuvo un F1 general de 0.96, con un umbral aproximado de 0.674. Este valor puede tomarse como referencia para calibrar la sensibilidad del diagnóstico.

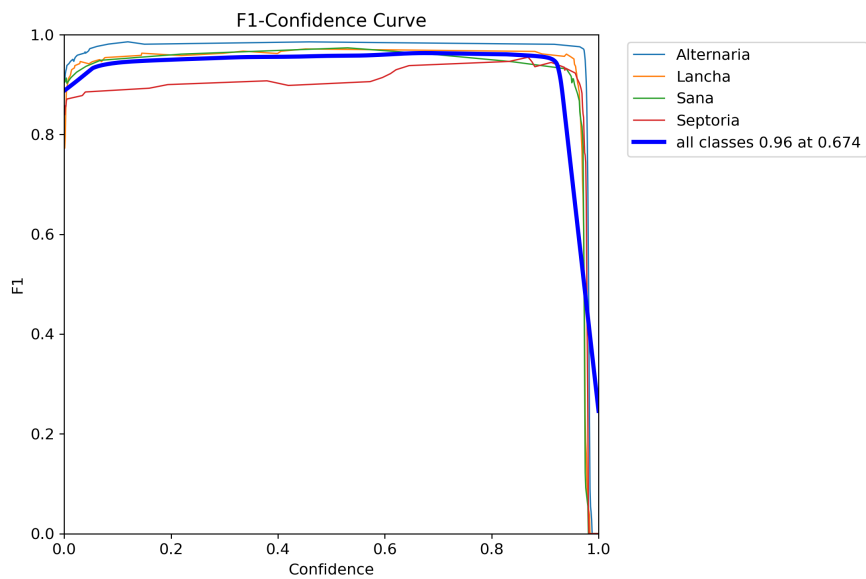


Figura 31.
Curva F1-confianza correspondiente al modelo seleccionado

Fuente: Elaboración propia.

En la Figura 32 se observa que la matriz de confusión normalizada muestra una concentración predominante en la diagonal principal, con pocas confusiones entre clases visualmente similares. Este comportamiento evidencia que el modelo logró diferenciar adecuadamente las categorías evaluadas, manteniendo una correspondencia consistente entre las etiquetas reales y las predicciones generadas.

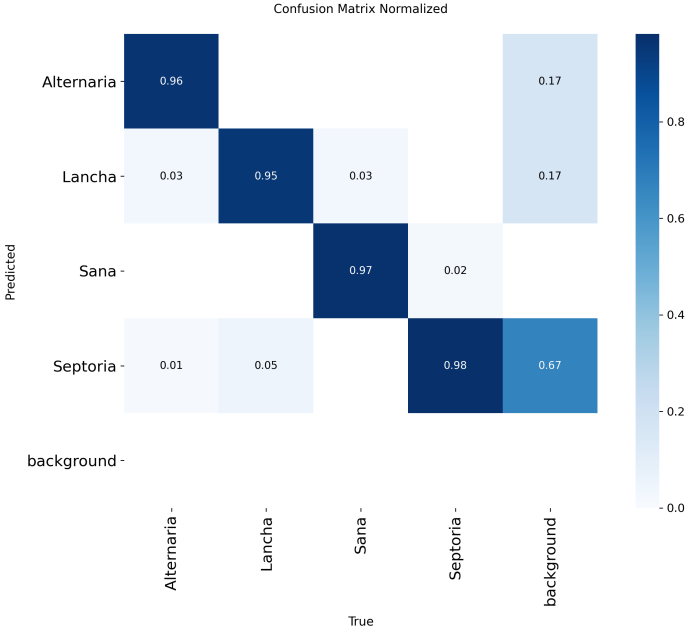


Figura 32.
Matriz de confusión normalizada del modelo seleccionado

Fuente: Elaboración propia.

4.2.8 Análisis visual de predicciones

En este apartado se presentan ejemplos representativos del conjunto de prueba, con el fin de visualizar la salida generada por el modelo seleccionado. Cada caso muestra la clase predicha y la caja delimitadora correspondiente sobre la hoja analizada.

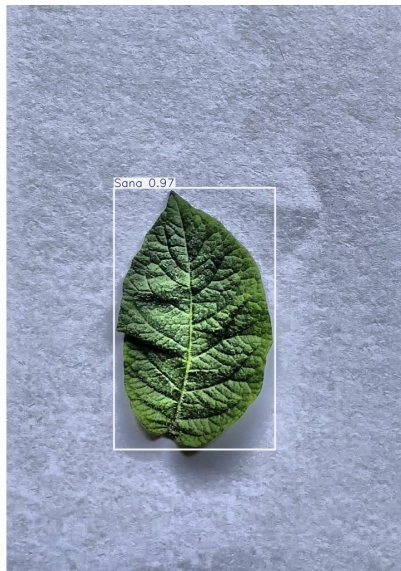
En la Figura 33 se observa el comportamiento del modelo frente a distintas condiciones visuales presentes en las imágenes evaluadas.



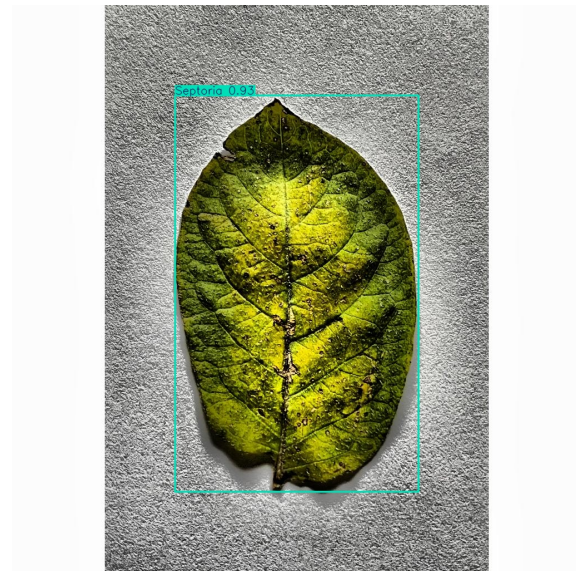
a) Alternaria



b) Lancha



c) Hoja sana



d) Septoria

Figura 33.

Ejemplos de predicción del modelo seleccionado en el conjunto de prueba

Fuente: Elaboración propia.

4.2.9 Interpretabilidad visual mediante EigenCAM

Se aplicó EigenCAM al modelo seleccionado con el propósito de analizar las regiones de la imagen que influyeron en la predicción. Esta técnica permitió verificar si el modelo concentraba su atención en la superficie foliar y en los patrones visuales asociados a la enfermedad, y evitar

que la respuesta dependiera principalmente del fondo o de elementos externos a la hoja.

En la Figura 34 se observa que las zonas de mayor activación se ubican sobre áreas con lesiones, manchas y cambios de coloración. Este comportamiento respalda los resultados obtenidos por el modelo, ya que evidencia que la predicción se relaciona con características visuales propias de la enfermedad y no únicamente con información contextual de la imagen.

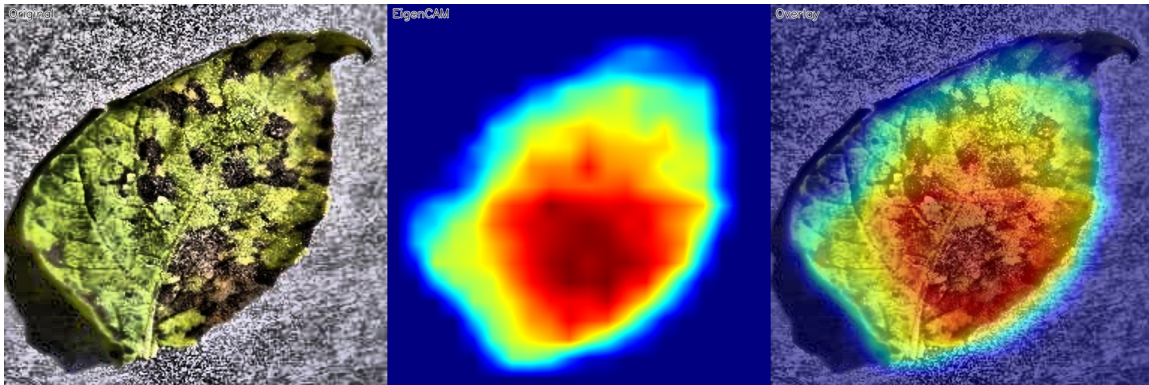


Figura 34.

Mapa de activación EigenCAM generado sobre una predicción del modelo seleccionado

Fuente: Elaboración propia.

4.2.10 Comparación con trabajos relacionados

Para contrastar el rendimiento obtenido, se consideró un trabajo reciente basado en detección de objetos aplicado a enfermedades foliares. La comparación se realizó con métricas propias de esta tarea, como mAP50, mAP50-95, recall y tiempo de inferencia, debido a que permiten analizar tanto la localización como la capacidad de detección del modelo.

La Tabla 66 presenta la comparación entre el modelo propuesto y el trabajo reciente [69], basado en YOLOv8 para la detección de enfermedades foliares. El contraste es directo. Se revisan la arquitectura empleada, la complejidad del modelo y las métricas principales de rendimiento.

Tabla 66.

Comparación del modelo seleccionado con un trabajo relacionado

Trabajo	Arquitectura	Parámetros	mAP50	mAP50-95	Recall	Inferencia
Aleinzi [69]	YOLOv8 ajustado	3.2 M	95.00 %	90.00 %	98.00 %	3–5 ms
Esta investigación	YOLO-26-N BoxGain	2.5 M	98.89 %	98.05 %	95.76 %	5.06 ms

Fuente: Elaboración propia.

Los resultados muestran que el modelo desarrollado alcanzó valores superiores en mAP50 y mAP50-95 frente al trabajo relacionado considerado. Además, aunque el tiempo de inferencia fue ligeramente mayor, el modelo mantiene una arquitectura ligera en términos de cantidad de parámetros y presenta una mayor calidad de localización en el conjunto de prueba.

4.2.11 Análisis final de los resultados del modelo de inteligencia artificial

En conjunto, los resultados obtenidos respaldan la selección del modelo YOLO-26-N con ajuste BoxGain y resolución de entrada de 1024 píxeles para su integración en CultivoScan. Esta configuración presentó un desempeño consistente en el conjunto de prueba, estabilidad frente a los resultados de validación, rendimiento favorable por clase y un tiempo de inferencia adecuado para su ejecución en un entorno móvil.

La configuración seleccionada mostró condiciones suficientes para apoyar el diagnóstico de enfermedades foliares en hojas de papa desde la aplicación móvil. Su capacidad de detección, precisión de localización y viabilidad técnica se respaldaron con las curvas de rendimiento, la matriz de confusión, los ejemplos visuales de predicción y el análisis complementario mediante EigenCAM.

CONCLUSIONES

A partir del desarrollo de la aplicación móvil, la construcción del conjunto de datos, el entrenamiento de los modelos de detección y la evaluación realizada con usuarios, se establecen las siguientes conclusiones:

1. La revisión teórica permitió sustentar el uso de inteligencia artificial, visión por computadora y modelos de detección de objetos como una alternativa viable para apoyar el diagnóstico de enfermedades foliares en cultivos de papa. Este fundamento permitió orientar la investigación hacia una necesidad concreta del sector agrícola, donde la identificación visual de enfermedades depende en gran medida de la experiencia del productor y de la disponibilidad de asistencia técnica.
2. La aplicación móvil CultivoScan permitió integrar el seguimiento agrícola con el diagnóstico visual de enfermedades en hojas de papa dentro de un mismo entorno funcional. La adopción de Scrum como metodología de desarrollo favoreció una construcción ordenada e incremental del sistema, permitiendo transformar los requerimientos definidos en una solución operativa para el usuario final.
3. El proceso de construcción del modelo de inteligencia artificial se organizó mediante CRISP-DM, lo que permitió mantener una secuencia metodológica desde la comprensión del problema hasta el despliegue del modelo en la aplicación móvil. Como resultado, se consolidó un conjunto de datos híbrido compuesto por imágenes públicas y fotografías propias recolectadas en campo, organizado para tareas de detección de objetos a nivel de hoja.
4. La evaluación experimental permitió identificar al modelo YOLO-26-N con ajuste Box-Gain y resolución de entrada de 1024 píxeles como la configuración con mejor rendimiento en el conjunto de prueba. Este modelo obtuvo un mAP50 de 98.89 %, un mAP50-95 de 98.05 %, una precisión de 96.99 % y un recall de 95.76 %, mostrando un desempeño alto en la detección de Alternaria, Lancha, Septoria y Hoja sana.

5. Los resultados por clase, las curvas de rendimiento y la matriz de confusión normalizada evidenciaron una respuesta consistente del modelo frente a las categorías evaluadas. De manera complementaria, el análisis mediante EigenCAM mostró que las zonas de mayor activación se concentraron principalmente sobre la superficie foliar y las regiones con síntomas visibles, lo que respalda que la predicción estuvo relacionada con patrones propios de la enfermedad y no únicamente con elementos del fondo.
6. En el contexto agrícola, los usuarios valoraron positivamente la aplicación móvil. La eficacia percibida tuvo una media general de 4.77 sobre 5. La eficiencia percibida fue ligeramente mayor, con 4.78 sobre 5. Con estos valores, la aplicación fue considerada útil, rápida y comprensible para apoyar el diagnóstico de enfermedades en hojas de papa.

RECOMENDACIONES

Las recomendaciones se plantean a partir de lo observado durante el desarrollo, entrenamiento, despliegue y evaluación de CultivoScan. Su propósito es orientar mejoras futuras del sistema, tanto en el modelo de inteligencia artificial como en la aplicación móvil, considerando las condiciones reales del sector agrícola y la experiencia obtenida con los usuarios participantes. En ese sentido, se presentan acciones que podrían fortalecer la continuidad técnica del proyecto:

1. **Extensión de las clases diagnósticas del modelo:** Para futuras versiones, es pertinente ampliar progresivamente las categorías evaluadas, incorporando otras enfermedades o afectaciones relevantes del cultivo de papa que no formaron parte del alcance inicial de esta investigación. Esta ampliación debería realizarse con el acompañamiento de especialistas agrónomos, especialmente durante la recolección, validación y anotación de las imágenes, con el fin de garantizar la calidad técnica del conjunto de datos y la confiabilidad de las nuevas clases incorporadas.
2. **Fortalecimiento y mantenimiento de la aplicación móvil:** CultivoScan debe mantenerse en un proceso de actualización continua, considerando la corrección de posibles fallos, la compatibilidad con nuevas versiones de Android y la incorporación progresiva de mejoras según las observaciones de los usuarios. Además, conforme la aplicación evolucione y aumente su alcance, será necesario fortalecer aspectos relacionados con la seguridad, privacidad y protección de los datos de los agricultores, especialmente al tratarse de una plataforma que gestiona información asociada con cuentas de usuario, parcelas y diagnósticos agrícolas.
3. **Optimización del modelo para entornos móviles:** La optimización del modelo debe mantenerse como una línea de mejora continua, considerando el tamaño del archivo, el tiempo de inferencia, el consumo de recursos y la compatibilidad con dispositivos Android de diferentes gamas. Este enfoque permitiría fortalecer la experiencia de uso en contextos rurales, donde la conectividad, el rendimiento del dispositivo o la disponibilidad de recursos pueden ser limitados.

4. **Evaluación continua con usuarios del sector agrícola:** Es importante mantener procesos de evaluación con agricultores, técnicos e ingenieros agrónomos en distintos contextos productivos, con el fin de conocer cómo se comporta la aplicación durante su uso real y qué aspectos requieren ajustes conforme varían las condiciones del cultivo. Esta retroalimentación permitiría fortalecer la precisión funcional del sistema, mejorar la interpretación de los resultados y orientar futuras actualizaciones de acuerdo con las necesidades reales del agricultor.

BIBLIOGRAFÍA

- [1] A. Villavicencio et al., “Sustainable Potato Production in the Mountain Area of Ecuador, an Approach to Increase Productivity with Small Scale Farmers,” *Agricultural Sciences*, vol. 13, n.º 10, págs. 1080-1090, oct. de 2022, ISSN: 2156-8553. DOI: 10.4236/as.2022.1310066. dirección: <https://www.scirp.org/journal/paperinformation?paperid=120472>.
- [2] H. A. Saragosin Lasluisa, “Revisión bibliográfica de las principales plagas y enfermedades del cultivo de papa (*Solanum tuberosum*) en la provincia de Cotopaxi en el año 2023,” Universidad Técnica de Cotopaxi, Latacunga, inf. téc., ago. de 2023. dirección: <https://repositorio.utc.edu.ec/items/e599338d-1351-47a2-9ef0-20d5da3d74aa>.
- [3] F. Basantes, J. P. Aragón Suárez, L. M. Albuja Illescas y L. d. R. Vásquez Hernández, “Diagnóstico de la situación actual de la producción y comercialización de la papa (*Solanum tuberosum* L.) en la Zona 1 del Ecuador,” *e-Agronegocios*, vol. 6, n.º 2, págs. 103-120, 2020, ISSN: 2215-3462. DOI: 10.18845/ea.v6i2.5103. dirección: <https://revistas.tec.ac.cr/index.php/eagronegocios/article/view/5103>.
- [4] H. J. Carranza Santos, “Diseño e implementación de un sistema IOT para la detección temprana de enfermedades en cultivos de papa mediante el procesamiento de imágenes,” Escuela Superior Politécnica de Chimborazo, Riobamba, inf. téc., nov. de 2024. dirección: <http://dspace.espoch.edu.ec/handle/123456789/22961>.
- [5] R. E. Grande y M. N. Ibarra Bonilla, “Red neuronal convolucional (CNN) de reconocimiento de plantas de maíz para un sistema de visión artificial,” *Investigación Aplicada, Un Enfoque en la Tecnología*, págs. 375-382, dic. de 2023, ISSN: 3594-035X.
- [6] V. A. González Huitrón, J. A. Medrano Hermosillo, A. E. Rodríguez Mata, J. A. Almaráz Damián y P. C. Ramírez, “Detección de enfermedades foliares en maíz mediante aprendizaje profundo,” *Komputer Sapiens*, vol. 17, págs. 40-58, ene. de 2025, ISSN: 2007-0691. dirección: https://www.researchgate.net/publication/387877416_Deteccion_de_enfermedades_foliares_en_maiz_mediante_aprendizaje_profundo.

- [7] M. O. Cordero Garcés, G. N. Parrales Cantos, D. A. Cobos Lucio, D. D. Carvajal Rivadeneira, L. A. Ceballos Baquerizo y A. M. Plúa Ponce, “Proyectos de Vinculación: Estrategias de Transferencia de Tecnología y conocimiento para el Desarrollo Rural Sostenible,” *inf. téc.*, mar. de 2025, págs. 1-71. dirección: <https://www.runaiki.es/index.php/runaiki/article/view/119>.
- [8] B. A. Saragosin Guamushig y A. R. Gavilánez Guanoluisa, “Desarrollo de un prototipo para la identificación automática de plagas y enfermedades en el cultivo de papa utilizando técnicas de inteligencia artificial en la ciudad de Latacunga,” Universidad Técnica de Cotopaxi, Latacunga, *inf. téc.*, mar. de 2024, pág. 118. dirección: <http://repositorio.utc.edu.ec/handle/27000/12029>.
- [9] F. Rodríguez Dávila, “Desarrollo de un sistema de visión artificial remoto para la detección de antracnosis o mancha foliar en los cultivos de fresa,” Universidad de Pamplona, Pamplona, Colombia, *inf. téc.*, dic. de 2021.
- [10] I. Miranda Cabrera, R. Alfonso de la Cruz, L. Guerra Arzuaga, S. Cuello, M. Bianchini y D. Pineda Medina, “A Mobile App for Detecting Potato Crop Diseases,” *Journal of Imaging*, vol. 10, n.º 2, pág. 47, 2024, ISSN: 2313-433X. DOI: 10.3390/jimaging10020047. dirección: <https://www.mdpi.com/2313-433X/10/2/47>.
- [11] F. D. Valle-Medina, L. J. Castillo-Heredia, M. Correa-Peralta y J. E. Guzman Seraquive, “Clasificación de enfermedades en hojas de papa utilizando Transformadores de Visión,” *Revista Digital Novasinergia*, vol. 8, n.º 1, págs. 142-156, ene. de 2025.
- [12] L. E. Sánchez Palacios, F. R. Martínez Alcívar, S. T. Torres Sánchez, A. C. Lascano Montes y G. N. Terán Guajala, “Agricultura de Precisión en el Ecuador,” *Ciencia Latina Revista Científica Multidisciplinar*, vol. 8, n.º 1, págs. 1532-1542, feb. de 2024, ISSN: 2707-2207. DOI: 10.37811/cl_rcm.v8i1.9547. dirección: https://www.researchgate.net/publication/378048176_Agricultura_de_Precision_en_El_Ecuador.
- [13] GeoPard Tech, *Understanding The Cycle of Precision Agriculture*, GeoPard Tech Blog, Infografía, dic. de 2023. dirección: <https://geopard.tech/blog/understanding-the-cycle-of-precision-agriculture-for-maximum-crop-potential/>.

- [14] J. Mendes et al., “Smartphone Applications Targeting Precision Agriculture Practices: A Systematic Review,” *Agronomy*, vol. 10, n.º 6, pág. 855, jun. de 2020, ISSN: 2073-4395. DOI: 10.3390/agronomy10060855. dirección: <https://www.mdpi.com/2073-4395/10/6/855>.
- [15] A. Siddiqua, M. A. Kabir, T. Ferdous, I. B. Ali y L. A. Weston, “Evaluating Plant Disease Detection Mobile Applications: Quality and Limitations,” *Agronomy*, vol. 12, n.º 8, pág. 1869, ago. de 2022, ISSN: 2073-4395. DOI: 10.3390/agronomy12081869. dirección: <https://www.mdpi.com/2073-4395/12/8/1869>.
- [16] I. Pacal et al., “A Systematic Review of Deep Learning Techniques for Plant Diseases,” *Artificial Intelligence Review*, vol. 57, n.º 11, nov. de 2024, ISSN: 1573-7462. DOI: 10.1007/s10462-024-10944-7. dirección: <https://link.springer.com/article/10.1007/s10462-024-10944-7>.
- [17] I. Miranda Cabrera et al., “Sistema para el diagnóstico de plagas de *Solanum tuberosum* L. mediante técnicas de inteligencia artificial,” *Revista de Protección Vegetal*, vol. 38, n.º 16, dic. de 2023, ISSN: 2224-4697. dirección: <https://cu-id.com/2247/v38e16>.
- [18] V. S. Dhaka, S. V. Meena, G. Rani, D. Sinwar, M. F. Ijaz y M. Woźniak, “A Survey of Deep Convolutional Neural Networks Applied for Prediction of Plant Leaf Diseases,” *Sensors*, vol. 21, n.º 14, pág. 4749, 2021.
- [19] J. Lu, L. Tan y H. Jiang, “Review on Convolutional Neural Network (CNN) Applied to Plant Leaf Disease Classification,” *Agriculture*, vol. 11, n.º 8, pág. 707, ago. de 2021, ISSN: 2077-0472. DOI: 10.3390/agriculture11080707. dirección: <https://www.mdpi.com/2077-0472/11/8/707>.
- [20] J. Pasalkar, G. Gorde, C. More, S. Memane y V. Gaikwad, “Potato Leaf Disease Detection Using Machine Learning,” en *CEUR Workshop Proceedings*, vol. 3283, CEUR-WS, nov. de 2023, págs. 949-954. DOI: 10.12944/carj.11.3.23. dirección: https://www.researchgate.net/publication/377370861_Potato_Leaf_Disease_Detection_Using_Machine_Learning.

- [21] A. Bhargava, A. Shukla, O. P. Goswami, M. H. Alsharif, P. Uthansakul y M. Uthansakul, "Plant Leaf Disease Detection, Classification, and Diagnosis Using Computer Vision and Artificial Intelligence: A Review," *IEEE Access*, vol. 12, págs. 37 443-37 469, 2024. DOI: 10.1109/ACCESS.2024.3373001. dirección: <https://doi.org/10.1109/ACCESS.2024.3373001>.
- [22] C. M. Badgujar, A. Poulosé y H. Gan, "Agricultural Object Detection with You Look Only Once (YOLO) Algorithm: A Bibliometric and Systematic Literature Review," *arXiv preprint*, ene. de 2024. DOI: 10.48550/arXiv.2401.10379. dirección: <https://arxiv.org/abs/2401.10379>.
- [23] K. N. Rahman, S. C. Banik, R. Islam y A. Al Fahim, "A Real-Time Monitoring System for Accurate Plant Leaves Disease Detection Using Deep Learning," *Crop Design*, vol. 4, n.º 1, feb. de 2025, ISSN: 2772-8994. DOI: 10.1016/j.crope.2024.100092. dirección: <https://doi.org/10.1016/j.crope.2024.100092>.
- [24] G. Jocher, A. Chaurasia y J. Qiu, *Ultralytics YOLOv8*, Repositorio oficial de GitHub de Ultralytics, ver. 8.0.0, Licencia: AGPL-3.0, ene. de 2023. dirección: <https://github.com/ultralytics/ultralytics>.
- [25] S. A. Salloum, T. Gaber, S. Vadera y K. Shaalan, "A Systematic Literature Review on Phishing Email Detection Using Natural Language Processing Techniques," *IEEE Access*, vol. 10, págs. 65 703-65 727, 2022. DOI: 10.1109/ACCESS.2022.3183083. dirección: <https://doi.org/10.1109/ACCESS.2022.3183083>.
- [26] S. B. Chaudhari, V. Wagaskar, M. Shaikh, V. Shirsath y O. Shelke, "Plant Disease Detection Using TensorFlow," *IJISSET – International Journal of Innovative Science, Engineering & Technology*, vol. 8, págs. 350-356, abr. de 2021, ISSN: 2348-7968. dirección: https://ijiset.com/vol8/v8s4/IJISSET_V8_I04_31.pdf.
- [27] D. Goklani, *Real-Time Plant Disease Detection Using Mobile Device with TensorFlow Lite and Flutter*, SSRN Electronic Journal (Preprint), Accessed: Jun. 30, 2025, ago. de 2024. dirección: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4921827.

- [28] O. Debauche, S. Mahmoudi, P. Manneback y F. Lebeau, “Cloud and Distributed Architectures for Data Management in Agriculture 4.0: Review and Future Trends,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, n.º 9, págs. 7494-7514, oct. de 2022.
- [29] A. M. Shimaoka, R. C. Ferreira y A. Goldman, “The Evolution of CRISP-DM for Data Science: Methods, Processes and Frameworks,” *SBC Reviews on Computer Science*, vol. 4, n.º 1, págs. 28-43, oct. de 2024. DOI: 10.5753/reviews.2024.3757. dirección: <https://www.researchgate.net/publication/385280269>.
- [30] J. S. Saltz y R. Heckman, “Integrating Agile and CRISP-DM Methodologies for Data Science Projects,” en *Proceedings of the 54th Hawaii International Conference on System Sciences (HICSS)*, ScholarSpace, ene. de 2021, págs. 1-10.
- [31] J. R. Macas Bermeo, “Desarrollo de una Aplicación para la Detección de Plagas en el Cultivo de Papas Aplicando Redes Neuronales Convolucionales,” Universidad Técnica de Cotopaxi, inf. téc., sep. de 2022, Accessed: Jun. 23, 2025. dirección: <https://repoadmin.utc.edu.ec/items/0e4d70eb-d2e5-488d-9339-0f29b22d4f1a>.
- [32] R. Kyamelia, R. Subharthy, P. T. Kumar y S. Chaudhuri Sheli, “Potato Plant Leaf Disease Detection Using Custom CNN Deep Net: A Step Towards Sustainable Agriculture,” en *2024 ITU Kaleidoscope: Innovation and Digital Transformation for a Sustainable World*, Institute of Electrical y Electronics Engineers Inc., 2024. DOI: 10.23919/ITUK62727.2024.10772791. dirección: <https://ieeexplore.ieee.org/document/10772791>.
- [33] T. A. Dame, G. B. Adera y D. W. Girmaw, “Deep Learning-Based Potato Leaf Disease Classification and Severity Assessment,” *Discover Applied Sciences*, vol. 7, n.º 7, jul. de 2025, ISSN: 3004-9261. DOI: 10.1007/s42452-025-07240-6. dirección: <https://link.springer.com/article/10.1007/s42452-025-07240-6>.
- [34] F. D. Valle-Medina, L. J. Castillo-Heredia, M. A. Correa-Peralta y J. E. Guzmán-Seraquive, “Clasificación de Enfermedades en Hojas de Papa Utilizando Transformadores de Visión,” *Novasinerгия Revista Digital de Ciencia, Ingeniería y Tecnología*, vol. 8, n.º 1,

- págs. 142-156, ene. de 2025. DOI: 10 . 37135 / ns . 01 . 15 . 06. dirección: <https://novasinerugia.unach.edu.ec/index.php/novasinerugia/article/view/562>.
- [35] J. H. Sinamenye, A. Chatterjee y R. Shrestha, “Potato Plant Disease Detection: Leveraging Hybrid Deep Learning Models,” *BMC Plant Biology*, vol. 25, n.º 1, pág. 647, mayo de 2025, ISSN: 1471-2229. DOI: 10 . 1186 / s12870 - 025 - 06679 - 4. dirección: <https://bmcplantbiol.biomedcentral.com/articles/10.1186/s12870-025-06679-4>.
- [36] S. M. Alhammad, D. S. Khafaga, W. M. El-Hady, F. M. Samy y K. M. Hosny, “Deep Learning and Explainable AI for Classification of Potato Leaf Diseases,” *Frontiers in Artificial Intelligence*, vol. 7, feb. de 2025, ISSN: 2624-8212. DOI: 10 . 3389 / frai . 2024 . 1449329. dirección: <https://www.frontiersin.org/articles/10.3389/frai.2024.1449329>.
- [37] D. Herrera-Poyatos et al., “Small Data Deep Learning Methodology for In-Field Disease Detection,” *arXiv preprint*, págs. 1-9, sep. de 2024. DOI: 10 . 48550 / arXiv . 2409 . 17119. dirección: <https://arxiv.org/abs/2409.17119>.
- [38] K. Schwaber y J. Sutherland, “The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game,” Scrum.org, Boston, MA, USA, inf. téc., nov. de 2020, págs. 1-14. dirección: <https://scrumguides.org/scrum-guide.html>.
- [39] Y. Ardilla y A. K. Hakim, “Digitalization of Correspondence Information Systems Using Scrum Model,” *Journal La Multiapp*, vol. 6, n.º 1, págs. 23-33, ene. de 2025, ISSN: 2716-3865. DOI: 10 . 37899 / journallamultiapp . v6i1 . 1814. dirección: <https://doi.org/10.37899/journallamultiapp.v6i1.1814>.
- [40] S. Qazi, M. A. Shah y A. Khan, “A Comparative Study of Agile Estimation Techniques: Planning Poker, T-Shirt Sizing, and Dot Voting,” *Journal of Software Engineering and Applications*, vol. 17, n.º 3, págs. 55-68, mar. de 2024.
- [41] N. Kimseng, D. A. Kurnia, I. Vuthy, R. W. Arifin y D. Setiyadi, “UI/UX Development Using Figma based on Inclusive Design,” *JINAV: Journal of Information and Visualiza-*

- tion, vol. 4, n.º 2, págs. 227-234, ene. de 2024. DOI: 10.35877/454RI.jinav2257. dirección: <https://doi.org/10.35877/454RI.jinav2257>.
- [42] K. Milojković, M. Živković y N. Bačanin Džakula, “Agile Multi-user Android Application Development With Firebase: Authentication, Authorization, and Profile Management,” en *Sinteza 2024 - International Scientific Conference on Information Technology, Computer Science, and Data Science*, Belgrade, Serbia: Singidunum University, 2024, págs. 405-412. DOI: 10.15308/Sinteza-2024-405-412. dirección: <https://portal.sinteza.singidunum.ac.rs/paper/1005>.
- [43] N. Ahmad, S. Singh, M. F. AlAjmi, A. Hussain y K. Raza, “CropGCNN: color space-based crop disease classification using group convolutional neural network,” *PeerJ Computer Science*, vol. 10, e2136, jul. de 2024. DOI: 10.7717/peerj-cs.2136. dirección: <https://doi.org/10.7717/peerj-cs.2136>.
- [44] N. Jiménez, S. Orellana, B. Mazon-Olivo, W. Rivas-Asanza e I. Ramírez-Morales, “Detection of Leaf Diseases in Banana Crops Using Deep Learning Techniques,” *AI*, vol. 6, n.º 3, pág. 61, mar. de 2025. DOI: 10.3390/ai6030061. dirección: <https://doi.org/10.3390/ai6030061>.
- [45] M. Thalor, M. Pathak, R. Jamadar y S. Khan, “Potato Leaf Insight: An AI-Driven System for Detection of Potato Leaf Disease,” *Current Agriculture Research Journal*, vol. 12, n.º 3, oct. de 2024. DOI: 10.12944/CARJ.12.3.09. dirección: <http://dx.doi.org/10.12944/CARJ.12.3.09>.
- [46] G. Sangar y V. Rajasekar, “Optimized classification of potato leaf disease using EfficientNet-LITE and KE-SVM in diverse environments,” *Frontiers in Plant Science*, vol. 16, pág. 1499909, mayo de 2025. DOI: 10.3389/fpls.2025.1499909. dirección: <https://doi.org/10.3389/fpls.2025.1499909>.
- [47] Y. Zhang, H. Wang y C. Li, “Plant Disease Identification under Imbalanced Dataset using Deep Learning Methods,” *Journal of Agricultural Informatics*, vol. 15, n.º 2, págs. 45-58, nov. de 2024.

- [48] Y. Chen, L. Wang y H. Zhang, "A novel efficient disease detection method with multi-scale learning and edge feature enhancement," *Frontiers in Plant Science*, vol. 16, pág. 1 666 955, 2025. DOI: 10.3389/fpls.2025.1666955.
- [49] R. Kumar, A. Singh y V. Sharma, "Impact of dataset quality and pre-processing on deep learning models for plant disease recognition," *Computers and Electronics in Agriculture*, vol. 228, pág. 109 123, ene. de 2025. DOI: 10.1016/j.compag.2024.109123.
- [50] D. Liu, P. He, Q. Wang, Y. He y F. Cheng, "Object detection with attribute tagging task: model design and evaluation on agricultural datasets," *Computers and Electronics in Agriculture*, vol. 230, pág. 109 880, mar. de 2025. DOI: 10.1016/j.compag.2024.109880.
- [51] Y. Wu, Z. Chen y M. Li, "Data-centric annotation strategies for plant disease detection: balancing consistency and model performance," *Precision Agriculture*, vol. 25, n.º 1, págs. 112-128, feb. de 2024.
- [52] V. Singh y A. Sharma, "Robust deep learning methodologies for plant disease identification: Handling data augmentation and leakage," *Ecological Informatics*, vol. 80, pág. 102 500, mayo de 2024. DOI: 10.1016/j.ecoinf.2024.102500.
- [53] E. Rachmawati, A. Kurniawan y A. Setyanto, "Enhancing Deep Learning-based Classification of Leaf Diseases using CLAHE and Data Augmentation," *Journal of Agricultural Informatics*, vol. 16, n.º 1, págs. 23-35, abr. de 2025.
- [54] K. Patel, C. Bhatt y A. Singh, "Optimal data augmentation strategies for robust plant leaf disease recognition under real-field conditions," *Artificial Intelligence in Agriculture*, vol. 11, pág. 100 050, ago. de 2024.
- [55] A. Nkonjoh et al., "YOLO-AgriNet: A Deep Learning-Based Model for Real-Time Plant Disease Detection in Precision Agriculture," *Journal of Computer and Communications*, vol. 13, págs. 181-204, ago. de 2025. DOI: 10.4236/jcc.2025.138009.
- [56] X. Chen, Y. Zhang y H. Wu, "Performance and Reproducibility Analysis of Local GPU-based Training for Agricultural Computer Vision," *IEEE Access*, vol. 13, págs. 5241-5256, ene. de 2025.

- [57] L. Zhao, S. Kim y A. Kumar, “Balancing accuracy and efficiency: Comparative analysis of lightweight YOLO and RT-DETR models for mobile edge deployment in crop disease detection,” *Computers and Electronics in Agriculture*, vol. 232, pág. 110 245, mayo de 2025. DOI: 10.1016/j.compag.2025.110245.
- [58] J. Wang, S. Liu e Y. Chen, “Optimizing YOLO architectures through dynamic data augmentation and hyperparameter tuning for robust agricultural object detection,” *Expert Systems with Applications*, vol. 245, pág. 123 011, jul. de 2024.
- [59] M. Li, C. Zhang y K. Wang, “Systematic hyperparameter optimization and ablation studies for deploying lightweight YOLO architectures in mobile agricultural applications,” *Smart Agricultural Technology*, vol. 9, pág. 100 450, oct. de 2024. DOI: 10.1016/j.atech.2024.100450.
- [60] H. Zhou, Y. Sun y X. Wang, “Regularization strategies and early stopping for preventing overfitting in YOLO-based agricultural disease detection,” *Computers and Electronics in Agriculture*, vol. 225, pág. 108 520, sep. de 2024.
- [61] A. Gomez, C. Martinez y L. Silva, “Methodological standards for unbiased evaluation and benchmarking of deep learning models in plant disease detection,” *Smart Agricultural Technology*, vol. 8, pág. 100 395, ago. de 2024.
- [62] M. Hasan, S. Islam y A. Rahman, “Evaluating the rigor of object detection metrics in precision agriculture: Why mAP50-95 is essential for disease localization,” *Ecological Informatics*, vol. 82, pág. 102 640, sep. de 2024. DOI: 10.1016/j.ecoinf.2024.102640.
- [63] D. Kim, J. Lee y S. Park, “Assessing generalization capability and stability of deep learning models for plant disease detection via rigorous dataset partitioning,” *Precision Agriculture*, vol. 25, n.º 5, págs. 2104-2122, nov. de 2024.
- [64] R. Sharma, A. Gupta y V. Kumar, “The accuracy-latency trade-off: Evaluating the technical viability of object detection models for mobile edge deployment in agriculture,” *Computers and Electronics in Agriculture*, vol. 240, pág. 110 488, feb. de 2026.

- [65] R. Silva, M. Oliveira y J. Santos, “Evaluating the impact of INT8 quantization versus FP32 precision on YOLO edge deployments for fine-grained plant disease detection,” *Computers and Electronics in Agriculture*, vol. 235, pág. 110-120, jul. de 2025. DOI: 10.1016/j.compag.2025.110120.
- [66] F. Mendoza, J. Valdes y R. Hernandez, “Robust plant disease lesion segmentation under varying illumination conditions using HSV color space analysis,” *Computers and Electronics in Agriculture*, vol. 218, pág. 108-150, abr. de 2024. DOI: 10.1016/j.compag.2024.108650.
- [67] C. Alvarez, M. Rojas y T. Fernandez, “Evaluating the quality in use of agricultural mobile applications: A framework based on efficacy and efficiency metrics for field diagnostics,” *Software Quality Journal*, vol. 33, págs. 45-62, ene. de 2025.
- [68] ISO/IEC, “ISO/IEC 25010:2023 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model,” International Organization for Standardization, Geneva, Switzerland, Standard, 2023.
- [69] M. Aleinzi, “Enhancing Precision Agriculture with YOLOv8: A Deep Learning Approach to Potato Disease Identification,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 16, n.º 4, 2025. DOI: 10.14569/IJACSA.2025.01604110. dirección: <http://dx.doi.org/10.14569/IJACSA.2025.01604110>.

ANEXOS

Anexo A. Instrumento de evaluación de la aplicación móvil

El formulario de evaluación fue elaborado en Microsoft Forms con el propósito de recopilar información sobre la experiencia de uso de la aplicación móvil CultivoScan. El instrumento se organizó en cuatro secciones: datos generales del encuestado, evaluación de la precisión del sistema, evaluación de la eficiencia de la aplicación y comentarios abiertos sobre la percepción general de uso.

Las preguntas permitieron caracterizar el perfil de los participantes, valorar la precisión del diagnóstico, analizar la rapidez del proceso de detección y recoger sugerencias de mejora. Esta estructura facilitó la obtención de información cuantitativa mediante una escala tipo Likert y de observaciones cualitativas a partir de las respuestas abiertas. En la Figura 35 se presenta la sección inicial del formulario, destinada a recopilar los datos generales de los participantes.

Sección 1

Datos generales del encuestado

La siguiente sección tiene como finalidad recopilar información general del encuestado, con el fin de caracterizar el perfil de los usuarios que participan en la evaluación de la aplicación móvil.

1. ¿Cuál es su cargo o rol que desempeña? *

Selecciona la respuesta

2. ¿Cuántos años de experiencia tiene en el área agrícola? *

Menos de 1 año

Entre 1 y 3 años

Entre 4 y 7 años

Más de 7 años

3. ¿Cuál es su nivel de experiencia en la identificación de enfermedades en cultivos de papa? *

Selecciona la respuesta

4. ¿Con qué frecuencia realiza diagnósticos de enfermedades en campo? *

Ocasionalmente

Mensualmente

Semanalmente

Diariamente

Figura 35.
Sección de datos generales del instrumento de evaluación

Fuente: Elaboración propia.

En la Figura 36 se muestra la sección orientada a evaluar la precisión del sistema en el diagnóstico de enfermedades en hojas de papa.

Sección 2

Evaluación de la precisión del sistema

En esta sección se evaluará qué tan precisa es la aplicación móvil CultivoScan al identificar correctamente las enfermedades presentes en las hojas de papa.

5. Percepción sobre la precisión de la aplicación *

	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
¿El diagnóstico proporcionado por la aplicación coincide con lo observado visualmente en la hoja analizada?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿La aplicación identifica correctamente la diferencia entre una hoja sana y una hoja con enfermedad?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿La enfermedad detectada por la aplicación corresponde al tipo de afección presente en la hoja (lancha, septoria o alternaria)?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿Los resultados de la aplicación son consistentes cuando se analizan hojas con características similares?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿La información presentada por la aplicación permite identificar correctamente la enfermedad detectada?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿Considera que la aplicación proporciona diagnósticos confiables para apoyar la identificación de enfermedades en campo?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿Las áreas señaladas por la aplicación corresponden a las zonas afectadas de la hoja?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura 36.
Sección de evaluación de la precisión del sistema

Fuente: Elaboración propia.

En la Figura 37 se presenta la sección utilizada para valorar la eficiencia de la aplicación durante el procesamiento de imágenes.

Sección 3

Evaluación de la eficiencia de la aplicación

En esta sección se evaluará qué tan eficiente es la aplicación móvil Agro Intel al procesar imágenes y mostrar los resultados durante su uso en campo.

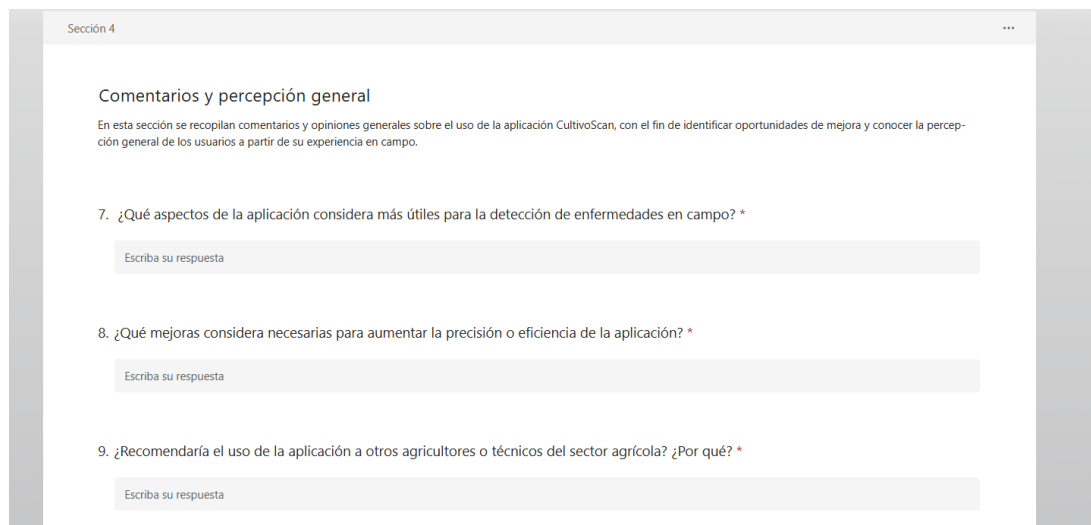
6. Percepción sobre la eficiencia de la aplicación *

	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
¿La aplicación muestra los resultados en un tiempo adecuado después de seleccionar o capturar la imagen?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿El proceso de análisis de la imagen se realiza de forma rápida y sin interrupciones?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿La aplicación funciona de manera fluida durante su uso en campo?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿La aplicación permite obtener un diagnóstico más rápido en comparación con los métodos tradicionales de identificación de enfermedades?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿Considera que la aplicación optimiza el tiempo dedicado a la identificación de enfermedades en cultivos de papa?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura 37.
Sección de evaluación de la eficiencia de la aplicación

Fuente: Elaboración propia.

En la Figura 38 se muestra la sección final destinada a recopilar comentarios y la percepción general de los participantes.



Sección 4

Comentarios y percepción general

En esta sección se recopilan comentarios y opiniones generales sobre el uso de la aplicación CultivoScan, con el fin de identificar oportunidades de mejora y conocer la percepción general de los usuarios a partir de su experiencia en campo.

7. ¿Qué aspectos de la aplicación considera más útiles para la detección de enfermedades en campo? *

Escriba su respuesta

8. ¿Qué mejoras considera necesarias para aumentar la precisión o eficiencia de la aplicación? *

Escriba su respuesta

9. ¿Recomendaría el uso de la aplicación a otros agricultores o técnicos del sector agrícola? ¿Por qué? *

Escriba su respuesta

Figura 38.

Sección de comentarios y percepción general

Fuente: Elaboración propia.

Anexo B. Matriz de respuestas de la encuesta

La matriz de respuestas presenta la codificación de los datos obtenidos mediante el formulario aplicado en Microsoft Forms a los 25 participantes. Esta información permite conservar la trazabilidad de las respuestas utilizadas para el análisis cuantitativo de la aplicación móvil, especialmente en las dimensiones de precisión del sistema y eficiencia de uso.

Para facilitar su lectura dentro del documento, las preguntas con escala tipo Likert fueron codificadas numéricamente de 1 a 5, manteniendo la equivalencia con las opciones originales del instrumento. La codificación empleada fue la siguiente: 1 = Totalmente en desacuerdo, 2 = En desacuerdo, 3 = Ni de acuerdo ni en desacuerdo, 4 = De acuerdo y 5 = Totalmente de acuerdo.

Los ítems E1–E7 corresponden a la evaluación de la precisión del sistema, mientras que los ítems F1–F5 pertenecen a la evaluación de la eficiencia de la aplicación móvil. Las respuestas abiertas no se incluyen en esta matriz, debido a que fueron sintetizadas en el análisis cualitativo de percepción general y oportunidades de mejora.

En la Tabla 67 se presenta la matriz codificada de respuestas tipo Likert obtenidas de los 25 participantes encuestados.

Tabla 67.

Matriz codificada de respuestas tipo Likert

Part.	E1	E2	E3	E4	E5	E6	E7	F1	F2	F3	F4	F5
P1	5	4	5	5	5	5	4	5	4	5	4	4
P2	5	5	5	4	5	4	5	5	4	5	4	5
P3	5	5	5	5	5	5	5	5	5	4	5	5
P4	4	4	4	4	4	4	5	5	5	4	4	4
P5	4	5	5	4	5	5	5	5	5	4	5	5
P6	4	5	5	4	5	4	5	5	4	5	4	5
P7	4	5	4	5	4	5	4	5	5	4	5	5
P8	4	5	4	5	4	5	4	5	4	5	4	5
P9	4	5	5	4	5	5	4	5	4	5	4	5
P10	4	5	4	5	4	5	5	5	4	5	4	5
P11	4	5	4	5	4	5	5	5	4	5	5	5
P12	4	5	4	5	4	5	5	5	4	5	5	5
P13	4	5	5	5	4	5	5	5	4	5	5	5
P14	4	5	5	5	5	5	5	4	5	5	5	4
P15	5	5	4	5	4	5	5	5	5	4	4	5
P16	5	5	5	5	5	5	5	5	5	5	5	5
P17	5	5	5	5	5	5	5	5	5	5	5	5
P18	5	5	5	5	5	5	5	5	5	5	5	5
P19	5	5	5	5	5	5	5	5	5	5	5	5
P20	5	5	5	5	5	5	5	5	5	5	5	5
P21	5	5	5	5	5	5	5	5	5	5	5	5
P22	5	5	5	5	5	4	5	5	5	5	4	5
P23	5	5	5	5	5	5	5	5	5	5	5	5
P24	5	5	5	5	5	5	5	5	5	5	5	5
P25	5	5	5	5	5	5	5	5	5	5	5	5

Fuente: Elaboración propia.

Anexo C. Publicación de la aplicación móvil en Google Play Store

La aplicación móvil CultivoScan fue publicada en Google Play Store como parte del producto tecnológico desarrollado en la investigación. Esta publicación respalda la disponibilidad de la aplicación para dispositivos Android al momento de la entrega del trabajo, mediante una fuente oficial de distribución. El enlace se presenta como evidencia de publicación y acceso a la ficha

oficial de la aplicación.

En la Tabla 68 se detallan los datos de disponibilidad de CultivoScan en Google Play Store, incluyendo la plataforma, el identificador del paquete, el alcance geográfico y el enlace oficial de acceso.

Tabla 68.

Publicación de la aplicación móvil en Google Play Store

Elemento	Descripción
Nombre de la aplicación	CultivoScan
Plataforma de publicación	Google Play Store
Sistema operativo	Android
Identificador del paquete	com.agrointel.app
Disponibilidad geográfica	Ecuador
Enlace oficial	https://play.google.com/store/apps/details?id=com.agrointel.app
Propósito	Respaldar la publicación y disponibilidad de la aplicación móvil desarrollada como producto tecnológico de la investigación.
Observación	El enlace corresponde a la ficha oficial disponible en Google Play Store al momento de la entrega del trabajo.

Fuente: Elaboración propia.

Anexo D. Repositorio de código y cuadernos de experimentación

El repositorio del proyecto reúne los cuadernos de Jupyter Notebook desarrollados en Python para documentar el proceso experimental del modelo de inteligencia artificial. Los archivos se encuentran en formato *.ipynb*, lo que facilita la conservación del código ejecutado, las salidas generadas y el historial técnico de cada etapa.

Este material respalda la trazabilidad del trabajo realizado, desde la configuración del entorno local hasta la exportación del modelo para su integración en la aplicación móvil. Además, permite revisar de forma ordenada los procedimientos aplicados durante el entrenamiento, la evaluación, la generación de métricas, las predicciones visuales y los mapas de activación EigenCAM.

En la Tabla 69 se presenta la información general del repositorio utilizado para respaldar el código y los cuadernos de experimentación.

Tabla 69.

Repositorio de código y cuadernos de experimentación

Elemento	Descripción
Repositorio	https://github.com/DiegoCuaycal/potato-leaf-vision.git
Formato de archivos	Cuadernos de Jupyter Notebook en formato <i>.ipynb</i> .
Contenido principal	Configuración de entorno, descarga del dataset, entrenamiento, validación, prueba, métricas, predicciones, EigenCAM y exportación del modelo.
Propósito	Respaldar la trazabilidad, revisión técnica y reproducibilidad del proceso experimental.

Fuente: Elaboración propia.

En la Tabla 70 se describe el contenido principal de cada cuaderno incluido en el repositorio.

Tabla 70.

Descripción general de los cuadernos de experimentación

Cuaderno	Descripción
Cuaderno 0	Configuración del entorno local, GPU, CUDA, PyTorch, Python y YOLO.
Cuaderno 1	Descarga y preparación inicial del conjunto de datos.
Cuaderno 2	Entrenamiento de los modelos de detección evaluados.
Cuaderno 3	Consolidación de resultados de los experimentos realizados.
Cuaderno 4	Generación del reporte técnico del mejor modelo experimental.
Cuaderno 5	Generación de mapas de activación EigenCAM.
Cuaderno 6	Verificación de la cantidad de imágenes utilizadas durante el entrenamiento.
Cuaderno 7	Obtención de resultados sobre el conjunto de prueba.
Cuaderno 8	Visualización de predicciones y gráficas con imágenes reales.
Cuaderno 9	Exportación del modelo para su integración en Flutter.
Cuaderno 10	Generación de gráficas y recursos visuales para documentación científica.

Fuente: Elaboración propia.

Anexo E. Publicación del conjunto de datos

El conjunto de datos utilizado en la investigación fue publicado en Zenodo con el propósito de garantizar su disponibilidad, trazabilidad y preservación académica. Esta publicación contiene las imágenes empleadas para el entrenamiento y evaluación del modelo, junto con sus anotaciones manuales mediante cajas delimitadoras.

El dataset está conformado por 3 060 imágenes distribuidas en cuatro clases: Septoria, Alternaria, Lancha y Hoja sana. Las imágenes provienen de dos fuentes: registros de campo capturados en cultivos de papa de la provincia del Carchi, Ecuador, y muestras complementarias del conjunto PlantVillage. Además, el conjunto fue preprocesado a una resolución de 1024×1024 píxeles, manteniendo un formato adecuado para tareas de detección de objetos.

En la Tabla 71 se presenta la información general de la publicación del conjunto de datos en Zenodo.

Tabla 71.

Publicación del conjunto de datos en Zenodo

Elemento	Descripción
Repositorio de datos	Zenodo
DOI	https://doi.org/10.5281/zenodo.20247346
Título del dataset	Annotated Dataset for Potato Foliar Disease and Healthy Leaf Detection
Número de imágenes	3 060 imágenes anotadas manualmente con cajas delimitadoras.
Clases consideradas	Septoria, Alternaria, Lancha y Hoja sana.
Formato publicado	Archivo comprimido en formato <i>.zip</i> .
Propósito	Respaldar la disponibilidad, trazabilidad y reutilización del conjunto de datos empleado en la investigación.

Fuente: Elaboración propia.