



UNIVERSIDAD LAICA “ELOY ALFARO” DE MANABÍ

EXTENSIÓN EL CARMEN

CARRERA EN INGENIERÍA EN SOFTWARE

Creada Ley No. 10 — Registro Oficial 313 de noviembre 13 de 1985

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

TEMA:

SISTEMA INFORMÁTICO CON ESQUEMAS POLIMÓRFICOS PARA EL
MODELADO DE DATOS FITOTÉCNICOS EN LA GRANJA EXPERIMENTAL
ULEAM EL CARMEN

AUTOR:

JOSE MANUEL ZAMORA LOOR

TUTOR:

ING. CARLOS VINICIO LÓPEZ RODRÍGUEZ

AÑO:

2026

CERTIFICACIÓN DEL TUTOR

 Uleam UNIVERSIDAD LAICA DE MANABÍ	NOMBRE DEL DOCUMENTO:	CÓDIGO: PAT-04-F-004
	CERTIFICADO DE TUTOR(A).	
	PROCEDIMIENTO: TITULACIÓN DE ESTUDIANTES DE GRADO BAJO LA UNIDAD DE INTEGRACIÓN CURRICULAR	REVISIÓN: 1 Página 1 de 1

CERTIFICACIÓN

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

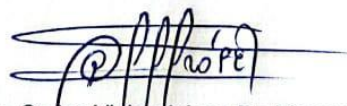
Haber dirigido, revisado y aprobado preliminarmente el Trabajo de Integración Curricular bajo la autoría del estudiante **JOSE MANUEL ZAMORA LOOR**, legalmente matriculado/a en la carrera de Tecnologías de la Información período académico 2026-1, cumpliendo el total de 288 horas, cuyo tema del proyecto o núcleo problémico es **"SISTEMA INFORMÁTICO CON ESQUEMAS POLIMÓRFICOS PARA EL MODELADO DE DATOS FITOTÉCNICOS EN LA GRANJA EXPERIMENTAL ULEAM EL CARMEN"**.

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

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

El Carmen, 31 de julio de 2026.

Lo certifico,



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

TRIBUNAL DE SUSTENTACIÓN



Universidad Laica Eloy Alfaro de Manabí

Extensión El Carmen

Carrera en Ingeniería en Software

TRIBUNAL DE SUSTENTACIÓN

Título del Trabajo de Titulación:

SISTEMA INFORMÁTICO CON ESQUEMAS POLIMÓRFICOS PARA EL MODELADO DE DATOS FITOTÉCNICOS EN LA GRANJA EXPERIMENTAL ULEAM EL CARMEN.

Modalidad:

Proyecto Integrador

Autor:

Zamora Loo Jose Manuel

Tutor:

Ing. Carlos Vinicio López Rodríguez

Tribunal de Sustentación:

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

Fecha de Sustentación:

26 de agosto del 2026

UNIVERSIDAD LAICA “ELOY ALFARO” DE MANABÍ

EXTENSIÓN EN EL CARMEN



DECLARACIÓN DE AUTORÍA

La responsabilidad del contenido de este Trabajo de Titulación, cuyo tema es: Sistema informático con esquemas polimórficos para el modelado de datos fitotécnicos en la granja experimental ULEAM El Carmen, corresponde exclusivamente a: Zamora Loor Jose Manuel con C.I.1316924834 y los derechos patrimoniales de la misma corresponden a la Universidad Laica “Eloy Alfaro” de Manabí.

— *Jose Manuel Zamora Loor.* —

Zamora Loor Jose Manuel

C.I.1316924834

DEDICATORIA

A mis abuelos Zambrano Rosa y Loo de la Cruz Manuel, mis padres Zambrano María y Zamora Alberto, por su apoyo incondicional y por estar presentes en mi vida.

A Yaneli Muñoz por acompañarme durante este proceso, formar parte de este camino y creer en mí.

A mis compañeros de carrera, Royers, Adrián, Diego, Angelo y Jean, por acompañar durante toda la carrera.

A todas las personas que estuvieron presentes durante todo el proceso.

Zamora Loo Jose

AGRADECIMIENTO

A mi tutor, Ing. Carlos Vinicio López Rodríguez.

A mis compañeros Royers, Adrián, Diego, Angelo y Jean, por el trabajo en equipo a lo largo del proyecto.

A todas las personas que contribuyeron a esta meta, mi agradecimiento infinito.

Zamora Loor Jose

ÍNDICE GENERAL

PORTADA	I
CERTIFICACIÓN DEL TUTOR	II
TRIBUNAL DE SUSTENTACIÓN	III
DECLARACIÓN DE AUTORÍA.....	IV
DEDICATORIA.....	V
AGRADECIMIENTO	VI
ÍNDICE GENERAL	VII
ÍNDICE DE TABLAS	XIII
ÍNDICE DE ILUSTRACIONES.....	XV
ÍNDICE DE ANEXOS	XVIII
RESUMEN	XIX
CAPÍTULO I	1
1 INTRODUCCIÓN	1
1.1 Introducción	1
1.2 Presentación del tema	2
1.3 Contextualización de la problemática	2
1.4 Planteamiento del problema.....	3
1.4.1 Problematización	3
1.4.2 Génesis del problema	4
1.4.3 Estado actual del problema	4
1.5 Diagrama causa-efecto	4

1.6	Objetivos	6
1.6.1	Objetivo general	6
1.6.2	Objetivos específicos	6
1.7	Justificación	7
1.8	Impactos esperados.....	7
1.8.1	Impacto tecnológico	7
1.8.2	Impacto social.....	8
1.8.3	Impacto ecológico	8
CAPÍTULO II.....		9
2	MARCO TEÓRICO	9
2.1	Antecedentes históricos.....	9
2.1.1	Evolución de los sistemas de bases de datos	9
2.1.2	Contexto agrícola y datos fitotécnicos	9
2.2	Investigaciones relacionadas.....	10
2.3	Definiciones conceptuales y categorías clave.....	13
2.3.1	Fundamentos de las bases de datos NoSQL	13
2.3.2	Modelos de datos orientados a documentos	14
2.3.3	Esquemas polimórficos.....	14
2.3.4	Teorema CAP y consistencia de datos	15
2.3.5	Operaciones transaccionales y patrón CRUD	16
2.3.6	Persistencia y replicación.....	16

2.3.7	Arquitectura hexagonal para el diseño del backend.....	17
2.3.8	Integración de servicios con Node.js y TypeScript.....	18
2.3.9	Generalidades del cultivo de Musa paradisiaca.....	19
2.3.10	Requerimientos edafoclimáticos del sector.....	19
2.3.11	Fenología y fases de desarrollo biológico	20
2.3.12	Parámetros biométricos y variables fitotécnicas.....	21
2.3.13	Control fitosanitario y principales enfermedades	21
2.3.14	Manejo de ensayos agrícolas y experimentación.....	22
2.3.15	Gestión de cosecha y rendimiento agrícola.....	23
2.3.16	Importancia socioeconómica y tecnificación del cultivo	23
2.3.17	Metodologías ágiles y el marco de trabajo Scrum en el desarrollo de software	24
2.4	Conclusiones del marco teórico.....	25
CAPÍTULO III		27
3	MARCO INVESTIGATIVO	27
3.1	Introducción	27
3.2	Tipo y métodos de investigación	27
3.2.1	Tipo de investigación	27
3.2.2	Investigación Aplicada.....	28
3.2.3	Investigación Explicativa.....	28
3.2.4	Métodos de investigación	29
3.2.5	Método Analítico-Sintético	29

3.2.6	Método Empírico.....	30
3.3	Fuentes de información	30
3.3.1	Fuentes primarias	30
3.3.2	Fuentes secundarias.....	31
3.4	Estrategia de recolección de datos.....	32
3.4.1	Población y muestra	32
3.4.2	Técnicas e instrumentos de recolección.....	33
3.5	Análisis e interpretación de resultados	34
3.5.1	Análisis cuantitativo: Encuesta a estudiantes	35
3.5.2	Análisis cualitativo: Entrevista a experto	45
CAPÍTULO IV		50
4	MARCO PROPOSITIVO	50
4.1	Introducción al capítulo.....	50
4.2	Descripción de la propuesta tecnológica	50
4.2.1	Arquitectura del software.....	50
4.2.2	Descripción de módulos principales	51
4.2.3	Tecnologías claves a utilizar	52
4.2.4	Modelo de desarrollo de software seleccionado y justificado	53
4.3	Recursos requeridos.....	53
4.3.1	Recursos humanos.....	53
4.3.2	Recursos tecnológicos.....	54

4.3.3	Recursos económicos	55
4.4	Desarrollo del software mediante metodología Scrum	56
4.4.1	Descripción del producto	57
4.4.2	Requerimientos del sistema	59
4.4.3	Diseño del sistema.....	63
4.4.4	Arquitectura del sistema	76
4.4.5	Roles Scrum.....	80
4.4.6	Definición de Hecho (DoD).....	82
4.4.7	Product Backlog	87
4.4.8	Sprint 1: Autenticación, Seguridad e integración de cámara.....	90
4.4.9	Resultado final del proyecto	104
CAPÍTULO V		115
5	EVALUACIÓN DE RESULTADOS.....	115
5.1	Introducción	115
5.2	Presentación y monitoreo de resultados	116
5.2.1	Planificación de la evaluación	117
5.2.2	Ejecución del monitoreo	120
5.3	Interpretación objetiva de resultados.....	129
CAPÍTULO VI		132
6	CONCLUSIONES Y RECOMENDACIONES	132
6.1	Conclusiones	132

6.2	Recomendaciones	133
	REFERENCIAS	134
	ANEXOS	139
	GLOSARIO	147

ÍNDICE DE TABLAS

Tabla 1: Plan de recolección de datos.	32
Tabla 2: Resultados e interpretación de la encuesta aplicada a estudiantes.	35
Tabla 3: Resultados e interpretación de la entrevista al experto.	45
Tabla 4: Recursos humanos requeridos para el proyecto tecnológico	54
Tabla 5: Recursos tecnológicos (Hardware y Software) requeridos.	54
Tabla 6: Presupuesto y recursos económicos.	55
Tabla 7 : Usuarios del sistema y sus funciones principales.	58
Tabla 8 : Especificación de Historia de usuario HU-01.	59
Tabla 9 : Especificación de historia de usuario HU-02.	60
Tabla 10 : Especificación de historia de usuario HU-03.	60
Tabla 11 : Especificación de historia de usuario HU-04.	61
Tabla 12 : Especificación de historia de usuario HU-05.	62
Tabla 13: Requerimientos funcionales del sistema AgroAR Plátano.	62
Tabla 14 : Especificación de caso de uso: Registro de observación fitotécnica.	64
Tabla 15 : Especificación de caso de uso: Sincronización asíncrona de datos.	65
Tabla 16 : Especificación de caso de uso: Generación de reporte de evolución.	67
Tabla 17: Estructura de la colección de análisis en MongoDB Atlas.	72
Tabla 18: Estructura de la tabla scan_reports en SQLite.	74
Tabla 19: Herramientas y tecnologías utilizadas en el desarrollo del proyecto.	79
Tabla 20: Descripción y asignación de roles Scrum en el proyecto.	81
Tabla 21: Criterios generales de aceptación y calidad del software.	84
Tabla 22: Product Backlog general: Funcionalidades, prioridades, Sprints y estado inicial.	88
Tabla 23: Planificación general del Sprint 1.	90

Tabla 24: Sprint Backlog del Sprint 1.....	92
Tabla 25: Prueba del formulario de registro de usuario.....	96
Tabla 26: Prueba del almacenamiento local en SQLite.	97
Tabla 27: Pruebas de sincronización con MongoDB Atlas.	98
Tabla 28: Pruebas de carga de imágenes en el sistema.....	99
Tabla 29: Incremento funcional obtenido al finalizar el Sprint 1.	100
Tabla 30: Estado actual del Product Backlog.....	102
Tabla 31: Funcionalidades implementadas en el prototipo AgroAR.	104
Tabla 32: Procesos principales evaluados en el sistema AgroAR.	116
Tabla 33: Planificación de la evaluación del sistema.	118
Tabla 34: Ejecución del monitoreo de procesos del sistema.	120
Tabla 35: Evaluación del proceso de autenticación y control de acceso.	122
Tabla 36: Evaluación del proceso de escaneo y análisis de la planta.	124
Tabla 37: Evaluación del proceso de sincronización con MongoDB Atlas.	126
Tabla 38: Precisión de los módulos de inteligencia artificial evaluados.	128
Tabla 39: Planificación de pruebas de ejecución de Inteligencia Artificial.	128

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Diagrama sobre la desestructuración y pérdida de trazabilidad en el registro de datos fitotécnicos de la Granja Experimental Uleam El Carmen.....	5
Ilustración 2: Diagrama de caso de uso: Registro de observación fitotécnica.	64
Ilustración 3: Diagrama de caso de uso: Sincronización asíncrona de datos.	65
Ilustración 4: Diagrama de caso de uso: Generación de reporte de evolución.....	66
Ilustración 5: Diagrama de secuencia: Registro de observación fitotécnica.	68
Ilustración 6: Diagrama de secuencia: Sincronización de datos asíncrona.	68
Ilustración 7: Diagrama de secuencia: Generación de reporte de evolución.....	69
Ilustración 8: Diagrama de estado: Entidad revisión.	70
Ilustración 9: Diagrama de estado: Entidad usuario.	70
Ilustración 10: Diagrama lógico de persistencia y sincronización de datos.	71
Ilustración 11: Visualización de registro de salud vegetal con el campo discriminador en MongoDB Atlas.	75
Ilustración 12: Visualización de registro de plaga con el campo discriminador en MongoDB Atlas.....	76
Ilustración 13: Diagrama arquitectónico del sistema: Cliente/Servidor, Modelo en Capas y MVC.....	78
Ilustración 14: Pantalla de inicio de sesión de AgroAR Plátano.....	94
Ilustración 15: Respuesta del endpoint de inicio de sesión en Postman.	94
Ilustración 16: Captura de la colección users en MongoDB Atlas.	96
Ilustración 17: Pantalla de inicio de sesión de la aplicación móvil.....	107
Ilustración 18: Pantalla de registro de usuario.	108
Ilustración 19: Dashboard o pantalla principal del estudiante.	108
Ilustración 20: Pantalla de cámara o selección de análisis.	109

Ilustración 21: Pantalla de resultado del análisis.	109
Ilustración 22: Pantalla del historial de revisiones.	110
Ilustración 23: Pantalla de detalle de una revisión.	110
Ilustración 24: Vista del PDF generado o archivo PDF abierto.	111
Ilustración 25: Pantalla del chatbot agrícola.	111
Ilustración 26: Pantalla del panel administrativo.	112
Ilustración 27: MongoDB Atlas mostrando las colecciones y documentos registrados.	112
Ilustración 28: Imagen subida en Cloudinary o campo imageUrl dentro de MongoDB Atlas.....	113
Ilustración 29: Panel de Render mostrando el backend activo o respuesta del endpoint /api/health.	113
Ilustración 30: Validación del inicio de sesión en Postman, con respuesta exitosa del backend y generación del token JWT.....	141
Ilustración 31: Pantalla de inicio de sesión del prototipo AgroAR Plátano ejecutado en entorno de pruebas.	141
Ilustración 32: Panel principal del usuario estudiante con resumen de revisiones y acceso al análisis de plantas.	142
Ilustración 33: Historial de revisiones sincronizadas filtrada por usuario entre la aplicación móvil y MongoDB Atlas.	142
Ilustración 34: Panel administrativo para la gestión de usuarios, análisis recibidos y resumen general del sistema.	143
Ilustración 35: Respuesta del chatbot agrícola sobre plagas comunes en el cultivo de plátano.	143

Ilustración 36: Evidencia de la entrevista semiestructurada aplicada al docente especialista de la Granja Experimental ULEAM El Carmen.....	144
Ilustración 37: Resultados de la encuesta sobre la aceptación de tecnologías de agricultura de precisión en la Granja Experimental ULEAM El Carmen.	144
Ilustración 38: Resultados de la encuesta sobre los principales obstáculos tecnológicos para el uso de una aplicación agrícola en El Carmen.	145
Ilustración 39: Resultados de la encuesta sobre la aceptación de recomendaciones generadas por un sistema tecnológico a partir del análisis de imágenes.....	145
Ilustración 40: Resultados de la encuesta sobre la importancia del uso de datos digitales en el manejo de cultivos dentro de la granja.	146

ÍNDICE DE ANEXOS

Anexo A: Asignación de tutor	139
Anexo B: Reporte del sistema antiplagio	140
Anexo C: Fotografías	141
Anexo D: Evidencia de entrevista y encuestas	144

RESUMEN

El presente trabajo de integración curricular consistió en crear un sistema informático con esquemas polimórficos para modelar datos fitotécnicos en la Granja Experimental Uleam El Carmen. La dificultad identificada se vincula con la forma en que se registra la información agrícola de manera manual y dispersa, lo cual complica el mantenimiento de un archivo que permita consultar o analizar documentos anteriores y contar con evidencias técnicas en el terreno, por lo tanto, se desarrolla el prototipo AgroAR para asistir en el cultivo de plátano.

Para el levantamiento de información se aplicó una investigación de tipo aplicada y explicativa, con un enfoque mixto, dirigida a una muestra de 40 estudiantes vinculados con las prácticas agrícolas de la Granja Experimental y a un docente especialista como informante clave, mediante encuesta estructurada y entrevista semiestructurada.

La aplicación permite a los usuarios registrarse e iniciar sesión, seleccionar o tomar imágenes, realizar análisis sobre plagas, hongos y salud de la planta, además, almacena las revisiones hechas, posibilita revisar el historial y crea reportes en PDF. Asimismo, incluye un enfoque offline-first utilizando SQLite, lo que permite guardar datos sin conexión y sincronizarlos más tarde con MongoDB Atlas a través de un backend creado con Node.js y TypeScript.

El desarrollo se estructuró a través de la metodología Scrum, empleando historias de usuario, sprints y pruebas de funcionalidad. Los resultados mostraron que el prototipo satisface el flujo principal establecido, pues facilita el registro, almacenamiento, sincronización y consulta de revisiones agrícolas.

Palabras clave: esquemas polimórficos, datos fitotécnicos, SQLite, MongoDB Atlas, AgroAR Plátano, offline-first.

Abstract

This curricular integration project consisted of creating a computer system with polymorphic schemas to model phytotechnical data at the Uleam El Carmen Experimental Farm. The identified difficulty is related to the manual and scattered way in which agricultural information is recorded, which complicates the maintenance of an archive that allows for consulting or analyzing previous documents and having technical evidence in the field. Therefore, the AgroAR prototype was developed to assist in banana cultivation.

For data collection, an applied and explanatory research approach was used, with a mixed-methods focus, targeting a sample of 40 students involved in the agricultural practices of the Experimental Farm and a specialist professor as a key informant, using a structured survey and semi-structured interview.

The application allows users to register and log in, select or take images, perform analyses on pests, fungi, and plant health, and also stores completed reviews, allows users to review the history, and creates PDF reports. It also includes an offline-first approach using SQLite, allowing data to be saved offline and later synchronized with MongoDB Atlas via a backend built with Node.js and TypeScript.

Development was structured using the Scrum methodology, employing user stories, sprints, and functional testing. The results showed that the prototype satisfies the established main workflow, facilitating the recording, storage, synchronization, and querying of agricultural reviews.

Keywords: polymorphic schemas, phytotechnical data, SQLite, MongoDB Atlas, AgroAR Plátano, offline-first.

CAPÍTULO I

1 INTRODUCCIÓN

1.1 Introducción

Manejar bien la información agrícola resulta clave para dar seguimiento a los cultivos, tomar decisiones técnicas acertadas y sostener el trabajo de investigación en los espacios de práctica universitaria. En la Granja Experimental Uleam El Carmen, por ejemplo, las actividades agrícolas generan de manera continua información sobre la condición de las plantas como observaciones visuales, imágenes y evaluaciones técnicas que deben archivarse de manera organizada, pero el inconveniente surge cuando esta información se escribe a mano o se encuentra dispersa en documentos sueltos, ya que se pierde la trazabilidad, se complica la consulta de antecedentes y se desaprovecha el potencial de esos datos para futuros análisis.

De esta necesidad nace la idea de construir una solución informática capaz de registrar, organizar y consultar datos fitotécnicos de manera más eficiente. Un reto particular de este tipo de información es que no siempre sigue la misma estructura: varía según el cultivo, el tipo de análisis o lo que se encuentre durante cada revisión. Por eso el proyecto apuesta por esquemas polimórficos, que permiten guardar información con formatos flexibles dentro de una base de datos NoSQL, sin exigir que todos los registros tengan exactamente los mismos campos.

Esta iniciativa se materializa en AgroAR, un modelo de aplicación móvil centrado en el cultivo de plátano, por eso con esta herramienta el usuario puede examinar imágenes capturadas con la cámara o seleccionadas de la galería, registrar observaciones, consultar su historial, crear informes en PDF y continuar trabajando, incluso sin conexión a internet, esto se consigue al integrar almacenamiento local con SQLite, sincronización con MongoDB Atlas, manejo de imágenes en Cloudinary y un backend en la nube.

La aplicación incluye además módulos de inteligencia artificial que ayudan en el análisis agrícola, especialmente en la identificación de plagas y la valoración de la salud de plantas, es importante señalar que estos resultados no reemplazan el juicio de un técnico especializado, sino que sirven como apoyo para guiar la revisión y documentar el proceso, para que el chatbot agrícola diseñado pueda aclarar inquietudes y brindar sugerencias sobre el cultivo de plátano.

El proyecto se llevó a cabo utilizando la metodología Scrum, desglosando el trabajo en historias de usuario, tareas, sprints y aumentos funcionales, esta metodología facilitó la organización del proceso de creación del prototipo en fases, comenzando con el análisis del problema, seguido del diseño de la arquitectura, la implementación de los módulos clave y, por último, la validación del sistema.

1.2 Presentación del tema

El presente proyecto de integración curricular tiene como propósito el diseño y desarrollo de un sistema informático basado en esquemas polimórficos, orientado al modelado de datos fitotécnicos en la Granja Experimental Uleam El Carmen. Con ello, se pretende dotar a la institución de una infraestructura digital robusta que posibilite la estructuración, el almacenamiento y la gestión eficiente de la información biológica y técnica generada a partir de las actividades agrícolas desarrolladas en dicho espacio.

1.3 Contextualización de la problemática

A escala mundial, la digitalización agrícola ha generado un aumento exponencial en la cantidad de datos biológicos y técnicos producidos por la industria. El mayor reto informático en este ámbito se encuentra en que los datos fitotécnicos son naturalmente diversos y cambiantes, históricamente, el esfuerzo por registrar esta variedad morfológica en base de datos relacionales inflexibles ha conducido a tablas saturadas de valores nulos, esquemas dispersos y un elevado costo de mantenimiento. Como respuesta, la ingeniería de software actual tiende

hacia la implementación de bases de datos NoSQL, donde el uso de esquemas polimórficos permite que documentos con diferentes estructuras coexistan eficientemente en una misma colección, brindar la flexibilidad necesaria para modelar la naturaleza cambiante de los cultivos.

De manera específica, en la Granja Experimental de la Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen, esta problemática se manifiesta en la ausencia de un sistema de persistencia de datos centralizado. Si bien este espacio constituye un laboratorio vivencial fundamental para las prácticas agrícolas y la investigación académica, la recolección de los datos fitotécnicos correspondientes a las distintas parcelas se lleva a cabo de forma manual y desestructurada.

Esta deficiencia organizativa y tecnológica trae consecuencias directas para la institución, dificulta el seguimiento cronológico preciso del desarrollo fenológico de los cultivos, incrementa el riesgo de pérdida de datos de investigaciones valiosos con el paso de los semestres, y limita severamente la capacidad de la granja para aprovechar su propia información en futuros análisis o en el desarrollo de herramientas de capacitación técnica interactivas basadas en datos reales.

1.4 Planteamiento del problema

1.4.1 Problematización

El centro del problema se encuentra en la gran variabilidad y heterogeneidad que define la información agronómica, las métricas fitotécnicas muestran variaciones significativas dependiendo de la especie cultivada, por lo tanto, las variables fenológicas obtenidas en una planta de plátano no tienen la misma estructura que las de un cultivo de ciclo corto, intentando plasmar esta variedad morfológica con estructuras rígidas y estandarizadas, o usando herramientas ofimáticas comunes, se produce una incompatibilidad de formatos, por ende esta restricción estructural fuerza a los investigadores a ajustar datos diversos a formatos rígidos,

resultando en registros inconsistentes, casillas vacías y una base de información fragmentada que impide el análisis relacional entre diferentes cultivos.

1.4.2 Génesis del problema

La causa de esta deficiencia se remonta a la fase inicial de formación y desarrollo de la Granja Experimental, en las etapas iniciales la recopilación de datos en el terreno se basaba en metodologías analógicas, sobre todo cuadernos de campo y formularios impresos, como método de registro, no obstante, a medida que la granja expandió su ámbito de acción, diversificó sus cosechas y aumentó la cantidad de prácticas preprofesionales e investigaciones realizadas, el volumen de información se multiplicó sin que los métodos de almacenamiento de datos progresaran al mismo ritmo, la falta de una planificación arquitectónica enfocada desde el principio en la gestión de la información llevó a que la granja acumulara datos empíricos sin poseer una estrategia de normalización técnica.

1.4.3 Estado actual del problema

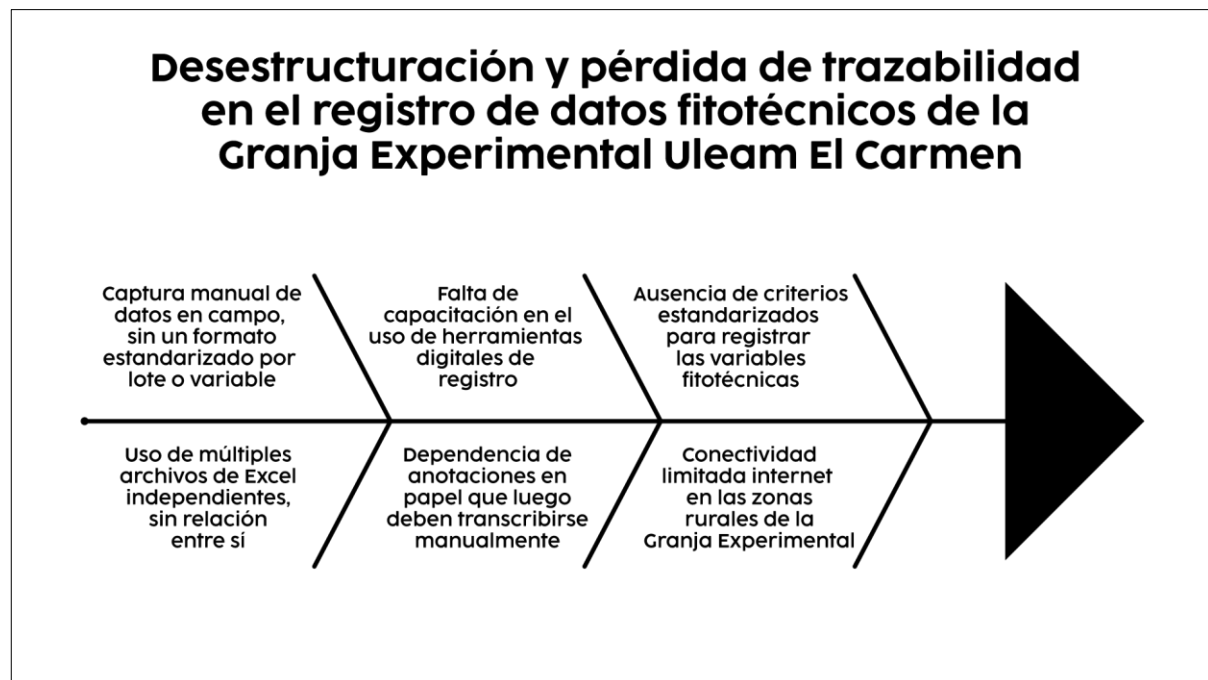
En la actualidad, la administración de la información en la Granja Experimental se encuentra en un estado crítico de desorganización, los datos fitotécnicos recopilados cada semestre actúan como silos separados, dispersos en varios documentos de Excel sin conexión entre ellos o registrados de forma manual, esa fragmentación implica una disminución gradual de la trazabilidad y del registro investigativo, por lo tanto, se vuelve operativamente impracticable realizar consultas complejas, elaborar reportes consolidados o comparar el desempeño de una parcela a través del tiempo sin depender de procesos manuales detallados, susceptibles a errores humanos.

1.5 Diagrama causa-efecto

Para comprender a profundidad la desestructuración de los datos fitotécnicos mencionada anteriormente, es necesario identificar las raíces del problema. En la **Figura 1** se

presenta el diagrama ishikawa, el cual establece las seis causas principales y sus respectivas consecuencias directas en la gestión de la información agrícola dentro la Granja Experimental.

Ilustración 1: Diagrama sobre la desestructuración y pérdida de trazabilidad en el registro de datos fitotécnicos de la Granja Experimental Uleam El Carmen.



1.6 Objetivos

1.6.1 Objetivo general

Desarrollar un sistema informático con esquemas polimórficos para el modelado de datos fitotécnicos en la granja experimental Uleam El Carmen.

1.6.2 Objetivos específicos

Definir la problemática actual en la gestión de datos fitotécnicos mediante el análisis del contexto agrícola y tecnológico a nivel macro, meso y micro, para establecer los fundamentos y la dirección de la investigación.

Fundamentar teóricamente las arquitecturas de bases de datos NoSQL y la aplicación de esquemas polimórficos, a partir de la revisión de literatura especializada y normativa técnica, con el fin de sustentar el desarrollo del modelado de datos fitotécnicos.

Analizar la situación actual de la administración de información fitotécnica en la Granja Experimental Uleam El Carmen, mediante técnicas de recolección de datos necesarias para el modelado flexible de las variables biológicas.

Crear un sistema informático fundamentado en esquemas polimórficos para el modelado de datos fitotécnicos, utilizando metodologías de desarrollo ágil, que permita una gestión efectiva y escalable de la información biológica de los cultivos.

Validar la funcionalidad y el rendimiento del sistema informático implementado, a través de pruebas de aceptación de usuario y validación técnica de los esquemas polimórficos, con el fin de garantizar el cumplimiento de los requerimientos identificados en la Granja Experimental.

1.7 Justificación

Académicamente, solucionar este desafío de persistencia y ordenamiento de datos fortalece a la comunidad universitaria. Los estudiantes, docentes e investigadores de la extensión contarán con la certeza de que la información recolectada en campo posee la cohesión estructural necesaria para someterse a consultas complejas y análisis estadísticos rigurosos, esto resulta en una mejora directa de la calidad de las prácticas preprofesionales, además de una mayor habilidad para producir estudios agronómicos basados en históricos de datos precisos, lo que facilita correlacionar variables de rendimiento y salud vegetal con un alto grado de confiabilidad.

Por último, en el contexto institucional, vencer esta barrera organizativa implica un avance significativo hacia la modernización agrícola, al establecer fundamentos sólidos en la gestión de su propia información, la Granja Experimental supera el atraso administrativo y se coloca estratégicamente para incorporar, en el futuro, nuevas tecnologías de precisión o herramientas interactivas de formación, de este modo, se asegura que la institución no solo produzca conocimiento agronómico en el campo, sino que también lo conserve y lo utilice de manera efectiva conforme a estándares técnicos actuales.

1.8 Impactos esperados

1.8.1 Impacto tecnológico

La ejecución de este proyecto implicará una actualización significativa en la estructura de la información de la Granja Experimental, al trasladar los registros manuales y las hojas de cálculo dispersas a un sistema informático centralizado, basado en base de datos NoSQL, se eliminarán los silos de información actuales, con la implementación de esquemas polimórficos proporcionará un grado de escalabilidad y adaptabilidad sin igual para la representación de datos biológicos diversos, permitiendo así la creación de una infraestructura digital sólida que

facilite, en el futuro, la incorporación de microservicios y la creación de aplicaciones interactivas avanzadas.

1.8.2 Impacto social

La iniciativa impactará directamente a la comunidad educativa de la extensión universitaria, especialmente a estudiantes, profesores e investigadores, al contar con una herramienta que asegure la trazabilidad y la integridad de la información fitotécnica, se mejorará la calidad de las prácticas preprofesionales y de los proyectos de graduación de carreras del campo agronómico, indirectamente este impulso de investigación académica también impactará en el sector agrícola local de El Carmen, al ayudar en la capacitación de profesionales más capacitados para tomar decisiones fundamentales en datos concretos.

1.8.3 Impacto ecológico

La correcta organización y el examen histórico de los datos fitotécnicos facilitarán un monitoreo más exacto del crecimiento fenológico y de la presencia de plagas en los cultivos, al tener acceso a información confiable y centralizada, los investigadores podrán mejorar la utilización de insumos agrícolas, promoviendo la adopción de prácticas de agricultura de precisión, esto ayudará a disminuir la huella química de la granja y a fomentar un manejo ecológicamente responsable de las áreas experimentales.

CAPÍTULO II

2 MARCO TEÓRICO

2.1 Antecedentes históricos

2.1.1 Evolución de los sistemas de bases de datos

Durante décadas, el almacenamiento de datos estuvo dominado por el modelo relacional, el cual organiza la información en tablas con filas y columnas predefinidas. Si bien este esquema garantizó la integridad de los datos en entornos transaccionales clásicos, con el tiempo presentó serias limitaciones al intentar manejar grandes volúmenes de información o datos con estructuras cambiantes, ya que obliga a modificar todo el esquema de la base de datos cada vez que se requería agregar un nuevo campo (Sarasa & Cabezuelo, 2021).

Como respuesta a la rigidez de las bases de datos tradicionales y a la necesidad de escalar aplicaciones web de forma ágil, surgió el movimiento NoSQL. Este nuevo paradigma introdujo modelos de datos alternativos, diseñados específicamente para gestionar datos semiestructurados o no estructurados sin la necesidad de esquemas fijos, permitiendo una adaptación más rápida a los cambios en los requerimientos del software (Wu & Mueller, 2024).

En la actualidad, la tecnología ha madurado hacia el uso de base de datos documentales que soportan esquemas polimórficos. Esta característica permite almacenar en una misma colección documentos con diferentes estructuras y atributos, facilitando el modelado de objetos complejos y heterogéneos del mundo real sin sacrificar el rendimiento ni la escalabilidad del sistema informático (Sarasa, 2021).

2.1.2 Contexto agrícola y datos fitotécnicos

La historia de la producción agrícola en la región está intrínsecamente ligada a la expansión de las grandes compañías fruteras durante el siglo XX. Este período marcó el inicio de la explotación sistemática del cultivo de musáceas, donde se establecieron las primeras dinámicas de exportación y control de tierras que definieron la estructura agraria de países

como Ecuador, transformando la economía local y las prácticas de cultivo tradicionales (Kepner & Soothill, 2022).

Con el crecimiento de la demanda global, el sector bananero se vio obligado a evolucionar desde prácticas empíricas hacia métodos más técnicos para mantener la competitividad. Esta transición supuso una mayor atención a los volúmenes de producción y al control de calidad, lo que a su vez generó la necesidad de gestionar una cantidad creciente de información sobre los procesos de cultivo, si bien los métodos de registro no evolucionaron al mismo ritmo que las técnicas agrícolas (Charvet & Espinoza, 2021).

Pese a la relevancia agronómica del plátano, los productores y centros experimentales continúan enfrentando brechas significativas en la gestión de sus datos fitotécnicos. La falta de adopción de tecnologías digitales modernas en las zonas rurales ha perpetuado el uso de registros manuales o dispersos, lo que limita la capacidad de análisis histórico y la eficiencia en la toma de decisiones críticas para el manejo de plagas y enfermedades en los cultivos (Bhuiyan et al., 2023).

2.2 Investigaciones relacionadas

En el contexto de la adopción de tecnologías digitales y la gestión de grandes volúmenes de información agraria, Ghiwari et al. (2017), diseñaron e implementaron un sistema de almacenamiento para datos agro informáticos en la India. En su investigación, los autores utilizaron Couchbase, una base de datos documental de la familia NoSQL, para gestionar métricas ambientales (temperatura, presión y precipitación) recopiladas mediante dispositivos de Internet de las Cosas (IoT). El objetivo principal fue superar la alta complejidad estructural y de almacenamiento que exige la jerarquía de los departamentos de agricultura de dicho país. Los resultados de sus pruebas de rendimiento demostraron una eficiencia computacional notable: mediante un clúster de tres servidores agrícolas, el sistema logró procesar y actualizar la información de 100 explotaciones agrícolas en tan solo 63 segundos,

aplicando además funciones de mapeo de datos específicas para modelar las necesidades del cultivo.

El análisis crítico de este antecedente resulta fundamental para el desarrollo del presente proyecto. La necesidad de abandonar el modelo relacional convencional para el manejo ágil y escalable de datos agrícolas concuerda con lo expuesto por Sarasa y Cabezuelo (2021), quienes sostienen que la flexibilidad estructural de las bases de datos orientadas a documentos es indispensable para evitar cuellos de botella. Mientras que la investigación citada resuelve el problema apoyándose en clústeres y mapeos fijos, el sistema propuesto en la ULEAM aplicará esquemas polimórficos directamente sobre el motor NoSQL. Esta evolución metodológica permitirá que el sistema no solo alcance un alto rendimiento transaccional, sino que además cuente con la flexibilidad estructural necesaria para modelar dinámicamente variables fitotécnicas heterogéneas, soportando múltiples especies biológicas dentro de una sola arquitectura, sin necesidad de rediseños ante cada nuevo tipo de cultivo.

Para enriquecer el análisis del estado del arte desde una óptica arquitectónica más amplia, Carvalho et al. (2025) llevaron a cabo una revisión sistemática completa de los modelos de bases de datos utilizados en la agricultura contemporánea, en este estudio, los autores analizaron las metodologías de modelado, integración y gestión de datos agrícolas elaboradas entre 2010 y 2023, centrándose particularmente en el ámbito del mejoramiento vegetal, los resultados de este estudio demuestran que el acceso a información agrícola organizada adecuadamente es fundamental para la toma de decisiones en la actualidad, subrayando la urgente necesidad de implementar técnicas y modelos de datos especializados que se ajusten a las particularidades y a la alta variabilidad del entorno agronómico.

Al vincular las conclusiones de esta revisión sistemática con el presente proyecto de titulación, se ratifica la pertinencia tecnológica de la investigación. El desafío de estructurar la digitalización agrícola también es respaldado por Saiz-Rubio y Rovira-Más (2020), quienes

enfatan que la gestión avanzada de datos de cultivos exige modelos especializados frente a la alta variabilidad del entorno agronómico. En este punto radica precisamente el valor tangible que aporta la arquitectura propuesta para la Granja Experimental Uleam, extensión El Carmen: al implementar esquemas polimórficos sobre una base de datos NoSQL, el proyecto materializa de manera concreta las recomendaciones planteadas por la literatura científica. Esta solución técnica permite gestionar la complejidad inherente a los datos fitotécnicos mediante un modelo altamente flexible, capaz de integrar distintas estructuras de información sin las restricciones propias de las bases de datos rígidas convencionales evaluadas en el desarrollo de la agroinformática.

En la misma línea de investigación enfocada en la flexibilidad arquitectónica, Wu y Mueller (2024) desarrollaron DynamoField, un marco de trabajo basado en base de datos NoSQL diseñado específicamente para gestionar la alta variabilidad de la información en ensayos de campo agrícolas. Los autores llevaron a cabo esta solución mediante Amazon Web Services DynamoDB, con el propósito de habilitar que pequeños grupos de investigación ajusten dinámicamente los esquemas de la base de datos sin comprometer la integridad del sistema, los resultados de su ejecución mostraron que la adopción de un enfoque NoSQL, simplifica significativamente la incorporación de nuevos tipos de datos biológicos provenientes de diversos colaboradores, superando las restricciones que imponen las base de datos relacionales en contextos de investigación agrícola.

El análisis de este artículo reciente constituye una validación técnica directa para la hipótesis del presente proyecto de titulación. La necesidad de modelar información agronómica variante a través de esquemas dinámicos se alinea con las directrices arquitectónicas documentadas por Giamas (2022) para bases de datos polimórficas. Sin embargo, a diferencia de propuestas que generan dependencia con servicios comerciales en la nube (AWS), el sistema creado para la Granja Experimental aplicará esquemas polimórficos de forma independiente,

esto proporcionará a la universidad la misma escalabilidad y flexibilidad evidenciada por Wu y Mueller en sus datos fitotécnicos, al tiempo que le brinda un control absoluto y soberano sobre la implementación y duración de su información, eliminando los gastos continuos de infraestructura en la nube.

2.3 Definiciones conceptuales y categorías clave

2.3.1 Fundamentos de las bases de datos NoSQL

El paradigma NoSQL (Not Only SQL) surge como una respuesta arquitectónica a las limitaciones de escalabilidad y flexibilidad impuestas por los sistemas de gestión de bases de datos relacionales (SGBDR). Como señala Phaltankar et al. (2020), a diferencia del modelo tabular tradicional, las bases de datos NoSQL, no requieren la declaración de un esquema fijo predefinido previo a la inserción de la información, lo que permite la persistencia ágil de datos no estructurados y semiestructurados.

A nivel arquitectónico, estos motores abandonan el modelo rígido de cumplimiento estricto ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) en favor de la semántica BASE (Básicamente Disponible, Estado Suave, Consistencia Eventual). Esto permite aplicar técnicas como el particionamiento de datos (sharding) a través de múltiples servidores, distribuyendo la carga transaccional sin generar los cuellos de botella que típicamente presentan los esquemas relacionales (Sarasa & Cabezuelo, 2021).

Bajo esta premisa de flexibilidad expuesta por Sarasa (2021), la arquitectura del sistema informático en la Granja Experimental Uleam extensión El Carmen resulta viable. Las mediciones biológicas en campo son intrínsecamente variables; limitar la recolección de datos a una estructura relacional obligaría a rediseñar la base de datos por cada nuevo experimento, por lo que el paradigma NoSQL actuará como un repositorio dinámico capaz de absorber todo tipo de mediciones.

2.3.2 Modelos de datos orientados a documentos

Dentro de la taxonomía de los sistemas NoSQL, las bases de datos orientadas a documentos representan uno de los enfoques más versátiles para el modelado de entidades complejas, en este modelo, la unidad fundamental de almacenamiento es un documento, generalmente organizado y codificado en formatos estándar como JSON o su versión binaria optimizada, BSON (Giamas, 2022).

Desde la perspectiva del rendimiento computacional, el modelo documental minimiza drásticamente las operaciones de cruce de estructuras (Joins). Al aplicar el principio de diseño centrado en la incrustación embedding, un solo acceso al motor de base de datos recupera el estado completo de la entidad, mejorando la latencia en la capa de red y reduciendo la carga computacional (Phaltankar et al., 2020).

Aplicando este principio de almacenamiento unificado descrito por Wu y Mueller (2024) al modelado de cultivo de *Musa paradisiaca*, un “Documento de planta” contendrá anidados todos sus historiales biológicos: fechas de siembra, controles y mediciones en un solo objeto jerárquico. Así, el backend entregará el perfil biológico completo sin ensamblar múltiples tablas.

2.3.3 Esquemas polimórficos

Aplicando el esquema polimórfico al diseño y modelado de bases de datos documentales, se define como la capacidad del motor de base de datos para almacenar documentos de distintas estructuras dentro de una misma colección unificada. Un esquema polimórfico permite que el sistema modele entidades del mundo real que comparten una clasificación común, pero que difieren significativamente en sus métricas específicas (Giamas, 2022).

En entornos backend actuales, el polimorfismo se gestiona de forma dinámica mediante un campo discriminador que se encuentra en la raíz del documento JSON. Este discriminador

actúa como una bandera lógica que instruye a la capa de acceso a datos sobre qué validaciones estructurales debe aplicar al momento de deserializar la información en la memoria del servidor (Phaltankar et al., 2020).

Para evitar el rediseño constante de tablas advertido por Wu y Mueller (2024), la adopción de esquemas polimórficos en los datos fitotécnicos resulta indispensable. Las variables biométricas de la fase vegetativa del plátano (como el diámetro de pseudotallo) poseen un esquema distinto a las métricas reproductivas; el polimorfismo garantiza que ambos registros vivan en la misma colección “Fenología” sin corromper la base de datos.

2.3.4 Teorema CAP y consistencia de datos

El Teorema CAP establece un principio fundamental inmutable en el diseño de bases de datos distribuidas: un sistema solo puede garantizar simultáneamente dos de las tres siguientes propiedades: Consistencia, Disponibilidad y Tolerancia a particiones (Phaltankar et al., 2020).

En la práctica operativa de los motores NoSQL, la selección arquitectónica entre consistencia estricta o alta disponibilidad se gestiona mediante la configuración de los conjuntos de réplicas y a las políticas de confirmación de escritura. Un enfoque de alta disponibilidad permitirá registrar el dato inmediatamente, asumiendo una sincronización asíncrona en segundo plano conocida como consistencia eventual (Giamas, 2022).

Teniendo en cuenta la tolerancia a particiones mencionada por Ghiwari et al. (2017) y la realidad operativa de la Granja Experimental, el backend debe enfocarse en una alta disponibilidad, al capturar información en áreas con conectividad móvil inestable, el modelo de documentación facilitará el almacenamiento de las observaciones fenológicas en cachés o nodos secundarios, asegurando la continuidad de los datos sin detener el trabajo de campo.

2.3.5 Operaciones transaccionales y patrón CRUD

El acrónimo CRUD (Crear, Leer, Actualizar y Eliminar) representa las cuatro operaciones fundamentales de persistencia que cualquier sistema de gestión de bases de datos debe ofrecer para interactuar con la capa de almacenamiento, como indica Giamas (2022), en el entorno de las bases de datos NoSQL orientadas a documentos, estas operaciones se ejecutan directamente sobre estructura JSON o BSON, permitiendo la manipulación íntegra de un registro sin necesidad de afectar múltiples tablas simultáneamente.

A nivel de ejecución arquitectónica, las bases de datos documentales modernas garantizan que las operaciones CRUD a nivel de un único documento sean estrictamente atómicas. Esto significa que si un documento contiene arreglos anidados o subdocumentos complejos, cualquier operación de actualización o escritura se completará en su totalidad o fallará por completo, evitando estados de datos corruptos o parciales en el disco (Sarasa, 2021).

Siguiendo los lineamientos de persistencia atómica descritos por Wu y Mueller (2024), las operaciones CRUD en el sistema de Granja Experimental Uleam extensión El Carmen permitirán gestionar el ciclo de vida de la información agrícola de manera segura. Al utilizar la aplicación móvil para actualizar la fase fenológica de una planta de musa paradisíaca, un investigador realizará una operación transaccional que alterará completamente el documento polimórfico en el servidor, asegurando que el historial de campo permanezca intacto ante posibles errores de red durante la actualización.

2.3.6 Persistencia y replicación

La persistencia de datos implica la habilidad de un sistema informático para almacenar información de forma duradera, resistiendo fallos de hardware o reinicios del servidor, En el ecosistema NoSQL, esta persistencia se potencia mediante la replicación, definida como el proceso de sincronizar y mantener copias idénticas de los datos a través de múltiples servidores

interconectados (Sarasa, 2021). Este mecanismo garantiza que la información no residiera en un único punto de fallo (SPOF).

A nivel arquitectónico, la replicación se implementa comúnmente a través de un patrón de Conjunto de Réplicas, el cual consta de un nodo primario que recibe todas las operaciones de escritura, y múltiples nodos secundarios que aplican estas operaciones de forma asíncrona en sus propios conjuntos de datos (Phaltankar et al., 2020). Si el nodo primario experimenta una caída, el sistema ejecuta un proceso de elección automatizado para promover a un nodo secundario como el nuevo primario, asegurando la alta disponibilidad sin intervención manual.

Aplicando este principio de redundancia descrito por Ghiwari et al. (2017) a la Granja Experimental Uleam extensión El Carmen, la replicación resulta vital para proteger el trabajo de campo. Al documentar el historial del cultivo de *musa paradisiaca*, la base de datos repartirá copias de los datos fitotécnicos en nodos secundarios, si el servidor principal de la universidad sufre un corte eléctrico, el sistema de recopilación garantizará que ninguna medición biométrica recopilada por los estudiantes se pierda, manteniendo la integridad de la investigación agrícola.

2.3.7 Arquitectura hexagonal para el diseño del backend

La arquitectura hexagonal, denominada también patrón de Puertos y Adaptadores, es un modelo de diseño de software que intenta separar de manera rigurosa el núcleo de la lógica empresarial de las dependencias tecnológicas externas, como bases de datos, interfaces de usuario o servicios ajenos (Vieira, 2022). Este método asegura que la aplicación se base en sus casos de uso internos y no en las restricciones de la infraestructura subyacente.

En su implementación técnica, esta arquitectura segmenta el backend en capas concéntricas. El “Dominio” central agrupa las entidades esenciales y las reglas de negocio, los “Puertos” son interfaces abstractas que establecen la forma en que el dominio se interactúa con el exterior, y los “Adaptadores” son las implementaciones específicas que vinculan esos

puertos con la realidad, como un adaptador para las bases de datos NoSQL o un adaptador para controladores HTTP (Sarasa & Cabezuelo, 2021). Esta inversión de control permite probar la lógica central de forma aislada mediante pruebas unitarias.

Siguiendo este patrón de aislamiento validado por Carvalho et al. (2025), el backend del proyecto fitotécnico no acoplará sus reglas biológicas directamente al motor de base de datos. El modelo de crecimiento de la Musa paradisiaca residirá en el dominio central, mientras que los esquemas polimórficos funcionarán como un adaptador de infraestructura. Si en el futuro la universidad decide cambiar de motor de base de datos, el código central que agrupa las métricas agrícolas se conservará sin alteraciones, demostrando un alto nivel de madurez en la ingeniería del software.

2.3.8 Integración de servicios con Node.js y TypeScript

La creación de servicios backend actuales necesita plataformas efectivas y lenguajes confiables. Node.js proporciona un entorno de ejecución asíncrono y no bloqueante impulsado por eventos, ideal para gestionar múltiples solicitudes de red simultáneas; mientras que TypeScript actúa como un superconjunto tipado estáticamente que compila a JavaScript, detectando errores semánticos y de estructura durante la fase del desarrollo (Rozentals, 2021).

La integración de ambas tecnologías permite construir microservicios o arquitecturas monolíticas modulares altamente predecibles. TypeScript facilita la implementación de interfaces y tipo genéricos que definen contratos de datos estrictos entre las distintas capas de la aplicación, como los puertos en la arquitectura hexagonal, mientras que Node.js asegura que la latencia de las operaciones de lectura y escritura hacia la base de datos se maneje mediante promesas, optimizando el consumo de CPU del servidor (Casciaro & Mammino, 2020).

Bajo este esquema de integración expuesto por Wu y Mueller (2024), el servicio backend orquestrará eficientemente las peticiones de la aplicación móvil agrícola. TypeScript garantizará mediante interfaces polimórficas que los datos biométricos de la Musa paradisiaca

enviados desde el campo posean el formato numérico correcto antes de ser procesados, y Node.js procesará estas inserciones concurrentes de múltiples usuarios en la Granja Experimental Uleam extensión El Carmen sin bloquear el hilo principal, asegurando un rendimiento fluido del sistema.

2.3.9 Generalidades del cultivo de *Musa paradisiaca*

El cultivo de *Musa paradisiaca*, conocido comúnmente como plátano, representa una de las especies herbáceas de mayor envergadura a nivel mundial. Desde el enfoque botánico y de seguridad alimentaria, esta musácea es considerada un cultivo esencial para la subsistencia de comunidades en áreas tropicales y subtropicales, destacando por su elevada necesidad de nutrientes y su sistema de raíces poco profundas (Organización de las Naciones Unidas para la Alimentación y la Agricultura FAO, 2022).

Históricamente, la consolidación de este cultivo ha sido influenciada por factores biológicos, sino también por un profundo proceso de expansión agroindustrial, según Kepner y Soothill (2022), el crecimiento de las plantaciones de musáceas en América Latina alteró la dinámica de la zona, transformando áreas completas en núcleos de exportación y requiriendo un control cada vez más estricto sobre la gestión de las plantaciones.

Esta categorización general es el punto de partida del modelo de datos. Tal como establecen Wu y Mueller (2024), estructurar taxonómicamente la especie en la base de datos es el primer paso para diseñar ensayos de campo funcionales, permitiendo a los investigadores segmentar la información recolectada y evitar la mezcla de datos con otras especies cultivadas en la misma región experimental.

2.3.10 Requerimientos edafoclimáticos del sector

El óptimo desarrollo de la *Musa paradisiaca* está estrictamente condicionado por los factores edafoclimáticos del entorno. La planta requiere suelos con excelente drenaje, textura franco-arenosa o franco-arcillosa, y rangos de temperatura que oscilen entre los 22C y 30C,

acompañados de una precipitación bien distribuida a lo largo del año para mantener el vigor foliar (Fernández et al., 2021).

Cualquier alteración en estas condiciones genera un impacto directo en el rendimiento agrícola. Estudios de campo de origen local, desarrollado por Quijije et al. (2025), demuestran que someter al plátano a condiciones de estrés hídrico, ya sea por sequía severa o inundación, altera drásticamente su desarrollo agronómico, afectando variables críticas como la altura de la planta y el diámetro del pseudotallo.

Para reducir los riesgos vinculados a esta sensibilidad climática, el sistema informático necesita incorporar el factor ambiental en su registro, siguiendo los principios de modelado propuestos por Carvalho et al. (2025), la aplicación móvil otorgará a los estudiantes la capacidad de relacionar las mediciones biométricas con las condiciones climáticas, creando así un registro de datos que permitirá análisis futuros sobre la adaptabilidad del cultivo en Manabí.

2.3.11 Fenología y fases de desarrollo biológico

La fenología del plátano implica el análisis de diversas etapas de crecimiento biológico que atraviesa la planta a través de su etapa de vida, este ciclo se segmenta en tres etapas principales, la fase vegetativa, correspondiente al desarrollo de hojas y pseudotallo, la etapa reproductiva, caracterizada por la aparición de la inflorescencia o fruto, y la etapa productiva, en la que se forma y se madura el racimo (FAO, 2022).

La supervisión exhaustiva de estas transiciones es esencial para la planificación agrícola contemporánea, según los estudios de Balcázar Daza y Arbeláez (2025), la correcta identificación de las necesidades biotecnológicas en los cultivos potenciales agrícolas demanda que los técnicos entiendan con claridad el estado metabólico del cultivo para realizar intervenciones adecuadas y efectivas.

En el ámbito de la ingeniería de software, la supervisión de estas fases valida completamente la adopción de esquemas polimórficos, siguiendo las recomendaciones de Wu

y Mueller (2024) para marcos de recolección NoSQL, el backend cambiará dinámicamente el esquema de validación dependiendo de la fase fenológica seleccionada; si la planta está en fase vegetativa, el sistema exigirá el número de hojas, pero si entra en fase reproductiva, solicitará el conteo de manos del racimo.

2.3.12 Parámetros biométricos y variables fitotécnicas

Los parámetros biométricos y las variables fitotécnicas constituyen los indicadores numéricos del estado de salud y vigor de la planta. En el manejo agronómico de musáceas, la evaluación integral incluye la medición sistemática de la circunferencia del pseudotallo a una altura estándar, la tasa de emisión foliar, la longitud y ancho de las hojas, y finalmente, el peso del racimo (Quijije et al., 2025).

La estandarización de estas medidas obedece a normativas de control y calidad para garantizar un adecuado manejo productivo. Establecer métricas exactas y libres de ambigüedad es un requisito documentado por Fernández et al. (2021), para detectar a tiempo anomalías en el crecimiento que puedan sugerir la presencia de patógenos o deficiencias nutricionales severas en el área de siembra.

Digitalizar este volumen de variables numéricas en campo presenta un desafío de latencia y almacenamiento. Utilizando las ventajas de las jerarquías documentales analizadas por Ghiwari et al. (2017), el motor de base de datos no relacional desarrollado en el presente proyecto tecnológico procesará las entradas biométricas enviadas desde los dispositivos móviles en tiempo real, garantizando que el análisis estadístico de los parámetros fitotécnicos esté siempre estructurado.

2.3.13 Control fitosanitario y principales enfermedades

La gestión fitosanitaria incluye las acciones tanto preventivas como curativas orientadas a salvaguardar el cultivo frente a la aparición de plagas y enfermedades, en el caso particular de las musáceas, enfermedades de alto riesgo como la marchitez por *Fusarium* o la

Sigatoka Negra representan amenazas biológicas severas, capaces de arruinar cultivos completos si no se supervisan con protocolos estrictos de bioseguridad (FAO, 2022).

El impacto de estas patologías trasciende el ámbito biológico y afecta directamente la estabilidad socioeconómica de las regiones productoras. Según se documenta en los análisis de la estructura agraria realizados por Kepner y Soothill (2022), la vulnerabilidad fitosanitaria exige una constante modernización de las prácticas agrícolas, con el fin de proteger la inversión del sector y sostener los niveles de producción a largo plazo.

En este escenario, la digitalización del estado de salud de la planta es innegociable. Basado en el marco de gestión de datos propuesto por Wu y Mueller (2024), el módulo de control fitosanitario de la aplicación móvil permitirá a los evaluadores adjuntar observaciones textuales y alertas tempranas directamente al documento de cada planta, agilizando la respuesta técnica ante cualquier brote infeccioso en la granja.

2.3.14 Manejo de ensayos agrícolas y experimentación

El manejo de ensayos agrícolas constituye la base empírica de la investigación agronómica, fundamentalmente en la observación sistemática y controlada de variables en el campo. Este proceso metodológico permite a los científicos y estudiantes evaluar la eficacia de nuevos tratamientos, fertilizantes o prácticas de manejo bajo condiciones ambientales reales (Wu & Mueller, 2024).

La transición hacia una experimentación más precisa requiere superar las metodologías tradicionales de registro en papel. Según apuntan Balcázar Daza y Arbeláez (2025), la inserción de herramientas tecnológicas en los potenciales agrícolas es un paso indispensable para minimizar el error humano durante la tabulación de información y estandarizar los protocolos de investigación.

Aplicando estos principios en la Granja Experimental El Carmen, el backend estructurado actuará como el núcleo central de la experimentación. Como se destaca en las

revisiones de modelado de Carvalho et al. (2025), un repositorio validado tecnológicamente garantiza que los datos fitotécnicos recolectados por distintas cohortes de estudiantes mantengan una integridad absoluta, fortaleciendo la calidad de las publicaciones científicas.

2.3.15 Gestión de cosecha y rendimiento agrícola

La gestión de la cosecha marca la culminación del ciclo biológico del plátano y el punto de medición definitiva del rendimiento agrícola. En esta fase, los indicadores de éxito productivo se concentran en el peso bruto del racimo, el número de manos, el diámetro de los dedos y el índice de madurez fisiológica al momento del corte (Fernández et al., 2021).

El rendimiento final de la cosecha está intrínsecamente ligado a los cuidados previos y a la resiliencia de la planta. Investigaciones locales como las de Quijije et al. (2025) evidencian que el peso y la calidad del racimo de *Musa paradisiaca* reflejan directamente el nivel de estrés hídrico y las condiciones agronómicas a las que estuvo sometido el cultivo durante sus fases vegetativas tempranas.

Para procesar este cierre de ciclo, el sistema informático ejecutará operaciones de agregación de datos complejos. Utilizando las bases NoSQL detalladas por Giamas (2022), el software podrá cruzar dinámicamente el peso final del racimo con todo el historial de mediciones guardado, entregando a los docentes reportes automatizados sobre la eficiencia productiva de cada lote experimental.

2.3.16 Importancia socioeconómica y tecnificación del cultivo

La importancia socioeconómica del cultivo de plátano en el territorio ecuatoriano, y específicamente en la provincia de Manabí, es innegable. Más allá de su valor nutricional, esta musácea dinamiza la economía rural, generando fuentes de empleo y posicionándose como un pilar fundamental de la soberanía alimentaria del país (FAO, 2022).

Esta relevancia estructural es el resultado de décadas de evolución agroindustrial. Tal como exponen Kepner y Soothill (2022), en su revisión histórica, las regiones tropicales

latinoamericanas han desarrollado una dependencia intrínseca hacia la eficiencia operativa de estos cultivos, lo que obliga a las instituciones educativas a buscar la mejora continua de sus técnicas agrarias.

Frente a esta realidad, la tecnificación del cultivo a través del desarrollo de software no es una opción, sino una obligación académica. Sustentado en la necesidad de digitalización sistemática expuesta por Carvalho et al. (2025), el sistema informático con esquema polimórfico desarrollado para la Uleam representa un aporte directo que moderniza la captura de datos fitotécnicos, vinculando la ingeniería de software con la tradición agrícola de Manabí.

2.3.17 Metodologías ágiles y el marco de trabajo Scrum en el desarrollo de software

El desarrollo de software contemporáneo ha evolucionado desde los enfoques secuenciales y rígidos como el modelo en cascada hacia paradigmas iterativos e incrementales conocidos como metodologías ágiles. Este enfoque prioriza la adaptabilidad ante los cambios constantes de requerimientos, la entrega continua de valor y la colaboración estrecha con el usuario final o los administradores del producto (Tummers et al., 2019). En el contexto de proyectos de ingeniería orientados a la investigación experimental, la flexibilidad metodológica resulta indispensable para absorber modificaciones imprevistas derivadas de la integración con herramientas externas o requerimientos cambiantes en el campo.

Dentro del ecosistema ágil, el marco de trabajo Scrum se posiciona como una de las estructuras organizativas más eficientes para la gestión y control de proyectos tecnológicos complejos. Scrum organiza el ciclo de vida del software en iteraciones temporales breves denominadas Sprints, permitiendo entregas funcionales parciales denominadas incrementos del producto (Sarasa & Cabezuelo, 2021). Este marco se fundamenta en pilares empíricos como la transparencia, la inspección y la adaptación, distribuyendo las responsabilidades a través de roles definidos como el Product Owner, el Scrum Master y el Scrum Team y apoyándose en artefactos clave como el Product Backlog y el Sprint Backlog (Casciaro & Mammino, 2020).

La adopción de Scrum en el desarrollo de soluciones informáticas agrícolas garantiza que la construcción del software mantenga una sincronización iterativa con las necesidades reales de validación en terreno, reduciendo los riesgos de integración y asegurando la calidad del producto final.

2.4 Conclusiones del marco teórico

El estudio detallado de la literatura técnica indica que los sistemas de gestión de bases de datos NoSQL, especialmente los orientados a documentos, son la infraestructura más adecuada para gestionar información con alta variabilidad, se mostró que la adopción de esquemas polimórficos suprime los inconvenientes de rigidez estructural característicos de los modelos relacionales, facilitando que un único repositorio contenga de manera dinámica métricas diversas sin perjudicar la integridad de los datos, asimismo, el estudio de patrones de diseño contemporáneos confirma que la combinación de una arquitectura hexagonal con la ejecución asíncrona de Node.js y la seguridad estática de TypeScript, garantiza la construcción de un backend escalable, tolerante a fallos y capaz de procesar múltiples operaciones transaccionales en tiempo real.

Por otro lado, la revisión de la literatura agronómica y biológica determina que el cultivo de *Musa paradisiaca* es un sistema vivo de alta complejidad que exige un monitoreo riguroso a lo largo de todas sus fases fenológicas. La obtención de las variables fitotécnicas como el diámetro del pseudotallo, la tasa de emisión de hojas y las métricas de rendimiento del racimo no es un proceso fijo, sino que está fuertemente influenciado por factores edafoclimáticos y riesgos fitosanitarios, se determina que el registro manual o a través de estructuras informáticas rígidas es ineficaz y susceptible a errores humanos, lo que afecta la calidad empírica de las investigaciones agrícolas.

La convergencia de ambas áreas teóricas refuerza la factibilidad técnica y científica del proyecto, se determina que la digitalización de los datos biométricos a través de herramientas

de agricultura 4.0 es una necesidad esencial para la actualización del sector, por ende, la implementación de un sistema informático ágil y clasificado permitirá a la Granja Experimental normalizar sus procedimientos de experimentación en campo, proteger su registro biológico ante eventualidades físicas o de red, y convertir los datos fitotécnicos dispersos en información organizada, favoreciendo de esta manera la toma de decisiones académicas.

CAPÍTULO III

3 MARCO INVESTIGATIVO

3.1 Introducción

Este capítulo detalla el diseño metodológico que guía la investigación, definiendo la ruta científica y los procedimientos necesarios para enfrentar la problemática de la recolección y manejo de datos fitotécnicos en la Granja Experimental, extensión El Carmen, la definición precisa de este marco investigativo es esencial para asegurar que la recolección de datos en el terreno satisfaga los criterios académicos de validez y confiabilidad requeridos en un proyecto de ingeniería.

Para lograr los objetivos planteados en la conceptualización del módulo informático y su esquema polimórfico, se detalla a continuación el enfoque epistemológico del estudio, especificando el tipo y los métodos teóricos y empíricos seleccionados. Así mismo, se delimitan las fuentes de información primarias y secundarias que nutrirán el análisis, junto con la población y muestra de estudio conformada por los actores directamente involucrados en el cultivo de la Musa paradisiaca.

Finalmente, este apartado describe las estrategias y los instrumentos de recolección de datos que se aplicarán en el entorno agrícola, así como las técnicas de procesamiento estadístico y lógico que permitirán transformar los requerimientos de los usuarios en especificaciones técnicas claras, sentado así las bases empíricas para el posterior desarrollo de la solución tecnológica.

3.2 Tipo y métodos de investigación

3.2.1 Tipo de investigación

La presente investigación se desarrolla bajo un enfoque mixto, al combinar el análisis cuantitativo de la encuesta con el análisis cualitativo de la entrevista, se enmarca dentro de la tipología aplicada y explicativa. Es aplicada porque utiliza conocimientos de ingeniería de

software para responder a una necesidad concreta, mejorar la recolección, organización y consulta de datos fitotécnicos en la Granja Experimental ULEAM El Carmen, mediante el desarrollo de un sistema informático orientado a resultados prácticos en la realidad empírica (Chávez Abad, 2015). A su vez, es explicativa porque analiza las causas del problema, como el registro manual, la dispersión de la información agrícola y la dificultad para conservar historiales confiables, para luego plantear una solución basada en almacenamiento local, sincronización de datos y esquemas polimórficos

3.2.2 Investigación Aplicada

La investigación aplicada es aquella que se caracteriza por buscar la resolución de problemas prácticos y específicos utilizando los conocimientos teóricos existentes, orientándose a generar resultados empíricos, soluciones tangibles o innovaciones tecnológicas que impacten directamente en la realidad estudiada (Chávez Abad, 2015). Este nivel de investigación se utilizó de manera directa en el proyecto al tomar los principios abstractos de la ingeniería de software y las arquitecturas de bases de datos NoSQL, para dar una respuesta práctica a la deficiencia organizativa de la Granja Experimental. El resultado de esta aplicación tecnológica se materializó en la construcción y despliegue del sistema informático AgroAR, resolviendo el problema de los registros manuales a través de una herramienta funcional y escalable.

3.2.3 Investigación Explicativa

Este tipo de investigación busca ir más allá de la simple descripción de un problema, enfocándose en encontrar, analizar y explicar las causas, razones y los efectos que provocan un fenómeno específico, estableciendo relaciones de causalidad claras (Arias y Covinos, 2021). En este trabajo, la investigación explicativa se empleó al analizar detalladamente las causas fundamentales que originaban la pérdida de trazabilidad agrícola, identificando que el registro en papel y la dispersión en archivos de Excel generaban cuellos de botella y vulnerabilidad

ante errores humanos. A partir de la comprensión exacta de estas causas y de la limitación del internet rural, se pudo justificar y establecer arquitectónicamente la necesidad de implementar un modelo de almacenamiento local basado en SQLite y un motor de base de datos con esquemas polimórficos.

3.2.4 Métodos de investigación

Para el desarrollo del estudio se utilizaron métodos teóricos y empíricos. Dentro de los métodos teóricos se aplicó el método analítico-sintético, el cual permite descomponer un fenómeno en sus partes principales para analizarlas por separado y luego integrarlas en una comprensión general del problema (Chávez Abad, 2015). En esta investigación, este método permitió estudiar el registro de datos agrícolas, las variables fitotécnicas del cultivo de plátano y las herramientas tecnológicas utilizadas en el prototipo, para después relacionar estos elementos en el diseño del sistema AgroAR.

Como método empírico se utilizó la investigación de campo, ya que permitió recoger información directa de los actores relacionados con la Granja Experimental. Esta información ayudó a comprender cómo se realiza actualmente el registro de datos, qué dificultades existen en el trabajo agrícola y qué requerimientos debía considerar el sistema para responder mejor al contexto real.

3.2.5 Método Analítico-Sintético

Este método constituye un procedimiento lógico que permite, por un lado, descomponer un fenómeno complejo en sus partes principales para examinarlas de manera individual (análisis) y, por otro lado, integrar esos elementos analizados de forma coherente para obtener una visión general y resolutive (síntesis) (Chávez Abad, 2015). En el marco de esta investigación, el método se utilizó inicialmente para estudiar de forma aislada las distintas variables fitotécnicas del cultivo de plátano, las carencias del trabajo en el terreno y las capacidades de tecnologías como Node.js y MongoDB. Posteriormente, durante la fase

propositiva, se realizó la síntesis al relacionar y ensamblar todas estas partes dispersas para diseñar la arquitectura final del sistema AgroAR, logrando que el software respondiera íntegramente al contexto agrícola evaluado.

3.2.6 Método Empírico

La investigación de campo consiste en la recolección de datos primarios directamente en el lugar donde ocurren los hechos o interactúan los sujetos de estudio, sin manipular ni controlar variables externas, lo que garantiza un alto nivel de fidelidad en la información recolectada (Zúñiga et al., 2023). Este método empírico se llevó a cabo al trasladar el levantamiento de información directamente al entorno real de la Granja Experimental ULEAM Extensión El Carmen. Al aplicar los instrumentos de recolección a la muestra de 40 estudiantes y al docente en su propio espacio de prácticas agrícolas, se logró comprender de primera mano cómo se realizaba el registro manual y cuáles eran las verdaderas limitaciones operativas en el lote, transformando estas observaciones empíricas en los requerimientos técnicos y funcionales que definieron el sistema.

3.3 Fuentes de información

3.3.1 Fuentes primarias

3.3.1.1 Encuestas

La encuesta es una técnica de recolección de información que permite obtener datos de un grupo de personas mediante preguntas previamente definidas. En esta investigación se utilizó para conocer la percepción de los participantes sobre el uso de una aplicación móvil para el registro, análisis y seguimiento de revisiones realizadas al cultivo de plátano.

El cuestionario estructurado se utilizó para llevar a cabo la encuesta, con preguntas enfocadas en determinar el grado de aceptación del prototipo, la necesidad de tener un historial de revisiones, la utilización de reportes en PDF y la conveniencia de incluir análisis fundamentados en la inteligencia artificial, la información obtenida posibilitó identificar

elementos importantes para el diseño y la validación del sistema, además, se estructuró con preguntas cerradas de opción múltiple y escala Likert de cinco puntos, totalmente de acuerdo a totalmente en desacuerdo, conformado por un total de 18 preguntas.

3.3.1.2 Entrevista

La entrevista es un método que facilita la obtención de información más específica mediante la conversación con individuos relacionados con el tema de investigación, en este estudio se utilizó como base para ahondar en las necesidades del sistema y entender más claramente los procesos que debían tenerse en cuenta en el prototipo. La entrevista se compuso de tres preguntas abiertas, organizadas en torno a tres ejes temáticos, el proceso actual de registro de datos, las variables fitotécnicas manejadas, y los requerimientos técnicos esperados del sistema.

Esta metodología fue utilizada en un profesor experto, el Ing. Paul González, vinculado a la gestión y el control técnico de la Granja Experimental, las preguntas se centraron en entender cómo se registran en la actualidad las variables fitotécnicas, qué cultivos y parámetros se manejan en el terreno, cuáles son las limitaciones del proceso manual de recopilación de datos, y qué grado de flexibilidad debe tener el sistema para adaptarse a nuevas especies o áreas de investigación.

La información obtenida permitió relacionar las necesidades reales de la granja con decisiones técnicas del prototipo, como el uso de almacenamiento offline, sincronización de datos y una base NoSQL con estructuras flexibles.

3.3.2 Fuentes secundarias

Las fuentes secundarias corresponden a la información obtenida de libros, artículos científicos, tesis, documentos académicos y material técnico relacionado con el tema de investigación. Estas fuentes sirvieron para construir la base teórica del proyecto y sustentar conceptos relacionados con el desarrollo móvil, bases de datos SQL y NoSQL, metodología

Scrum, inteligencia artificial aplicada a la agricultura y enfermedades o plagas que afectan al cultivo de plátano.

La revisión de estas fuentes permitió fundamentar técnicamente el desarrollo del prototipo, comparar la propuesta con trabajos previos y comprender las herramientas utilizadas en el sistema, como Flutter, SQLite, MongoDB Atlas, Cloudinary, Render y los modelos de inteligencia artificial integrados.

3.4 Estrategia de recolección de datos

La estrategia de recolección de datos define los procedimientos, la población de estudio y los instrumentos que materializan el levantamiento empírico de la información. Esta estructura metodológica es indispensable para transformar las necesidades de la Granja Experimental en requerimientos técnicos y funcionales para el sistema informático.

Tabla 1: Plan de recolección de datos.

Fecha de inicio	Fecha de fin	Actividad	Responsable
14 de abril del 2026	15 de abril del 2026	Aplicación de la encuesta	Zamora Loor Jose Manuel
16 de abril del 2026	16 de abril del 2026	Aplicación de la entrevista	Zamora Loor Jose Manuel

3.4.1 Población y muestra

3.4.1.1 Población

La población del estudio estuvo conformada por 43 estudiantes vinculados con las prácticas agrícolas desarrolladas en la Granja Experimental Uleam Extensión El Carmen. Estos estudiantes fueron considerados porque representan a los futuros usuarios operativos del sistema, al participar en actividades relacionadas con la observación, registro y seguimiento del cultivo de plátano. Además, se consideró como informante clave a un docente especialista

de la Granja Experimental, debido a su conocimiento sobre las variables fitotécnicas y los procesos actuales de registro.

3.4.1.2 Muestra

El muestreo no probabilístico por conveniencia es una técnica que consiste en seleccionar a los individuos de estudio en función de su accesibilidad, presencia y disponibilidad inmediata para el investigador en el momento exacto de la recolección de los datos, descartando a aquellos sujetos que no pueden ser contactados (Zúñiga et al., 2023).

En la presente investigación, se aplicó un muestreo no probabilístico por conveniencia para definir una muestra final de 40 estudiantes, a partir de la población finita de 43. No se ejecutó una muestra censal debido a que 3 estudiantes no asistieron a las prácticas agrícolas en la Granja Experimental durante los días 14 y 15 de abril de 2026, fechas exactas en las que se llevó a cabo la aplicación de la encuesta estructurada mediante Google Forms. Por consiguiente, se trabajó exclusivamente con los 40 individuos que mostraron total accesibilidad y participación directa durante las jornadas de evaluación en el lote agrícola.

3.4.2 Técnicas e instrumentos de recolección

Para la recolección de información se utilizó la encuesta como técnica principal, aplicada mediante un cuestionario estructurado digital en Google Forms. Esta encuesta se realizó los días 14 y 15 de abril de 2026 a los estudiantes vinculados con las prácticas agrícolas de la Granja Experimental ULEAM Extensión El Carmen. El instrumento permitió obtener datos organizados sobre la percepción de los estudiantes respecto al registro manual de información agrícola, las dificultades al manejar datos variables en campo y la aceptación de una aplicación móvil conectada a una base de datos documental.

La entrevista semiestructurada se empleó como técnica cualitativa, aplicada el día 16 de abril de 2026 a un docente especialista de la Granja Experimental. La guía de preguntas permitió profundizar en los procesos actuales de registro, las variables fitotécnicas utilizadas,

las limitaciones del trabajo en campo y los criterios técnicos necesarios para orientar la estructura del backend y el modelo de almacenamiento propuesto. Además, el instrumento fue revisado y validado por el tutor del trabajo de titulación antes de su aplicación, con el fin de garantizar la claridad y pertinencia de las preguntas respecto a los objetivos de la investigación.

3.5 Análisis e interpretación de resultados

El procesamiento de la información obtenida a través de las encuestas aplicadas a los estudiantes se fundamentará en el análisis cuantitativo mediante estadística descriptiva. Este procedimiento metodológico consiste en tabular los datos brutos, calcular sus frecuencias relativas o absolutas, y proyectarlos en gráficos que faciliten la comprensión visual de los patrones de respuestas (Zúñiga et al., 2023). La interpretación de estas métricas permitirá cuantificar objetivamente la deficiencia del actual registro manual de datos fitotécnicos y medir el nivel de viabilidad operativa para la inserción tecnológica en la Granja Experimental extensión El Carmen.

Por otro lado, la información extraída de las entrevistas a los docentes exigirá un procesamiento de corte estrictamente cualitativo, Arias y Covinos (2021). El análisis cualitativo busca comprender la profundidad técnica y el significado de los testimonios expertos, más allá de su mera frecuencia numérica. En este caso, las respuestas de los especialistas agrónomos serán decodificadas y categorizados lógicamente para identificar las reglas de validación biológica del cultivo (*Musa paradisiaca*), transformando su conocimiento empírico en los esquemas y jerarquías que exigirá la base de datos documental.

Finalmente, la fase de interpretación consolidará los hallazgos de ambos instrumentos mediante una triangulación de resultados. Los datos estadísticos de los estudiantes validarán la necesidad de la interfaz y la experiencia de usuario, mientras que el conocimiento experto de los docentes estructurará el comportamiento interno del backend. Esta convergencia analítica se constituirá como el insumo principal para el levantamiento definitivo de los requerimientos

funcionales y no funcionales, marcando la transición metodológica hacia la fase propositiva y el diseño arquitectónico de la solución informática.

3.5.1 Análisis cuantitativo: Encuesta a estudiantes

Tabla 2: Resultados e interpretación de la encuesta aplicada a estudiantes.

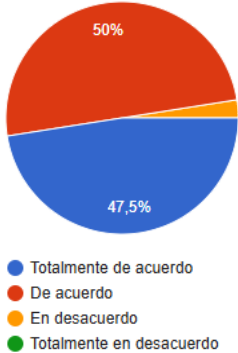
Gráfico	Preguntas	Justificación
 50% 47,5% - Totalmente de acuerdo - De acuerdo - En desacuerdo - Totalmente en desacuerdo	¿Crees que la implementación de tecnologías de “Agricultura de precisión” mejoraría la toma de decisiones fitosanitarias en la Granja experimental de la Uleam?	Un grupo abrumador dentro de los encuestados tiene la necesidad de modernizar los procesos agrícolas. Al agrupar las respuestas favorables, se evidencia que la casi totalidad de los estudiantes cree que la introducción de herramientas de agricultura de precisión optimizaría las decisiones fitosanitarias en el campo. Esto demuestra directamente la validación y necesidad de desarrollar este proyecto de software, indicando que la transición del registro manual a un sistema integrado con una base de datos documental no es un

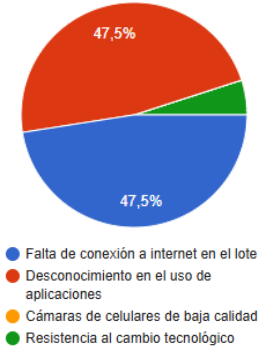
Gráfico	Preguntas	Justificación
		requisito teórico, sino una necesidad operativa práctica planteada por los usuarios para proteger la salud de su cultivo de Musa paradisiaca.
 ● Falta de conexión a internet en el lote ● Desconocimiento en el uso de aplicaciones ● Cámaras de celulares de baja calidad ● Resistencia al cambio tecnológico	¿Cuál consideras que es el principal obstáculo tecnológico para que los estudiantes o productores utilicen una app de este tipo en El Carmen?	Los resultados señalan dos obstáculos principales, empatados en magnitud, que dificultan las operaciones agrícolas: la falta de conexión a internet en el lote y el desconocimiento en el uso de aplicaciones. Con esto se evidencian requisitos no funcionales importantes durante el desarrollo: tener acceso a la aplicación de manera offline con total funcionamiento que permita el almacenamiento local temporal y sincronización bidireccional, y, asimismo, una interfaz de usuario extremadamente simple e

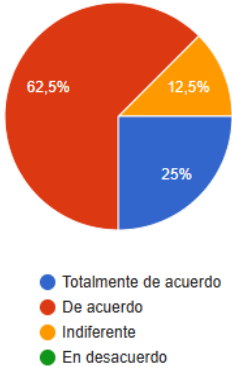
Gráfico	Preguntas	Justificación
		intuitiva que garantice el uso sencillo frente a la minoría que presenta resistencia al cambio tecnológico.
 ● Totalmente de acuerdo ● De acuerdo ● Indiferente ● En desacuerdo	¿Estarías de acuerdo en que un sistema tecnológico recomiende acciones específicas basadas en el análisis de una fotografía, en lugar de depender del criterio humano?	Se observa una alta aceptación de las herramientas tecnológicas automatizadas de diagnóstico para el campo. La gran mayoría de los encuestados está a favor de que el software recomiende acciones específicas basadas en evidencias fotográficas, considerando oportuna la reducción de la dependencia exclusiva del criterio y la experiencia empírica humana para tomar acciones correctivas.

Gráfico	Preguntas	Justificación
 A pie chart illustrating the responses to the question '¿Qué tan importante considera el uso de datos digitales (como registros o monitoreo) en el manejo de cultivos dentro de la granja?'. The chart shows that 52.5% of respondents consider it 'Muy importante' (Very important), 45% consider it 'Importante' (Important), 2.5% consider it 'Poco importante' (Not very important), and 0% consider it 'Nada importante' (Not important at all). A legend below the chart identifies the colors: blue for 'Muy importante', red for 'Importante', yellow for 'Poco importante', and green for 'Nada importante'.	¿Qué tan importante considera el uso de datos digitales (como registros o monitoreo) en el manejo de cultivos dentro de la granja?	Observando los resultados, se determina un alto nivel de madurez y conocimiento tecnológico por parte de los futuros usuarios. La tendencia casi absoluta demuestra que los encuestados consideran que los apuntes y registros digitales son fundamentales e indispensables para la administración agrícola moderna. Todo esto apunta a la necesidad de implementar un sistema de bases de datos robusto como el modelo documental, para garantizar la integridad, persistencia y disponibilidad de la información técnica de la Musa paradisiaca.

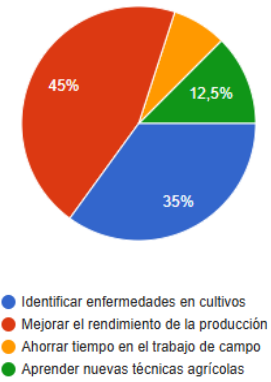
Gráfico	Preguntas	Justificación
 ● Identificar enfermedades en cultivos ● Mejorar el rendimiento de la producción ● Ahorrar tiempo en el trabajo de campo ● Aprender nuevas técnicas agrícolas	¿Qué beneficio le gustaría obtener principalmente de una aplicación agrícola en sus actividades prácticas?	Los resultados reflejan las expectativas operativas de los usuarios finales, quienes priorizan la utilidad agrícola orientada a resultados directos. La opción de mayor interés es la búsqueda de mejoras en el rendimiento de la producción, seguida de cerca por la necesidad de identificar enfermedades en los cultivos. Estos datos respaldan firmemente la necesidad de integrar la interfaz móvil y los módulos funcionales para la salud vegetal del plátano; la tecnología propuesta aportará un valor biológico real a las explotaciones agrícolas, más allá de la simple digitalización rápida de datos.

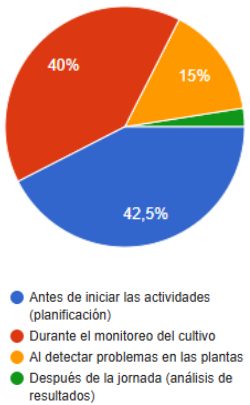
Gráfico	Preguntas	Justificación
 ● Antes de iniciar las actividades (planificación) ● Durante el monitoreo del cultivo ● Al detectar problemas en las plantas ● Después de la jornada (análisis de resultados)	¿En qué momento cree que sería más útil utilizar una aplicación agrícola durante el trabajo en la granja?	Los datos indican que las expectativas de uso del sistema se dividen casi a partes iguales entre dos momentos clave: la fase de planificación antes de iniciar las actividades y la fase del monitoreo directo del cultivo en pleno trabajo de campo. Esta alta necesidad de interacción simultánea y preventiva resalta la importancia técnica de adoptar la arquitectura offline-first junto con una base de datos local SQLite, que permita tanto la capacitación previa (como el uso de realidad aumentada) como el registro in situ sin requerir conexión a la red.

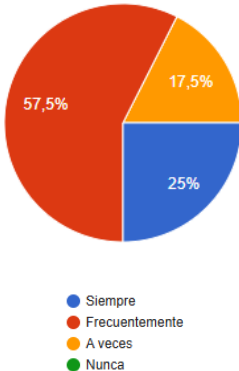
Gráfico	Preguntas	Justificación
 ● Siempre ● Frecuentemente ● A veces ● Nunca	¿Qué tan seguido necesitan tomar diferentes datos a la vez, como medir el tallo, tomar fotos o hacer apuntes, para saber cómo está la planta de plátano?	Los resultados muestran una alta exigencia de capacidad de procesamiento unificado para adquirir datos heterogéneos durante las labores. Al sumar las categorías de mayor constancia, se evidencia que la amplia mayoría de los encuestados manifiesta la necesidad frecuente de registrar de forma simultánea variables numéricas, biométricas y de texto. Esto respalda directamente la adopción de un motor de base de datos NoSQL con un esquema polimórfico, lo que mejora considerablemente las funciones de escritura y almacenamiento en los dispositivos móviles en el campo, al permitir guardar estructuras dinámicas y

Gráfico	Preguntas	Justificación										
		variadas en un solo documento.										
● Falta de herramientas tecnológicas adecuadas ● Dificultad para organizar la información ● Falta de conocimiento en el manejo de sistemas digitales ● No presenta dificultades <table><thead><tr><th>Categoría</th><th>Porcentaje</th></tr></thead><tbody><tr><td>Falta de herramientas tecnológicas adecuadas</td><td>42,5%</td></tr><tr><td>Dificultad para organizar la información</td><td>32,5%</td></tr><tr><td>Falta de conocimiento en el manejo de sistemas digitales</td><td>17,5%</td></tr><tr><td>No presenta dificultades</td><td>7,5%</td></tr></tbody></table>	Categoría	Porcentaje	Falta de herramientas tecnológicas adecuadas	42,5%	Dificultad para organizar la información	32,5%	Falta de conocimiento en el manejo de sistemas digitales	17,5%	No presenta dificultades	7,5%	Al momento de pasar toda esta información (fotos, medidas y notas) a los apuntes o cuadernos que ya lleva, ¿qué es lo que más le cuesta trabajo o le da problema?	La encuesta demuestra claramente la ineficiencia de la entrada manual de datos en el campo. Una porción mayoritaria de los encuestados reportó problemas directos al intentar transcribir y estructurar datos, apuntando a la dificultad para organizar la información y a la falta de herramientas tecnológicas adecuadas como los obstáculos predominantes. El sistema sugerido resuelve esto al disminuir la entrada manual y minimizar el error humano, centralizando y vinculando automáticamente las variables biométricas con los registros multimedia desde el móvil.
Categoría	Porcentaje											
Falta de herramientas tecnológicas adecuadas	42,5%											
Dificultad para organizar la información	32,5%											
Falta de conocimiento en el manejo de sistemas digitales	17,5%											
No presenta dificultades	7,5%											

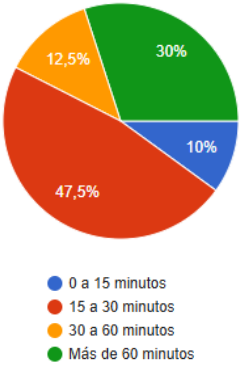
Gráfico	Preguntas	Justificación
 ● 0 a 15 minutos ● 15 a 30 minutos ● 30 a 60 minutos ● Más de 60 minutos	Si tiene fotos por un solo lado y los apuntes por otro, ¿Cuánto tiempo se demora en juntar todo para saber qué ha pasado con una planta o un lote?	Estos datos reflejan un considerable retraso en las operaciones, evidenciando el cuello de botella que representa el registro analógico. Sumando a quienes tardan rangos de tiempo moderados y altos, se observa que la inmensa mayoría de los usuarios pierde un tiempo muy valioso intentando vincular las variables numéricas con las evidencias fotográficas. Mediante el sistema propuesto y el uso de esquemas polimórficos, se asocian automáticamente los registros fotográficos y los datos biométricos dentro del mismo documento en el momento exacto de la captura, reduciendo el

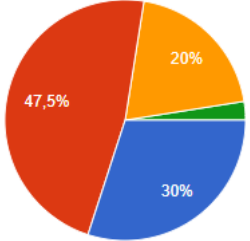
Gráfico	Preguntas	Justificación
		tiempo de vinculación de datos a cero.
 ● Mejor toma de decisiones ● Ahorro de tiempo ● Mayor organización de la información ● Mayor precisión en el análisis	Si tuviera una aplicación que junte solo las fotos y las medidas de cada planta de plátano, y le dé el resumen listo, ¿En qué cree que le ayudaría más en su trabajo?	Estos resultados respaldan el valor operativo y la capacidad de resolución de problemas de este proyecto tecnológico. La principal expectativa de los usuarios es el ahorro de tiempo, posicionándose como la prioridad número uno, seguida inmediatamente por la mejora en la toma de decisiones. Esto indica que la verdadera contribución del sistema informático reside no solo en la digitalización, sino en la capacidad del backend para integrar variables heterogéneas y generar resúmenes estructurados al instante. Esto demuestra la idoneidad del esquema polimórfico de MongoDB,

Gráfico	Preguntas	Justificación
		que permite realizar consultas e integrar información para cada planta en un único ciclo de lectura, optimizando así el rendimiento general de la aplicación.

3.5.2 Análisis cualitativo: Entrevista a experto

Para complementar el análisis y establecer los requisitos lógicos del sistema informático, se utilizó una guía de entrevista semiestructurada con un docente especialista encargado de la gestión de la Granja Experimental, dada la naturaleza técnica y cualitativa de este recurso, los resultados no se presentan con frecuencias estadísticas, sino que se estructuran mediante matrices interpretativas que relacionan las necesidades agronómicas del campo con las soluciones de software y mantenimiento de datos sugeridos..

Tabla 3: Resultados e interpretación de la entrevista al experto.

Ítem/Pregunta	Respuesta del Experto	Interpretación
¿Cuáles son las variables fitotécnicas que se miden actualmente en los diferentes cultivos de la granja experimental y cómo difieren estas métricas entre una especie y otra?	En la actualidad el análisis se concentra principalmente en el cultivo del plátano, considerando variables como el perímetro del pseudotallo, la cantidad de hojas funcionales, el peso del	La información que brinda el experto revela la variabilidad presente en los datos agronómicos administrados en la finca, desde el punto de vista de la ingeniería de software, esta característica

Ítem/Pregunta	Respuesta del Experto	Interpretación
	racimo y el número de manos, sin embargo, la finca también dispone de espacios dedicados a la agricultura de chocolate y naranjas, en lo que respecta al cacao, los criterios de evaluación varían de manera significativa, ya que se examinan elementos como el grosor del tronco principalmente, el número de mazorcas generadas por el árbol y la existencia de moniliasis, estas discrepancias ponen de manifiesto la necesidad de utilizar estructuras de registro flexibles, puesto que no es adecuado utilizar un mismo formato de análisis de una musácea y de un árbol frutal, teniendo en cuenta sus diversas características morfológicas y producción.	complica la utilización de bases de datos relacionales con esquemas definidos, dado que podrían surgir múltiples campos sin datos o ser necesarias tablas extras. Para representar cada tipo de cultivo, en consecuencia, se justifica la implementación de una base de datos NoSQL, específicamente MongoDB, apoyada en esquemas polimórficos. Este enfoque permite almacenar y gestionar diferentes estructuras de datos biométricos dentro de una colección centralizada, manteniendo la coherencia e integridad de la información.

Ítem/Pregunta	Respuesta del Experto	Interpretación
Pensando en el futuro y la experimentación, ¿Con qué frecuencia se introducen nuevas especies de plantas o se modifican los parámetros de estudio, y qué nivel de flexibilidad espera el sistema para registrar estos nuevos datos?	Debido al carácter experimental y universitario de la granja, las necesidades de recolección de datos cambian constantemente. En cada período académico o proyecto de titulación pudieron incorporarse nuevos cultivos, como pitahaya o maracuyá, además de variables que antes no eran consideradas, entre ellas el índice de verdor foliar o la humedad específica del suelo en la zona radicular. Por esta razón el sistema debe contar con una estructura completamente flexible. Depender de un desarrollador para crear nuevas tablas cada vez que se requiera registrar un parámetro distinto provocaría que la solución	La afirmación del entrevistado indica que las condiciones y exigencias de la granja cambian de forma continua. Este contexto apoya la creación de un backend utilizando Node.js y TypeScript, fundamentado en una arquitectura escalable que puede manejar objetos JSON flexibles y adaptables. A través de la aplicación de modelos flexibles, como los discriminadores de Mongoose, el sistema tiene la capacidad de añadir nuevas especies de plantas y parámetros fitotécnicos sin requerir cambios frecuentes en la lógica del servidor. De este modo, se promueve la mantenibilidad, adaptabilidad y actualidad de

Ítem/Pregunta	Respuesta del Experto	Interpretación
	quedara desactualizada en poco tiempo. Por lo tanto, es necesaria una plataforma que pueda crear y modificar formularios de forma dinámica, según las demandas de cada estudio.	la solución tecnológica a largo plazo.
¿Cómo es el flujo actual de recolección, almacenamiento y análisis de la información fitotécnica en campo, y cuáles son las principales deficiencias, errores o cuellos de botella que enfrenta con este método?	La recogida de información se lleva a cabo de manera manual. Los alumnos se trasladan al campo con formularios impresos, pizarras y lápices para recoger los datos directamente en el lugar. Estas condiciones someten los documentos a la luz solar y la lluvia, lo que puede dañarlos o dificultar la legibilidad de la información contenida. Después, los registros tienen que ser llevados al laboratorio o a la oficina para que sean escritos	El análisis permite reconocer dos dificultades principales en el proceso vigente: la ausencia de conectividad en las áreas de cultivo y la duplicación en el registro de información, primero en formularios físicos y luego en archivos de Excel. Esta situación sustenta la implementación de una aplicación móvil basada en una arquitectura Offline-First. El uso de una base de datos local integrada, como SQLite, facilita el almacenamiento de la

Ítem/Pregunta	Respuesta del Experto	Interpretación
	en hojas de cálculo de Excel. La repetición de tareas es el problema central del proceso, ya que requiere numerosas horas de trabajo y eleva considerablemente la probabilidad de incurrir en errores al momento de digitación. Asimismo, la escasez de conectividad en áreas rurales ha restringido la utilización de plataformas comerciales en la nube, lo que perpetúa la dependencia de los documentos en papel.	información aun cuando no exista acceso a internet, mientras que el mecanismo de sincronización asíncrona permite transferir los datos al servidor una vez restablecida la conexión. Con ello, se reducen los errores asociados a la transcripción manual y se optimizan considerablemente los tiempos destinados a la organización y el procesamiento de los registros académicos

CAPÍTULO IV

4 MARCO PROPOSITIVO

4.1 Introducción al capítulo

Este capítulo aborda los elementos fundamentales de la ingeniería y la solución de problemas del análisis. Empezando con los análisis cuantitativos y cualitativos vistos en el capítulo anterior que evidencian las faltas del registro manual de datos agrícolas y afirman la necesidad técnica, este capítulo detalla la conceptualización, el diseño y la arquitectura de la solución de software propuesta. El objetivo principal de este marco es traducir las necesidades recabadas de los participantes en la granja piloto en una arquitectura de software robusta y escalable, adaptada a las condiciones operativas reales.

Para lograrlo, la propuesta adopta una arquitectura hexagonal, basada en los principios más recientes de ingeniería de software. Este modelo arquitectónico garantiza una clara separación de responsabilidades entre la lógica de negocio central para el cultivo de banano y las interfaces con la infraestructura externa. Además, el diseño incorpora un enfoque Mobile First Offline, respaldado por un backend desarrollado con un entorno de ejecución eficiente (como Node.js).

4.2 Descripción de la propuesta tecnológica

La propuesta tecnológica se fundamenta en el diseño y desarrollo de un sistema informático centralizado, cuyo propósito principal es actuar como el motor de persistencia y modelado de datos para el ecosistema de capacitación agrícola de la Granja Experimental. A continuación, se detallan los lineamientos estructurales, lógicos y metodológicos que rigen su construcción.

4.2.1 Arquitectura del software

El diseño del sistema emplea una arquitectura híbrida orientada a servicios, dividida lógicamente entre la infraestructura en la nube y el cliente móvil. En el backend del servidor,

se aplica una arquitectura hexagonal con puertos y adaptadores, este patrón aísla la lógica de negocio (las reglas agronómicas de la Musa paradisiaca) de las herramientas externas, permitiendo que el sistema reciba datos de múltiples fuentes como los modelos de inteligencia artificial o el Chatbot sin alterar el núcleo de la base de datos.

Desde la perspectiva del cliente móvil utilizado en campo, la arquitectura adopta un enfoque estrictamente “Offline-First”. Dada la comprobada falta de conectividad a internet en los terrenos agrícolas, el sistema enfatiza la lectura y escritura en una base de datos local integrada en el dispositivo, las transacciones se almacenan de forma segura y temporal en el equipo del usuario hasta que se detecta una red estable, momento en el cual un proceso en segundo plano asume la responsabilidad de transferir los registros hacia el servidor central.

4.2.2 Descripción de módulos principales

El sistema se compone de cuatro módulos fundamentales que garantizan el flujo, la flexibilidad y la persistencia de la información fitotécnica:

- Módulo de gestión de accesos y autenticación: Encargado de la seguridad transversal del sistema. Define y restringe las operaciones mediante el control de roles de usuario, diferenciando los privilegios entre el personal docente y los estudiantes.
- Módulo de modelado fitotécnico polimórfico: Constituye el núcleo central de la investigación. Este módulo gestiona una colección documental con estructuras dinámicas. Es capaz de almacenar registros morfológicamente distintos en una misma entidad; por ejemplo, procesando datos booleanos simples cuando los módulos externos de IA detectan Sigatoka Negra o Picudo Negro, y almacenando estructuras matemáticas extendidas al recibir el porcentaje calculado de la Salud Vegetal del plátano.
- Módulo de historial y mensajería: Un repositorio diseñado para persistir las conversaciones, interacciones y sugerencias agronómicas generadas por el asistente virtual ChatBot, garantizando la trazabilidad del soporte técnico brindado al estudiante.

- **Módulo de sincronización asíncrona:** El sistema encargado de la armonización de datos, monitorea el estado de la red y transfiere ordenadamente los registros desde la persistencia local del dispositivo hacia la base de datos en la nube, garantizando la integridad referencial y evitando la pérdida de información de las prácticas agrícolas.

4.2.3 Tecnologías claves a utilizar

- **Node.js y TypeScript:** El entorno de ejecución del backend es Node.js, utilizando su capacidad de asincronía para gestionar múltiples solicitudes de los módulos de IA al mismo tiempo, TypeScript se utiliza para ofrecer un tipado riguroso, disminuyendo significativamente los errores durante el desarrollo al establecer interfaces precisas para los modelos de datos.
- **MongoDB y MongoDB Compass:** MongoDB es el gestor de bases de datos no relacional NoSQL más importante, su característico enfoque basado en documentos JSON/BSON es crucial para concretar el esquema polimórfico, deshaciendo la inflexibilidad de las tablas relacionales, MongoDB Compass se utiliza como la interfaz gráfica oficial para administrar, consultar y validar localmente el comportamiento de colecciones.
- **SQLite:** Funciona como el motor de base de datos relacional compacto y que está integrado en el cliente móvil. Su escaso uso de recursos y gran compatibilidad lo establecen como la solución perfecta para asegurar el almacenamiento temporal offline-first en el sitio.
- **Visual Studio Code:** Elegido como el entorno de desarrollo integrado (IDE) principal. Su extenso ecosistema de extensiones, compatibilidad nativa con TypeScript y terminal incorporada mejoran los tiempos de codificación, depuración y gestión de versiones del proyecto.

4.2.4 Modelo de desarrollo de software seleccionado y justificado

Para la construcción del sistema se adoptó la metodología de trabajo ágil Scrum. La elección de esta metodología se justifica por la naturaleza colaborativa e interdependiente del proyecto integrador, al segmentar el ciclo de vida del software en iteraciones breves, Scrum proporcionó la agilidad requerida para ajustar y cambiar las configuraciones de la base de datos conforme los modelos de inteligencia artificial y el ChatBot de otros módulos establecían sus formatos de salida JSON, este método incremental y repetitivo aseguró que el sistema de backend y la consistencia de los datos permanecieran alineados con el avance de los módulos de inteligencia artificial y del ChatBot, desarrollados de manera independiente por compañeros de proyectos de titulación afines, disminuyendo los riesgos de integración final.

4.3 Recursos requeridos

La implementación de la tecnología propuesta, en particular el desarrollo de un sistema informático con arquitectura multimodal, requirió numerosos recursos esenciales. A continuación, se detallan los recursos humanos, técnicos y financieros que hicieron posible la programación, las pruebas y el despliegue del proyecto.

4.3.1 Recursos humanos

Al tratarse de un desarrollo de software altamente técnico y enfocado en la estructuración de base de datos no relacionadas, la ejecución del proyecto recae operativamente sobre un único perfil encargado de la ingeniería, diseño y programación. No obstante, este sistema informático no operó de forma aislada, sino que se concibió como el motor central de almacenamiento dentro de un macro-proyecto universitario enfocado en el cultivo de plátano. Por esta razón, la construcción del ecosistema tecnológico general requirió una coordinación técnica y sincronización constante con un equipo de investigación paralelo (conformado por compañeros de la misma carrera), quienes desarrollaron de manera independiente los modelos de Inteligencia Artificial y el motor del Chatbot agrícola. De este modo, aunque el recurso

humano interno del presente sistema fue individual, la metodología de trabajo exigió una dinámica de integración colaborativa externa.

Tabla 4: Recursos humanos requeridos para el proyecto tecnológico

Denominación	Características	Cantidad
Desarrollador principal.	Estudiante de la carrera de Ingeniería en Software encargado de la arquitectura, diseño de los esquemas polimórficos, codificación del backend y pruebas locales.	1
Equipo de integración	Compañeros de proyectos de titulación afines, responsables de proveer los endpoints y servicios externos correspondientes a los modelos de Inteligencia Artificial y el Chatbot agrícola.	5

4.3.2 Recursos tecnológicos

El desarrollo exigió hardware con capacidad suficiente para ejecutar entornos de programación y bases de datos en simultáneo, así como dispositivos móviles para la validación del modelo “Offline-First”. A nivel lógico, se emplearon herramientas de código abierto y capas gratuitas.

Tabla 5: Recursos tecnológicos (Hardware y Software) requeridos.

Denominación	Características	Cantidad
Laptop Lenovo (Estación de desarrollo)	Procesador Intel Core I5-11300H (11 Gen) @ 3.10 GHz, 16 GB RAM DDR4 (3200 MHz), GPU NVIDIA GeForce GTX 1650 (4 GB), Almacenamiento dual SSD NVMe (238 GB Samsung + 932 GB Kingston), Sistema operativo 64 bits.	1

Denominación	Características	Cantidad
Dispositivo móvil (Entorno de pruebas)	Smartphone Tecno Pova 6. Utilizando como cliente físico para validar la persistencia local de datos mediante SQLite y el comportamiento asíncrono.	1
Visual Studio Code	Entorno de desarrollo integrado (IDE) principal utilizado para la codificación del backend con TypeScript y Node.js.	1 Licencia (Open source)
MongoDB Compass	Interfaz gráfica oficial empleada para el diseño, validación y gestión de los documentos con esquemas polimórficos a nivel local.	1 Licencia (Open source)

4.3.3 Recursos económicos

El presupuesto del proyecto fue confeccionado teniendo en cuenta los recursos empleados durante la creación del sistema AgroAR, a pesar de que muchas de las herramientas utilizadas fueron gratuitas o de código abierto, se calcularon los costos estimados vinculados al ordenador, teléfono móvil, acceso a internet, transporte y tiempo de desarrollo por el investigador, esto hace posible ofrecer una estimación más precisa de los recursos requeridos para llevar a cabo el proyecto.

Tabla 6: Presupuesto y recursos económicos.

Denominación	Características	Cantidad estimada
Computador portátil	Equipo utilizado para el desarrollo del sistema, programación del backend, aplicación móvil, pruebas locales y redacción del documento.	\$650.00

Denominación	Características	Cantidad estimada
Teléfono inteligente	Dispositivo móvil utilizado para instalar la aplicación, realizar pruebas de escaneo, validar almacenamiento local y comprobar el funcionamiento en campo.	\$250.00
Internet	Internet usado en la investigación bibliográfica, uso de repositorios para versionado motor de base de datos, Cloudinary, Render.	\$60.00
Transporte	Desplazamiento hacia la Granja Experimental para recolección de información, aplicación de encuesta y entrevista.	\$20.00
Horas de desarrollo	Tiempo en análisis, diseño, programación, pruebas, documentación y ajustes del prototipo.	\$300.00
Herramientas de software	Uso de Flutter, Node.js, TypeScript, SQLite, Visual Studio Code, MongoDB Atlas, Cloudinary y Render en planes gratuitos o de código abierto.	\$0.00
Total inversión directa	Valor referencial de los recursos utilizados para el desarrollo del proyecto.	\$1,280.00

4.4 Desarrollo del software mediante metodología Scrum

El desarrollo del sistema informático se llevó a cabo mediante la metodología ágil Scrum, debido a que ofrece la adaptabilidad requerida para afrontar los retos asociados con lo arquitectura Offline-First y la organización flexible de los datos. En los apartados siguientes se describen los artefactos y procedimientos elaborados a lo largo de las distintas etapas del ciclo de desarrollo.

4.4.1 Descripción del producto

4.4.1.1 Propósito del producto

El propósito del sistema es apoyar el registro, análisis y seguimiento de revisiones agrícolas realizadas sobre plantas de plátano en la Granja Experimental de la ULEAM Extensión El Carmen. El sistema permite que el estudiante realice escaneos desde la aplicación móvil, capture imágenes, obtenga resultados técnicos relacionados con plagas, hongos y salud vegetal, y conserve la información aun cuando no exista conexión a internet.

El producto tiene como objetivo combinar una arquitectura móvil y offline con almacenamiento local a través de SQLite y su posterior sincronización con una base de datos centralizada en MongoDB Atlas, asimismo, incorpora almacenamiento de imágenes en la nube mediante Cloudinary, autenticación segura con JWT, creación de informes en PDF y una herramienta de soporte técnico para preguntas.

4.4.1.2 Funcionalidades clave

- Inicio de sesión con credenciales verificadas, de igual manera el registro.
- Mostrar una interfaz diferente dependiendo el rol del usuario ya sea estudiante o administrador.
- Captura de imágenes usando la cámara integrada del celular.
- Elegir el tipo de análisis que desea realizar como, salud vegetal, hongos, plagas o general.
- Revisiones con datos como fecha, explicación técnica, porcentaje de riesgo.
- Memoria local en SQLite para el uso sin conexión a internet.
- Sincronización automática de resultados pendientes hacia la base de datos central cuando hay conexión a internet desde el celular.
- Almacenamiento de imágenes en Cloudinary para permitir su consulta desde diferentes dispositivos.

- Consulta del historial de revisiones realizadas.
- Visualización del detalle técnico de cada revisión.
- Generación de reportes PDF con la información del análisis.
- Integración de un asistente técnico agrícola para consultas sobre el cultivo de plátano.
- Panel administrativo para revisión general de información del sistema.

4.4.1.3 Usuarios objetivo

Tabla 7 : Usuarios del sistema y sus funciones principales.

Usuario	Descripción
Estudiante	Es el usuario en el campo, que utiliza la aplicación para iniciar sesión, escanear plantas, tomar fotos, registrar observaciones, visualizar el historial y reportes PDF. Puede usar sin conexión a internet la aplicación, ya que se guardan localmente los resultados en SQLite hasta que puedan sincronizarse con la base de datos central.
Administrador	Usuario que supervisa la información existente dentro de la aplicación. Puede acceder a un panel diferenciado para revisar métricas generales, consultar datos centralizados y verificar la información sincronizada desde los dispositivos móviles. Su función principal es apoyar el control técnico y académico de los registros generados durante las prácticas agrícolas.

4.4.1.4 Criterios de éxito del producto

- El usuario puede registrarse e iniciar sesión correctamente desde la aplicación móvil.
- El sistema diferencia el acceso entre estudiante y administrador.
- La aplicación toma una imagen y muestra como resultado una revisión técnica de la imagen.

- Los resultados de la revisión se guardan en SQLite de manera correcta cuando el celular no tiene conexión.
- Los resultados pendientes se sincronizan con la base de datos central MongoDB Atlas al detectar conexión a internet.
- Las imágenes capturadas se guardan en Cloudinary y quedan vinculadas a la cuenta que realizó el escaneo.
- El historial permite ver detalles del registro de todas las revisiones guardadas.
- El reporte PDF contiene la información principal del análisis realizado.
- La aplicación mantiene un funcionamiento estable en el dispositivo móvil usado para pruebas.
- La estructura de MongoDB permite almacenar diferentes tipos de análisis mediante un modelo flexible y polimórfico.

4.4.2 Requerimientos del sistema

4.4.2.1 Historias de usuario

4.4.2.1.1 Historia de usuario: Autenticación y acceso seguro

Tabla 8 : Especificación de Historia de usuario HU-01.

Elemento	Descripción
Código	HU-01
Prioridad	Alta
Descripción	Como usuario de la plataforma, deseo autenticarme mediante mis credenciales para ingresar a las funciones y módulos habilitados de acuerdo con el rol que tengo asignado.
Criterios de aceptación	1. El sistema debe comprobar que el nombre de usuario y contraseña registrados coincidan con la información almacenada en la base de datos.

Elemento	Descripción
	2. Después de validar correctamente las credenciales, el sistema debe crear un token JWT seguro para mantener activa la sesión del usuario. 3. Una vez iniciada la sesión, el usuario debe ser dirigido a una interfaz específica según corresponda al rol de estudiante o docente.

4.4.2.1.2 Historia de usuario: Escaneo y análisis de planta

Tabla 9 : Especificación de historia de usuario HU-02.

Elemento	Descripción
Código	HU-02
Prioridad	Alta
Descripción	Como estudiante, deseo escanear una planta de plátano utilizando la cámara del dispositivo para obtener un análisis relacionado con plagas, hongos, salud vegetal o una revisión general.
Criterios de aceptación	La aplicación debe permitir capturar una imagen. El sistema debe permitir seleccionar el tipo de análisis. El resultado debe mostrar el nivel de riesgo, resumen técnico y recomendaciones según la información obtenida. La información recopilada debe organizarse temporalmente en memoria mediante una estructura basada en el formato JSON

4.4.2.1.3 Historia de usuario: Almacenamiento local sin conexión

Tabla 10 : Especificación de historia de usuario HU-03.

Elemento	Descripción
Código	HU-03
Prioridad	Crítica

Elemento	Descripción
Descripción	Como estudiante que realiza actividades de campo en zonas sin acceso a internet, deseo que la aplicación almacene automáticamente la información en la base de datos local SQLite, con la finalidad de evitar la pérdida de los datos recopilados.
Criterios de aceptación	1. Cuando no existe conexión a internet, la información registrada debe guardarse de manera permanente en la base de datos local SQLite. 2. El proceso de almacenamiento en el dispositivo debe completarse en un tiempo máximo de cuatro segundos. 3. Cada registro guardado localmente debe identificarse con el estado <<Pendiente de sincronización>> hasta que pueda enviarse a la base de datos central.

4.4.2.1.4 Historia de usuario: Sincronización con MongoDB

Tabla 11 : Especificación de historia de usuario HU-04.

Elemento	Descripción
Código	HU-04
Prioridad	Crítica
Descripción	Como usuario, deseo que las revisiones guardadas localmente se sincronicen automáticamente con MongoDB cuando el dispositivo recupere conexión a internet.
Criterios de aceptación	La aplicación debe detectar el estado de conexión. Los registros pendientes deben enviarse al backend en formato JSON. El backend debe validar el lote recibido. Los registros sincronizados deben quedar marcados como sincronizados en SQLite.

4.4.2.1.5 Historia de usuario: Historial y reportes PDF

Tabla 12 : Especificación de historia de usuario HU-05.

Elemento	Descripción
Código	HU-05
Prioridad	Alta
Descripción	Como usuario, deseo consultar el historial de revisiones y generar reportes PDF para disponer de evidencia técnica de los análisis realizados.
Criterios de aceptación	El historial debe mostrar las revisiones guardadas. El usuario debe poder visualizar el detalle de cada revisión. El sistema debe generar un PDF con la información técnica del análisis.

4.4.2.2 Requerimientos funcionales

Tabla 13: Requerimientos funcionales del sistema AgroAR Plátano.

Código	Requerimiento funcional
RF-01	El sistema debe permitir el registro e inicio de sesión de usuarios mediante correo y contraseña.
RF-02	El sistema debe verificar el rol del usuario para mostrar la interfaz que corresponde.
RF-03	La aplicación debe registrar imágenes de una planta de plátano mediante la cámara integrada del dispositivo.
RF-04	La aplicación debe otorgar el tipo de análisis que se desea realizar como plagas, hongos, salud vegetal o análisis general.
RF-05	El sistema debe crear una revisión con datos como, fecha, imagen, resumen técnico, porcentaje de riesgo.

Código	Requerimiento funcional
RF-06	La aplicación debe almacenar los registros en SQLite cuando el dispositivo no tenga conexión a internet.
RF-07	El sistema mantiene un estado de sincronización para cada registro que esta local.
RF-08	La aplicación automáticamente debe sincronizar los registros pendientes con el backend cuando detecte conexión a internet.
RF-09	El backend debe verificar, registrar y aceptar los registros sincronizados con la base de datos central.
RF-10	El sistema debe guardar las imágenes y guardar la URL correspondiente en la base de datos central MongoDB Atlas.
RF-11	La aplicación debe permitir consultar el historial de revisiones realizadas.
RF-12	El sistema debe permitir visualizar el detalle técnico de cada revisión registrada.
RF-13	La aplicación debe generar un reporte PDF con la información del análisis realizado.
RF-14	El sistema debe permitir consultar recomendaciones mediante el asistente técnico agrícola integrado.
RF-15	El administrador debe poder acceder a un panel diferenciado para revisar información general del sistema.

4.4.3 Diseño del sistema

4.4.3.1 Casos de uso

4.4.3.1.1 Caso de uso: Registro de observación fitotécnica

Este caso de uso representa la captura inicial de información biométrica en campo mediante la aplicación móvil operando bajo la arquitectura Offline-First.

Ilustración 2: Diagrama de caso de uso: Registro de observación fitotécnica.

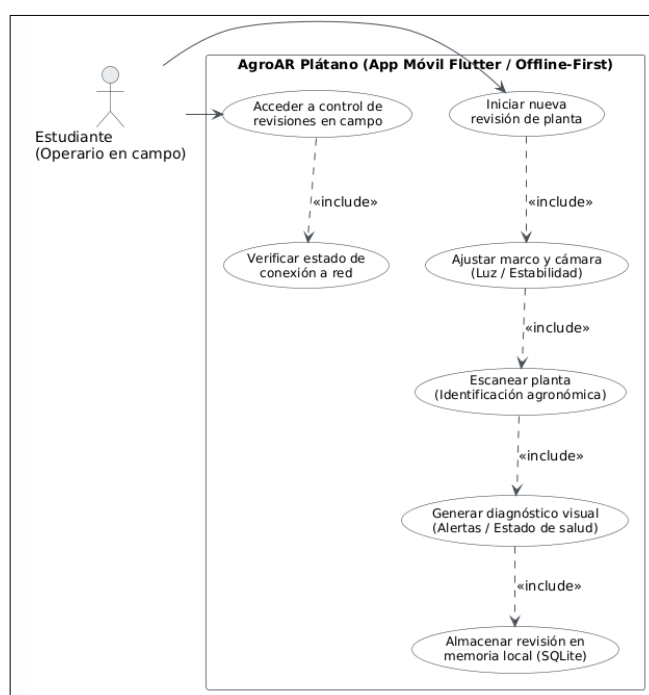


Tabla 14 : Especificación de caso de uso: Registro de observación fitotécnica.

Elemento	Descripción
Actor principal	Estudiante (Operario de campo).
Precondiciones	El usuario debe de instalar la aplicación en su celular y estar frente a la planta que va a escanear.
Flujo principal	1. El usuario está en la pantalla principal y presiona el botón “Iniciar nuevo análisis”. 2. La aplicación abre la pantalla de escaneo donde procede a realizar el escaneo de la planta. 3. El usuario ajusta el zoom de la cámara sobre el plátano y presiona el botón que tiene de ícono la cámara. 4. Los módulos de IA procesan la imagen y emiten un resultado con alertas según sea el caso. 5. EL usuario confirma la revisión para su persistencia en el sistema.

Elemento	Descripción
Flujo alternativo	El sistema ejecuta en segundo plano el proceso de “Verificando conexión”. Si no se detecta red, el diagnóstico queda guardado automáticamente en el contador de revisiones locales de SQLite a la espera de sincronización.
Postcondiciones	La nueva revisión se incrementa en el contador de “Guardadas” y queda lista para consultar en el historial o exportarse a PDF.

4.4.3.1.2 Caso de uso: Sincronización asíncrona de datos

Este caso de uso ilustra la lógica de fondo donde el sistema detecta conectividad y autoriza la transferencia en bloque de los registros hacia el servidor central.

Ilustración 3: Diagrama de caso de uso: Sincronización asíncrona de datos.

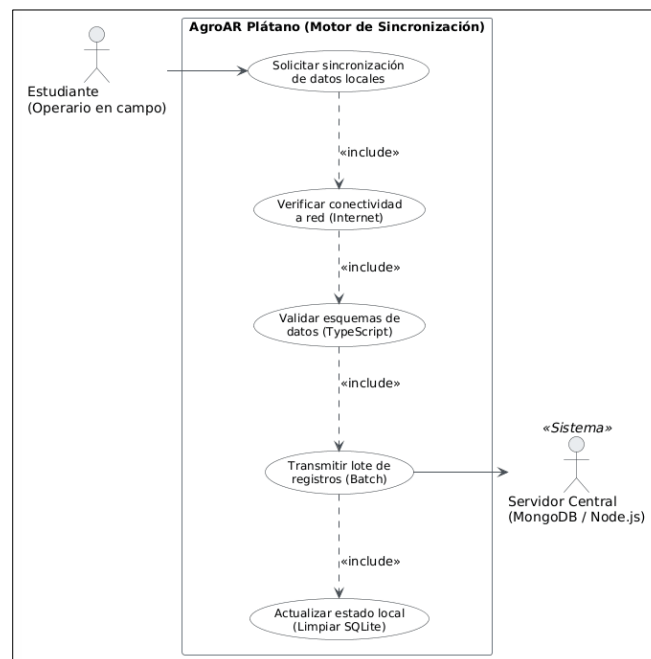


Tabla 15 : Especificación de caso de uso: Sincronización asíncrona de datos.

Elemento	Descripción
Actor principal	Sistema (Ejecución en segundo plano) / Estudiante

Elemento	Descripción
Precondiciones	El dispositivo móvil debe establecer conexión a internet y contener registros locales con el estado “Pendiente”.
Flujo principal	1. El algoritmo detecta el restablecimiento de la red. 2. Se empaquetan los registros pendientes en formato JSON. 3. El sistema móvil dispara la petición POST hacia el backend. 4. El controlador en el servidor valida el token y procesa los esquemas polimórficos. 5. La información se consolida en la base de datos MongoDB.
Postcondiciones	Los registros se marcan como “Sincronizado” y quedan centralizados en la nube.

4.4.3.1.3 Caso de uso: Generación de reporte de evolución

Este escenario describe la extracción de la información ya procesada, permitiendo al docente obtener un entregable descargable con fines analíticos.

Ilustración 4: Diagrama de caso de uso: Generación de reporte de evolución.

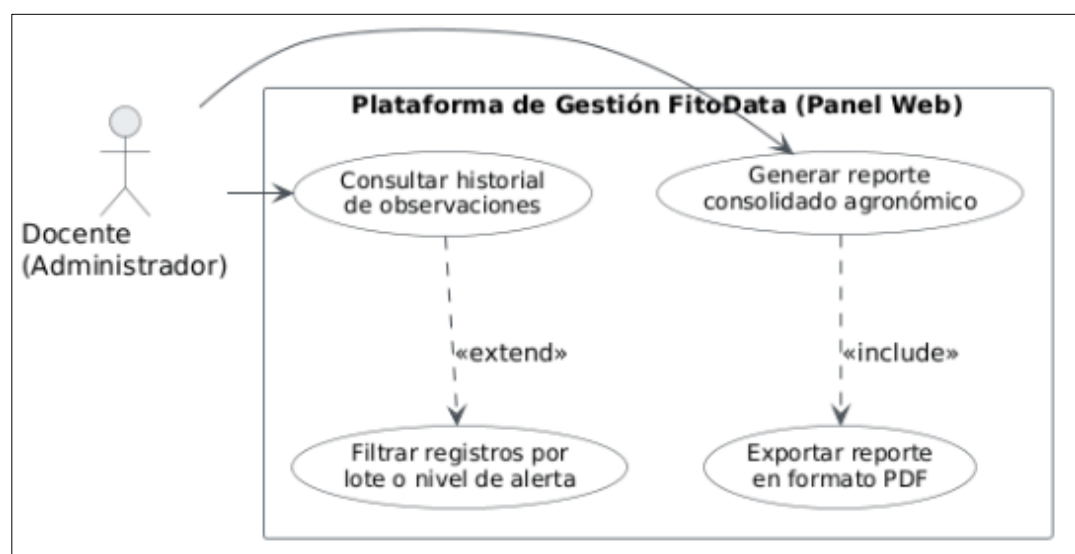


Tabla 16 : Especificación de caso de uso: Generación de reporte de evolución.

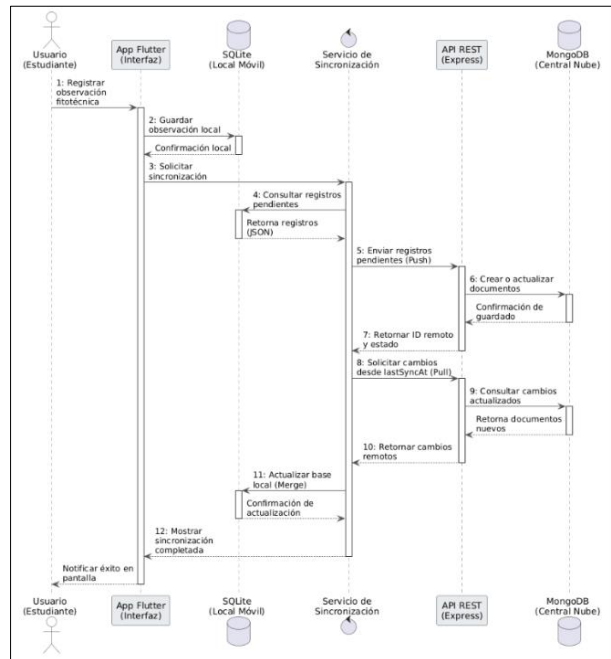
Elemento	Descripción
Actor principal	Administrador (Docente)
Precondiciones	El usuario debe tener los privilegios administrativos y el servidor debe contener los datos sincronizados.
Flujo principal	1. El administrador ingresa al módulo de reportes. 2. Selecciona los filtros deseados. 3. Solicita la exportación de la información. 4. El backend compila los datos requeridos y genera un archivo PDF descargable.
Postcondiciones	El usuario obtiene un documento de salida con la tabulación de la información para la toma de decisiones.

4.4.3.2 Diagramas de secuencia

4.4.3.2.1 Diagrama de secuencia: Registro de observación fitotécnica

El diagrama representa el proceso de comunicación que se desarrolla completamente de forma local y sin conexión a internet, durante este flujo, el operario registra la información en campo y la aplicación interactúa exclusivamente con el mecanismo de almacenamiento integrado en el dispositivo móvil.

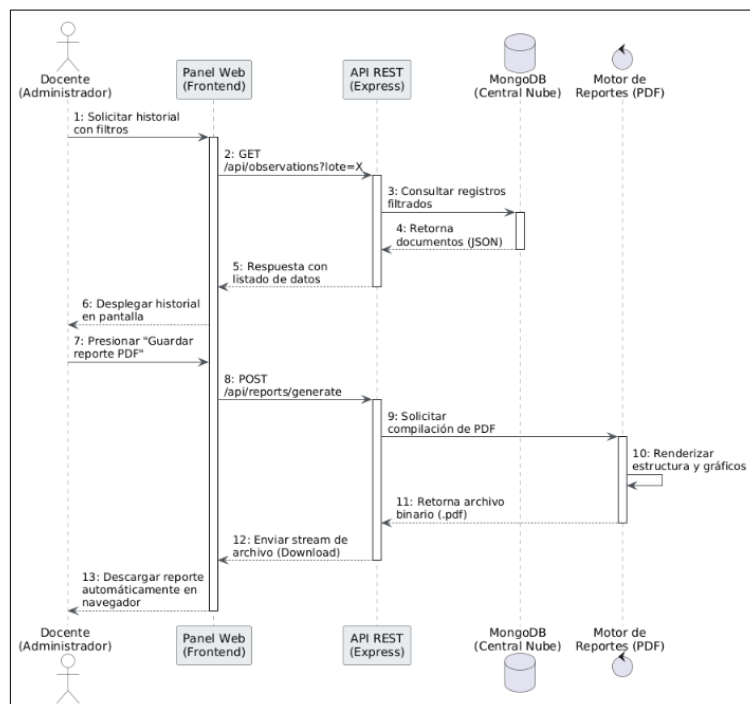
Ilustración 5: Diagrama de secuencia: Registro de observación fitotécnica.



4.4.3.2.2 Diagrama de secuencia: Sincronización de datos asíncrona

El diagrama ilustra el flujo de automatización y comunicación con el servidor, mediante el cual los registros almacenados localmente son verificados y posteriormente enviados a la nube, empleando una estructura de datos basada en esquemas polimórficos.

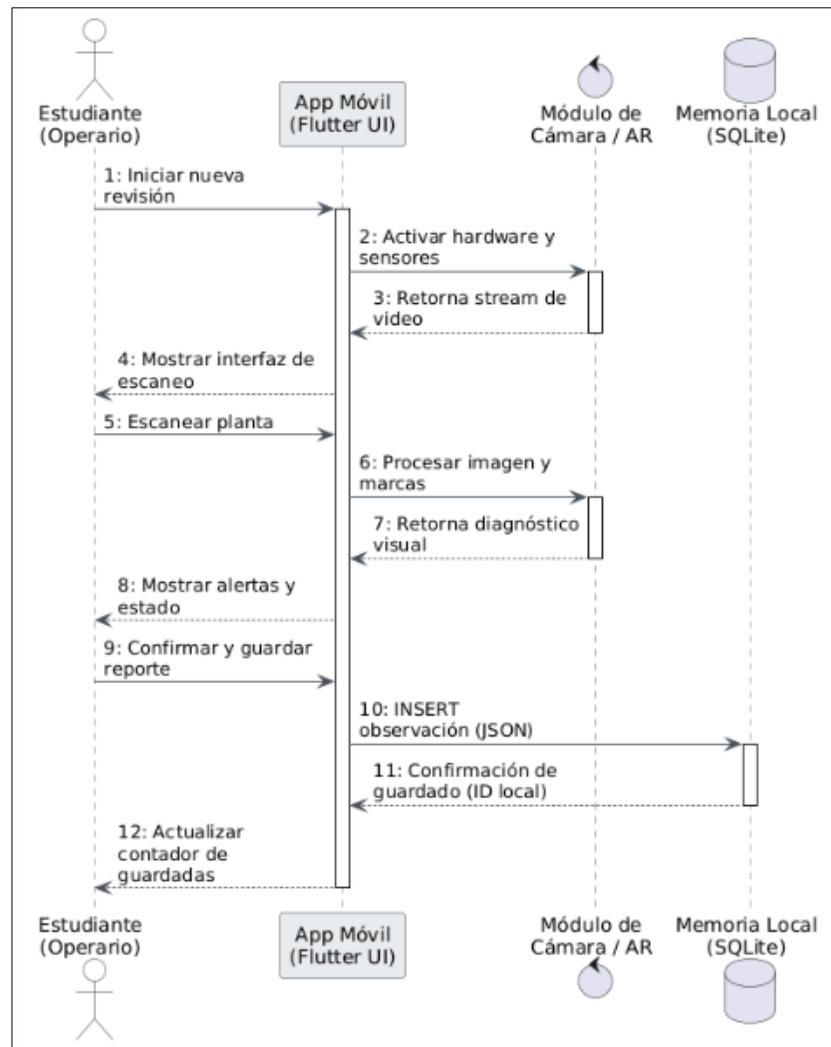
Ilustración 6: Diagrama de secuencia: Sincronización de datos asíncrona.



4.4.3.2.3 Diagrama de secuencia: Generación de reporte de evolución

El diagrama representa el proceso mediante el cual el administrador consulta y obtiene la información previamente consolidada, detallando la secuencia de la solicitud del producto final.

Ilustración 7: Diagrama de secuencia: Generación de reporte de evolución.

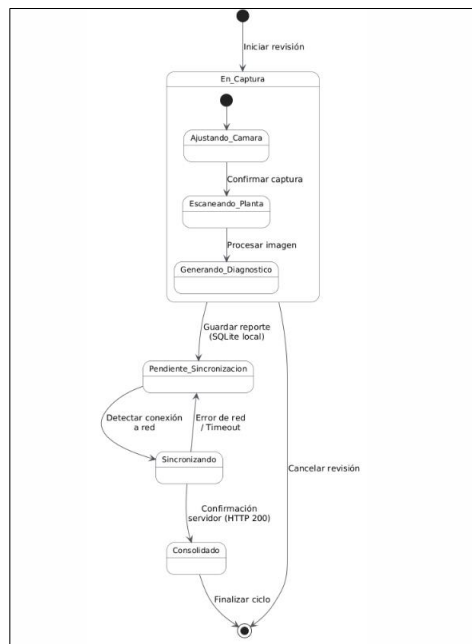


4.4.3.3 Diagramas de estado

4.4.3.3.1 Diagrama de estado: Entidad revisión

Este diagrama representa los estados transicionales de una revisión agronómica desde su creación en el dispositivo móvil en campo hasta su consolidación en el servidor central.

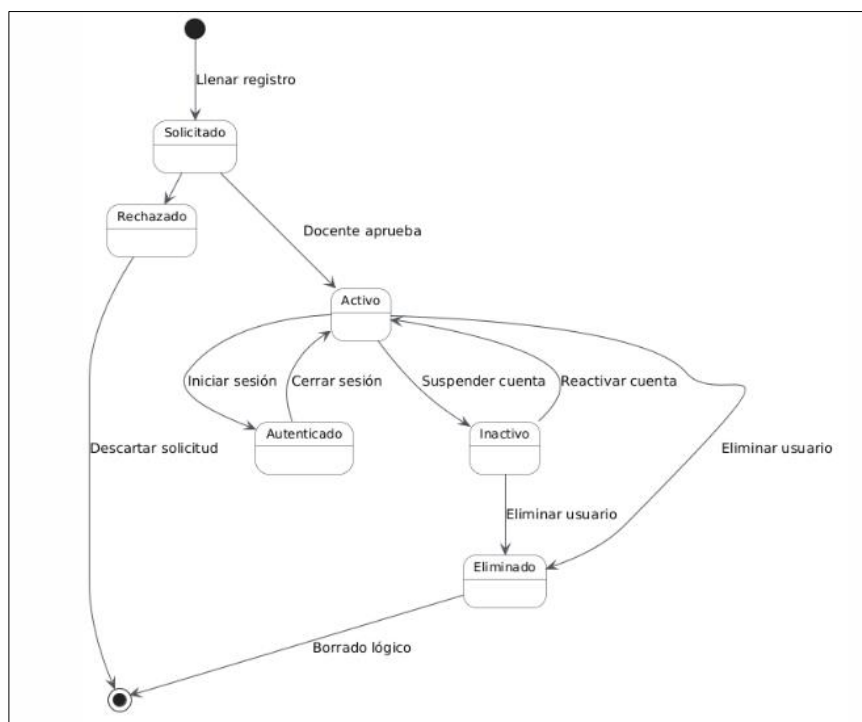
Ilustración 8: Diagrama de estado: Entidad revisión.



4.4.3.3.2 Diagrama de estado: Entidad usuario

Este diagrama detalla el ciclo de vida de la cuenta de un operario (estudiante) dentro de la plataforma, ilustrando los niveles de acceso y los controles administrativos que ejerce el docente.

Ilustración 9: Diagrama de estado: Entidad usuario.

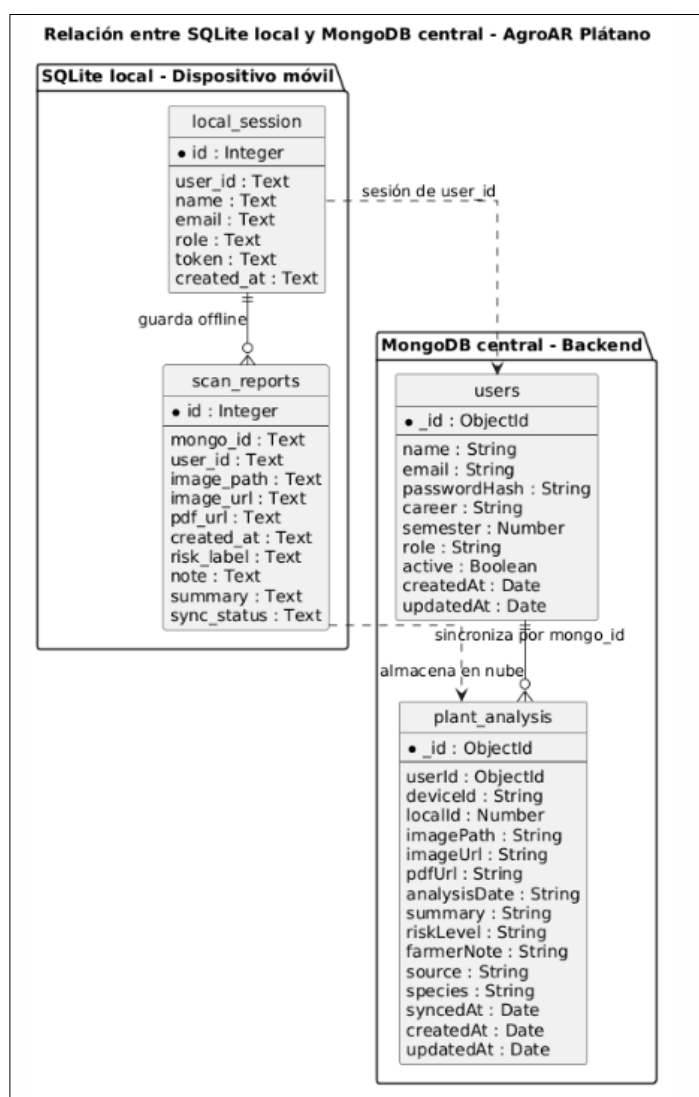


4.4.3.4 Diagrama de Base de Datos

La aplicación implementa una arquitectura de persistencia híbrida diseñada para soportar el funcionamiento Offline-First. Para lograr esto, el modelo de datos se divide en dos capas físicas y lógicas distintas, una base de datos relacional embebida en el dispositivo móvil y una base de datos centralizada orientada a documentos en la nube.

Debido a la naturaleza heterogénea de ambos motores SQL y NoSQL, se presenta un diagrama lógico unificado. En este esquema, las líneas continuas representan relaciones físicas estrictas dentro de cada entorno, mientras que las líneas punteadas ilustran el mapeo lógico de sincronización y transferencia de datos entre el dispositivo móvil y la nube.

Ilustración 10: Diagrama lógico de persistencia y sincronización de datos.



4.4.3.5 Diccionario de datos

A continuación, se detalla la estructura de los principales campos que conforman el modelo de datos del sistema, tanto en la colección documental de MongoDB Atlas como en la base de datos SQLite, evidenciando la aplicación práctica del esquema polimórfico mediante el campo discriminador que termina el tipo de análisis registrado.

MongoDB Atlas

Tabla 17: Estructura de la colección de análisis en MongoDB Atlas.

Campo	Tipo	Descripción
_id	ObjectId	Identificador único del documento generado automáticamente por MongoDB.
userId	ObjectId	Referencia al usuario que generó el registro del análisis.
analysisType	String	Campo discriminador del esquema polimórfico. Indica el tipo de análisis registrado (en este documento: "complete_analysis").
deviceId	String	Identificador del dispositivo desde el cual se generó el escaneo.
localId	Int32	Identificador numérico incremental asignado en SQLite antes de sincronizar.
localUuid	String	Identificador único generado en el dispositivo para evitar duplicados al sincronizar.
imagePath	String	Ruta local de la imagen capturada en el dispositivo móvil.
imageUrl	String	Enlace de la imagen almacenada en Cloudinary.

Campo	Tipo	Descripción
analysisDate	String	Fecha y hora en que se realizó el análisis en el dispositivo.
Summary	String	Resumen textual generado automáticamente con el resultado del análisis.
riskLevel	String	Nivel de riesgo detectado en la planta (ej. "medio").
farmerNote	String	Observación ingresada por el usuario durante el escaneo.
Source	String	Origen del registro (ej. "flutter_sqlite", indica que vino de sincronización local).
syncedAt	Date	Fecha y hora en que el registro fue sincronizado con MongoDB Atlas.
detectedConditions	Array	Lista de plagas o condiciones detectadas por el módulo de IA.
fungusResult	Object	Subdocumento con los resultados del análisis de hongos/enfermedades.
healthResult	Object	Subdocumento con las métricas del análisis de salud vegetal.
createdAt	Date	Marcas de tiempo generadas automáticamente por Mongoose.
updatedAt	Date	Marcas de tiempo generadas automáticamente por Mongoose.

SQLite

Tabla 18: Estructura de la tabla scan_reports en SQLite.

Campo	Tipo	Descripción
id	Integer (PK, AUTOINCREMENT)	Identificador local auto incremental del registro.
local_uuid	Text (NOT NULL, UNIQUE)	Identificador único generado en el dispositivo; evita duplicados al sincronizar con MongoDB.
user_id	Text	Referencia al usuario autenticado que generó el registro.
analysis_type	Text (DEFAULT 'general_scan')	Campo discriminador del esquema polimórfico; determina el tipo de análisis.
analysis_details_json	Text (DEFAULT '{}')	Estructura JSON en texto plano con los resultados detallados del análisis.
image_path	Text (NOT NULL)	Ruta local de la imagen capturada, antes de subirse a Cloudinary.
created_at	Text (NOT NULL)	Fecha y hora de creación del registro en el dispositivo.
risk_label	Text (NOT NULL)	Nivel de riesgo detectado.
note	Text (DEFAULT '')	Observación ingresada por el usuario.
summary	Text (DEFAULT '')	Resumen textual generado con el resultado del análisis.

Campo	Tipo	Descripción
sync_status	Text (DEFAULT 'pending')	Estado de sincronización: "pending" o "synced".
mongo_id	Text	Referencia al _id asignado por MongoDB una vez sincronizado.
image_url	Text	Enlace de Cloudinary, se llena tras subir la imagen.
detected_conditions	Text (DEFAULT '{}')	Lista de condiciones/plagas detectadas, en formato JSON de texto.

En las siguientes ilustraciones se aprecia la diferencia de un registro de salud vegetal y plaga.

Ilustración 11: Visualización de registro de salud vegetal con el campo discriminador en MongoDB Atlas.

```

_id: ObjectId('6a62e8dac1828914f7edf2b4')
userId: ObjectId('6a62e88fc1828914f7edf2a4')
analysisType: "health_analysis"
deviceId: "agroar-integrado-tecno"
localId: 2
localUuid: "cf056fd7-037e-4360-8438-965854f0a960"
imagePath: "/data/user/0/com.example.uleam_banana_ar_hud/cache/CAP5888444477499658..."
imageUrl: "https://res.cloudinary.com/ajvhuvui/image/upload/v1784867033/agroar/pl..."
analysisDate: "2026-07-23T23:23:51.351349"
summary: "Escaneo de salud vegetal con resultado confirmado. Resultado: Dar segu..."
riskLevel: "medio"
farmerNote: "Escaneo realizado desde el visor AR."
source: "flutter_sqlite"
syncedAt: 2026-07-24T04:23:54.307+00:00
▼ detectedConditions: Array (1)
  ▼ 0: Object
    type: "health"
    name: "Salud comprometida"
    riskLevel: "medio"
    confidence: 0.7898104786872864
    _id: ObjectId('6a62e8dac1828914f7edf2b5')
    healthStatus: "Salud comprometida"
    vigorScore: 4.2
  ► recommendations: Array (1)
    __v: 0
  createdAt: 2026-07-24T04:23:54.434+00:00
  updatedAt: 2026-07-24T04:23:54.434+00:00

```

Ilustración 12: Visualización de registro de plaga con el campo discriminador en MongoDB Atlas.

```
_id: ObjectId('6a637377fde5e2d4e8bf1aea')
userId: ObjectId('6a5f7a4b848e2a8447b9fdf1')
analysisType: "plague_analysis"
deviceId: "agroar-integrado-tecno"
localId: 2
localUuid: "cd4234b6-ed1c-4956-a23b-2e97c8efc0f7"
imagePath: "/data/user/0/com.example.uleam_banana_ar_hud/cache/CAP3946032721425655..."
imageUrl: "https://res.cloudinary.com/ajvhuvui/image/upload/v1784902518/agroar/pl..."
analysisDate: "2026-07-24T09:15:16.245245"
summary: "Escaneo de plagas con hallazgo confirmado. Resultado: Revisar pronto."
riskLevel: "alto"
farmerNote: "Escaneo realizado desde el visor AR."
source: "flutter_sqlite"
syncedAt: 2026-07-24T14:15:19.163+00:00
▼ detectedConditions: Array (1)
  ▼ 0: Object
    type: "plague"
    name: "Picudo negro"
    riskLevel: "alto"
    confidence: 0.3593747019767761
    _id: ObjectId('6a637377fde5e2d4e8bf1aeb')
    plagueName: "Picudo negro"
    confidence: 0.3593747019767761
    affectedArea: "Hoja, tallo o pseudotallo visible"
    __v: 0
    createdAt: 2026-07-24T14:15:19.292+00:00
    updatedAt: 2026-07-24T14:15:19.292+00:00
```

4.4.4 Arquitectura del sistema

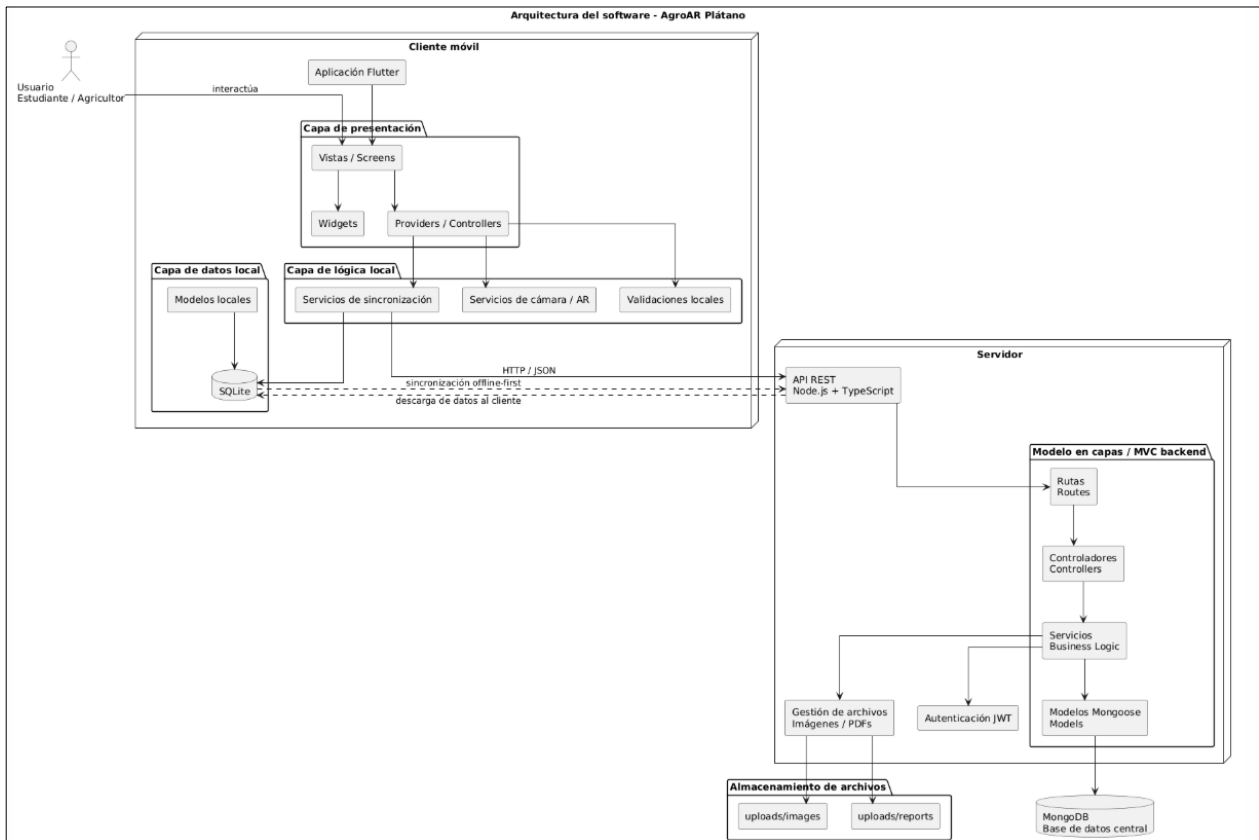
4.4.4.1 Arquitectura general

El sistema está diseñado bajo el paradigma arquitectónico Cliente-Servidor, distribuyendo la carga operativa entre los dispositivos móviles de los usuarios en campo y un servidor centralizado en la nube. Para garantizar el aislamiento de responsabilidades, la mantenibilidad y la escalabilidad del código, el sistema implementa un Modelo en Capas fusionado con los principios del patrón MVC Modelo-Vista-Controlador.

1. Cliente Móvil interfaz multicapa desarrollada con Flutter, funciona como el cliente principal. Internamente aplica un modelo organizado en tres capas.

- Capa de presentación: se encarga de las pantallas que ve el usuario y de mantener la interfaz reactiva a través de Widgets y Providers.
 - Capa lógica local: Reúne en un solo punto los servicios que dependen del hardware, junto con las validaciones internas y el motor encargado de sincronizar datos.
 - Capas de datos local: Se ocupa de todo el acceso a SQLite, la base de datos embebida en el celular, y con ello asegura que la información se conserve incluso sin conexión, siguiendo el enfoque Offline-First.
2. Servidor backend construido sobre Node.js y TypeScript, cumple la función de servidor central del sistema. Su flujo de trabajo sigue en patrón MVC adaptando a servicios API REST, dividido en cuatro componentes.
- Routes: reciben y canalizan las peticiones HTTP entrantes.
 - Controllers: Coordinan las respuestas hacia el cliente y verifican que los datos recibidos sean correctos.
 - Services: Centralizan la lógica de negocio más compleja, junto con la generación de tokens JWT y el manejo de archivos.
 - Models: Corresponden a los esquemas de Mongoose que se comunican con la capa de persistencia.
3. Almacenamiento y Base de Datos Central: Los componentes finales del sistema, responsables de guardar los archivos binarios como imágenes, reportes PDF y consolidar la información fitotécnica en la base de datos NoSQL MongoDB.

Ilustración 13: Diagrama arquitectónico del sistema: Cliente/Servidor, Modelo en Capas y MVC.



4.4.4.2 Tecnologías utilizadas

Para el desarrollo, implementación y mantenimiento del sistema se adoptó un conjunto de tecnologías actuales, enfocadas en la creación de soluciones multiplataforma y arquitecturas con capacidad de crecimiento. La selección tecnológica se fundamentó en la necesidad de garantizar el funcionamiento de la aplicación bajo un modelo Offline-First durante las actividades de campo, sin descuidar la transferencia segura y eficiente de la información hacia los servicios alojados en la nube.

En los apartados siguientes se describen los lenguajes de programación, marcos de desarrollo, sistemas gestores de bases de datos y entornos de desarrollo integrados empleados durante las diferentes etapas del ciclo de vida del software.

Tabla 19: Herramientas y tecnologías utilizadas en el desarrollo del proyecto.

Categoría técnica	Tecnología/Herramienta	Uso dentro del proyecto
Frontend	Flutter	Construcción de la aplicación móvil multiplataforma.
Lenguaje	Dart	Programación de pantallas, controladores, servicios y modelos de la app.
Backend	Node.js	Entorno de ejecución del servidor.
Lenguaje	TypeScript	Aplicación de tipado estricto y organización de la lógica del servidor.
API	Express	Definición de rutas REST para autenticación, la sincronización, el manejo de imágenes y las consultas.
Base de datos local	SQLite	Almacenamiento con conexión de las revisiones directamente en el dispositivo móvil.
Base de datos central	MongoDB Atlas	Persistencia NoSQL de usuarios, revisiones y estructuras polimórficas.
Modelado de datos	Mongoose	Definición de esquemas y modelos para MongoDB.
Seguridad	JWT	Protección de rutas y sostenimiento de la sesión autenticada.
Almacenamiento de imágenes	Cloudinary	Carga y consulta de imágenes capturadas durante los escaneos.

Categoría técnica	Tecnología/Herramienta	Uso dentro del proyecto
Infraestructura backend	Render	Publicación del servidor para su acceso desde cualquier red.
Chatbot	Servicio Agrobot	Atención de consultas técnicas relacionadas con el cultivo de plátano.
Reportes	Librería PDF en Flutter	Elaboración de reportes técnicos en formato PDF.
Control de versiones	Git y GitHub	Administración del código fuente, las ramas, los commits y los Pull Requests.
Pruebas de API	Postman	Validación manual de endpoints del backend.
Gestión de base de datos	MongoDB Compass / Atlas UI	Consulta y verificación de documentos almacenados.

4.4.5 Roles Scrum

Para la planificación, gestión y desarrollo del sistema AgroAR, se adecuó la metodología ágil Scrum a las características propias de un proyecto académico de investigación y desarrollo tecnológico. Debido a que la mayor parte de la ejecución está a cargo de un solo investigador, los roles definidos por Scrum fueron distribuidos de manera estratégica, con el propósito de asegurar tanto la validación funcional desde el área agronómica como la correcta implementación técnica desde la ingeniería de software.

A continuación, se describen los tres roles principales de Scrum, las personas responsables de asumirlos y las funciones específicas que cumplen dentro del desarrollo del proyecto.

Tabla 20: Descripción y asignación de roles Scrum en el proyecto.

Rol Scrum	Participantes asignados	Función dentro del proyecto
Product Owner	Tutor académico y responsables de la Granja Experimental ULEAM Extensión El Carmen.	Representan las necesidades del usuario final y orientan las prioridades del sistema. Su función consiste en comprobar que el prototipo cumpla con los requerimientos agrícolas, más que nada en lo relacionado con el registro, el análisis y el seguimiento de las revisiones realizadas sobre el plátano.
Scrum Master	Jose Manuel Zamora Llor.	Le corresponde ordenar el trabajo del proyecto y distribuir las tareas a lo largo de los distintos sprints, además de detectar a tiempo cualquier bloqueo técnico que pudiera frenar el desarrollo. Su rol incluye comprobar que las actividades planificadas se ejecuten conforme a lo establecido por la metodología Scrum.
Scrum Team	Jose Manuel Zamora Llor y equipo de integración del proyecto.	Desarrollan e integran las funcionalidades del sistema. En este módulo se incluye la aplicación móvil en Flutter, persistencia local en SQLite, backend en Node.js/TypeScript, sincronización con MongoDB Atlas, almacenamiento de imágenes en Cloudinary, generación de

Rol Scrum	Participantes asignados	Función dentro del proyecto
		reportes PDF y conexión con otros módulos del proyecto.

4.4.6 Definición de Hecho (DoD)

4.4.6.1 Criterios generales

En el desarrollo del sistema AgroAR, los criterios generales comprenden el conjunto de condiciones técnicas y funcionales que todo módulo, historia de usuario o incremento del software debe cumplir para ser considerado finalizado, de acuerdo con la Definition of Done, y estar preparado para su implementación en el entorno operativo de la finca experimental. Estos lineamientos funcionan como un mecanismo transversal de control de calidad, revisado de forma constante durante las etapas de construcción, integración y validación del proyecto.

A continuación, se presentan los requisitos generales de calidad que orientan el funcionamiento del software.

- **Compilación y ejecución correcta del código fuente:** El código fuente correspondiente al cliente móvil desarrollado en Flutter/Dart y al servidor central implementado con Node.js/TypeScript debe compilarse, construirse y ejecutarse adecuadamente, sin presentar excepciones no controladas, fallos del sistema ni advertencias críticas durante el proceso de compilación. Todo fragmento de código incorporado al proyecto debe ser revisado y depurado previamente en el entorno de desarrollo integrado.
- **Cumplimiento de las historias de usuario:** Cada funcionalidad implementada debe responder completamente a las especificaciones, reglas de negocio y criterios de aceptación establecidos en la historia de usuario correspondiente. Según esta norma, ninguna tarea puede considerarse completa hasta que satisfaga realmente la necesidad

agrícola o administrativa que dio origen a los requisitos del usuario final, ya sea el estudiante que utiliza el sistema en el campo o el profesor que lo gestiona.

- **Integración modular y coherencia arquitectónica:** Cada nueva función que se añada debe integrarse perfectamente con los componentes del sistema existentes, incluyendo la cámara y el módulo de realidad aumentada, el servicio de autenticación basado en JWT, el mecanismo de almacenamiento local SQLite y la base de datos central MongoDB. Bajo ninguna circunstancia dicha adición debe degradar, interrumpir o deshabilitar el funcionamiento de los módulos que se utilizan actualmente.
- **Consistencia visual e integridad de la interfaz:** La aplicación móvil y el panel de administración necesitan poseer una interfaz sencilla, intuitiva, armoniosa y bien estructurada, esto significa, adherirse a las pautas corporativas y a los principios de ergonomía, lo que incluye una navegación ajustada a diversos tamaños de pantalla, compatibilidad con modos de visualización claro y oscuro, y la completa eliminación de sobrecarga de contenido, botones inadecuados o imágenes deformadas.
- **Estabilidad y tolerancia a fallos bajo el enfoque Offline-First:** Considerando que el sistema será utilizado en zonas agrícolas con conectividad limitada, los módulos encargados del registro de observaciones fitotécnicas deben asegurar el almacenamiento local de la información en SQLite, incluso cuando no exista acceso a internet. El software se considera apto únicamente si mantiene los datos sin pérdidas y puede sincronizarlos con el servidor una vez que la conexión sea restablecida.
- **Validación, integridad y seguridad de la información:** Toda información ingresada mediante formularios o capturada durante el trabajo de campo debe ser validada tanto en el cliente como en el servidor, utilizando esquemas definidos y mecanismos de tipado, como Mongoose y TypeScript. Esto permite evitar el ingreso de datos

inconsistentes, corruptos o solicitudes no autorizadas que no cuenten con un token de seguridad válido.

Tabla 21: Criterios generales de aceptación y calidad del software.

Criterio general	Descripción técnica del requisito	Método de verificación en el proyecto
Compilación sin errores	El código desarrollado en Flutter/Dart y Node.js/TypeScript debe compilarse y ejecutarse adecuadamente, sin presentar errores críticos, advertencias importantes ni fallos durante la ejecución.	Ejecución de comandos de construcción en la terminal y revisión de los registros generados durante la depuración en Visual Studio Code.
Cumplimiento funcional	Cada funcionalidad debe responder completamente a los requerimientos agronómicos y técnicos establecidos en su respectiva historia de usuario.	Comparación de los entregables desarrollados con el Product Backlog validado por el Product Owner.
Integración modular	El módulo implementado debe conectarse y funcionar correctamente con los demás componentes del sistema, como SQLite, API REST, MongoDB y cámara, sin afectar funcionalidades existentes.	Realización de pruebas de integración entre la aplicación móvil y los endpoints del servidor desarrollado en Express.
Consistencia visual (UI)	Las pantallas del sistema deben mantener un diseño claro, adaptable y ergonómico, evitando errores visuales	Ejecución de pruebas de usabilidad y revisión del renderizado en dispositivos

Criterio general	Descripción técnica del requisito	Método de verificación en el proyecto
	como desbordamiento de texto, elementos desalineados o problemas de visualización.	móviles con distintas resoluciones de pantalla.
Operatividad Offline-First	El sistema debe permitir el registro de datos fitotécnicos en campo aun sin conexión a internet y sincronizarlos automáticamente cuando se detecte disponibilidad de red.	Pruebas de interrupción de conectividad mediante modo avión durante la captura, almacenamiento y posterior sincronizado de los datos.

4.4.6.2 Criterios específicos del proyecto

4.4.6.2.1 Módulo de autenticación, seguridad y gestión de roles

- **Ingreso permitido únicamente para usuarios autorizados:** El acceso a la aplicación móvil o al panel web debe habilitarse solo para cuentas cuyo estado se encuentre marcado como **Activo** dentro de la base de datos. En caso de que el usuario tenga un estado **Solicitado**, **inactivo** o **Eliminado**, el sistema deberá impedir el inicio de sesión y mostrar una notificación en pantalla indicando la restricción correspondiente.
- **Control de permisos según el rol del usuario:** El sistema debe limitar el acceso a las vistas y funcionalidad de acuerdo con el rol asignado. Los usuarios con el rol **Estudiante** podrán utilizar únicamente las funciones relacionadas con el registro de revisiones en campo y la consulta de su historial personal. Por otro lado, los usuarios con rol **Administrador** tendrán acceso al panel administrativo, desde donde podrán gestionar usuarios, supervisar registros y generar reportes en formato PDF.

- **Vigencia y control de tokens de sesión:** El token JWT generado durante el proceso de autenticación debe contar con un tiempo de expiración previamente definido. Cuando la sesión caduque, el sistema deberá solicitar nuevamente las credenciales del usuario antes de permitir operaciones que requieran comunicación real con la API REST.

4.4.6.2.2 Módulo de registro y captura fitotécnica en campo

- **Asignación automática de autoría y fecha de registro:** Al guardar una revisión, el sistema debe incorporar automáticamente el identificador único del usuario autenticado y la fecha y hora exacta de la captura obtenida desde el dispositivo. Estos datos deben registrarse en segundo plano y no podrán ser modificados manualmente por el operario, ya que forman parte de la trazabilidad del sistema.
- **Optimización de evidencias fotográficas:** Antes de almacenar una imagen en la base de datos local o enviarla al servidor, el módulo de cámara debe redimensionar y comprimir la fotografía capturada. Este proceso permitirá obtener un archivo más liviano y adecuado para la transferencia, evitando superar los límites de envío o consumir excesivamente el almacenamiento del dispositivo móvil.

4.4.6.2.3 Módulo de sincronización y arquitectura offline-first

- **Asignación inicial del estado de sincronización:** Todo registro generado desde la aplicación móvil sin conexión a internet debe almacenarse en SQLite con el atributo **sincronizado** en **false** y con el estado **PENDIENTE_SINCRONIZACION**. Mientras no exista una conexión estable, el sistema debe impedir cualquier intento de envío hacia el servidor.
- **Actualización del estado tras una sincronización exitosa:** Cuando el servicio de sincronización envíe correctamente un reporte y servidor Node.js responda con un código HTTP 200 junto con el identificador generado en MongoDB, la aplicación móvil deberá actualizar el registro en SQLite. Para ello, cambiará el valor de **sincronizado** a

true y su estado **SINCRONIZADO**, evitando que el mismo dato sea enviado nuevamente en futuras sincronizaciones.

- **Manejo seguro de errores durante la transmisión:** Si se produce una falla de conexión mientras se envía un grupo de revisiones pendientes al backend, el sistema no debe borrar, reemplazar ni alterar los datos almacenados localmente en SQLite. En su lugar, el registro deberá conservarse o regresar al estado **PENDIENTE_SINCRONIZACION**, informando al operario que la operación podrá intentarse nuevamente cuando exista conectividad.

4.4.6.2.4 Módulo de historial, consultas y reportes PDF

- **Filtrado por criterios agrícolas específicos:** El módulo de historial del panel web debe permitir la consulta de observaciones mediante filtros combinables, tales como rango de fechas, usuario y estado de salud del cultivo.
- **Generación estandarizada de reportes PDF:** Al utilizar la opción **Generar reporte PDF**, el sistema debe generar un documento que incluya obligatoriamente el membrete institucional de la ULEAM, una tabla resumen con los indicadores consolidados del cultivo y las miniaturas de las evidencias fotográficas asociadas a la planta consultada.

4.4.7 Product Backlog

4.4.7.1 Product Backlog

El Product Backlog constituye el listado principal de funcionalidades, actividades técnicas y requerimientos arquitectónicos necesarios para el desarrollo completo del sistema. Cada elemento incluido proviene de una historia de usuario previamente validada con el usuario final y ha sido distribuido en iteraciones de trabajo organizadas cronológicamente, denominadas Sprints.

Según los lineamientos metodológicos establecidos para el proyecto, cada funcionalidad cuenta con un nivel de prioridad, clasificado como alta, media o baja, y con un

estado de ejecución, que puede ser por hacer, en progreso, completada o cancelada. Al tratarse de la planificación inicial del proyecto, todas las actividades se registran inicialmente en el estado **Por hacer**, el cual será actualizado progresivamente conforme se desarrollen los ciclos de trabajo.

Tabla 22: Product Backlog general: Funcionalidades, prioridades, Sprints y estado inicial.

ID	Módulo	Funcionalidad	Sprint asignado	Prioridad	Estado actual
HU-01	Autenticación.	Implementar registro, inicio de sesión, validación JWT, sesión local y control de acceso por roles.	Sprint 1.	Alta.	En progreso.
HU-02	Escaneo y análisis.	Integrar cámara, captura de imagen, selección del tipo de análisis e interpretación técnica del resultado.	Sprint 1.	Alta.	En progreso.
HU-03	Persistencia local.	Guardar revisiones en SQLite con estado pendiente o sincronizado para	Sprint 2.	Crítica.	Pendiente.

ID	Módulo	Funcionalidad	Sprint asignado	Prioridad	Estado actual
		funcionamiento offline-first.			
HU-04	Sincronización.	Sincronizar registros entre SQLite y MongoDB mediante backend Node.js/TypeScript y API REST.	Sprint 2.	Crítica.	Pendiente.
HU-05	Historial y reportes.	Consultar revisiones, visualizar detalles, almacenar imágenes en la nube y generar reportes PDF.	Sprint 3.	Alta.	Pendiente.

4.4.7.2 Priorización

Para definir la importancia de cada elemento del Product Backlog, clasificados en alta, media o baja, se analizó el grado de impacto que cada tarea tiene sobre la viabilidad técnica y agronómica del sistema.

- **Prioridad alta:** Se asignó a las funcionalidades esenciales relacionada con la arquitectura Offline-First y la captura de información en campo, correspondiente

principalmente a los Sprints 1 y 2. Si componentes como la cámara integrada en Flutter, la base de datos local SQLite o el mecanismo de sincronización con MongoDB presentan fallos, el registro fitotécnico en zonas sin conectividad no podría realizarse adecuadamente.

- **Prioridad media:** Incluye las actividades vinculadas con la seguridad, la auditoría de datos y las interfaces de gestión para el administrador, desarrollados en el Sprint 3. Estas tareas permiten evitar la duplicidad de información, asegurar la integridad de los registros y facilitar su revisión desde un entorno web.
- **Prioridad baja:** Agrupa funcionalidades complementarias de administración y generación documental, planificadas para el Sprint 4. Aunque la exportación de reportes PDF y la gestión de lotes aportan valor institucional para la ULEAM, el sistema puede operar técnicamente en campo sin depender inicialmente de estos módulos.

4.4.8 Sprint 1: Autenticación, Seguridad e integración de cámara

4.4.8.1 Planificación del Sprint

Tabla 23: Planificación general del Sprint 1.

SPRINT 1: Autenticación, Seguridad e integración de cámara			
Duración:	3 semanas (05/01/2026 – 25/01/2026)	Prioridad:	Alta
Objetivo:	Esta etapa se enfoca en finalizar la instalación inicial del servidor backend y la aplicación móvil, además de integrar el módulo de seguridad utilizando tokens JWT, asimismo, incorpora la integración de las bibliotecas nativas de Flutter requeridas para activar la cámara y posibilitar el escaneo de plantas en el campo.		
Historias de usuario:	HU-01: Registro, inicio de sesión generación del token JWT y control de acceso por rol. HU-02: Integración de cámara en Flutter para captura, selección de imagen y análisis de planta		
Tareas de desarrollo:	• El servidor backend se configuró con su configuración básica, utilizando Node.js, TypeScript y Express. • Sincrónicamente, se configuró un entorno de desarrollo de aplicaciones móviles (DAE), desarrollado con Flutter y el lenguaje de programación Dart. • También se configuró un endpoint de autenticación (POST /api/auth/login), responsable de generar tokens JWT para gestionar de forma segura las sesiones de los usuarios.		

SPRINT 1: Autenticación, Seguridad e integración de cámara	
	• Diseño e implementación de la pantalla de inicio de sesión en la aplicación móvil. • Configuración de los permisos requeridos en Android e IOS para acceder al hardware de la cámara. • Ejecución de pruebas unitarias para validar el proceso de autenticación y comprobar la captura visual en un dispositivo físico.
Plan de entrega:	• Preparación de los entornos de desarrollo en Node.js y Flutter. Durante esta actividad se crearon los repositorios del proyecto mediante Git y se dejó configurada la estructura inicial del servidor utilizando Express. También se instalaron las dependencias necesarias para el desarrollo y se organizó la base del proyecto móvil multiplataforma en Flutter. • Tiempo de desarrollo: 12 horas. • Fecha de inicio y finalización: 05/01/2026 - 06/01/2026. • Implementación del sistema de autenticación utilizando JWT. Se llevó a cabo la implementación de la lógica requerida para mejorar el inicio de sesión de los usuarios. Se integró el backend con MongoDB, lo que facilitó la validación de las credenciales de que las cuentas estaban activas. Asimismo, se creó un middleware para verificar los tokens JWT en las rutas aseguradas del sistema. • Tiempo de desarrollo: 24 horas. • Fecha de inicio y finalización: 07/01/2026 - 11/01/2026. • Construcción de la pantalla de inicio de sesión en Flutter. Se diseñó y programó la interfaz de acceso de la aplicación móvil. En el formulario se añadieron validaciones para evitar el envío de campos incompletos o incorrectos. Asimismo, la pantalla fue conectada con la API REST para realizar la autenticación y almacenar de forma segura el token generado durante la sesión. • Tiempo de desarrollo: 16 horas. • Fecha de inicio y finalización: 12/01/2026 - 14/01/2026. • Configuración de permisos y acceso a la cámara del dispositivo. En esta etapa se realizaron las configuraciones necesarias en Android e iOS para solicitar autorización al usuario antes de utilizar la cámara. También se integraron los componentes nativos con Flutter, permitiendo que la aplicación pudiera acceder correctamente al hardware del dispositivo.

SPRINT 1: Autenticación, Seguridad e integración de cámara	
	• Tiempo de desarrollo: 20 horas. • Fecha de inicio y finalización: 15/01/2026 - 19/01/2026. • Implementación de la pantalla de escaneo y del marco de referencia AR. Se desarrolló la vista destinada al escaneo agronómico en tiempo real. Esta pantalla incluyó un marco visual para ayudar al usuario a ubicar correctamente la hoja de plátano frente a la cámara, además de los controles necesarios para realizar la captura de la imagen. • Tiempo de desarrollo: 24 horas. • Fecha de inicio y finalización: 20/01/2026 - 23/01/2026. • Ejecución de pruebas y corrección de errores de integración. Finalmente, se llevaron a cabo pruebas completas del flujo principal de la aplicación. Se verificó que el usuario pudiera iniciar sesión, acceder a la cámara sin inconvenientes y capturar una imagen desde el dispositivo. Durante estas pruebas también se identificaron y corrigieron fallos relacionados con el consumo de memoria y la integración entre los distintos componentes. • Tiempo de desarrollo: 16 horas. • Fecha de inicio y finalización: 24/01/2026 - 25/01/2026.

4.4.8.2 Sprint Backlog

Tabla 24: Sprint Backlog del Sprint 1.

Historia de usuario	Tarea	Duración estimada
HU-01	Crear la estructura inicial del backend con Node.js, TypeScript y Express.	8 horas
HU-01	Configurar conexión con MongoDB Atlas para la gestión de usuarios.	6 horas
HU-01	Implementar registro e inicio de sesión mediante correo y contraseña.	10 horas

Historia de usuario	Tarea	Duración estimada
HU-01	Generar y validar tokens JWT para proteger rutas del sistema.	8 horas
HU-01	Guardar sesión local del usuario en la aplicación móvil.	6 horas
HU-02	Configurar permisos de cámara en Android.	5 horas
HU-02	Integrar cámara en Flutter para capturar imágenes.	10 horas
HU-02	Diseñar la pantalla inicial de escaneo de planta.	12 horas
HU-02	Implementar selección visual del tipo de análisis.	8 horas
HU-02	Validar navegación entre dashboard, cámara y pantalla de resultado.	6 horas

4.4.8.3 Desarrollo del Sprint

4.4.8.3.1 Funcionalidad de acceso al sistema (Autenticación)

El sistema permite que los usuarios ingresen a la aplicación AgroAR Plátano mediante un correo y una contraseña previamente registrados. Esta funcionalidad permite identificar al usuario que realiza los análisis de las plantas y asociar los registros guardados en SQLite y MongoDB con una cuenta específica, el sistema contempla dos usuarios: estudiante y administrador. Al iniciar sesión, el backend valida las credenciales, genera un token JWT y la aplicación guarda la sesión localmente para mantener el acceso activo.

Evidencia de la funcionalidad



Ilustración 14: Pantalla de inicio de sesión de AgroAR Plátano

Aquí se observa el formulario donde el usuario ingresa su correo y contraseña para acceder al sistema. También se puede agregar una captura de la respuesta del backend en Postman, donde se evidencia que el inicio de sesión fue exitoso y que el servidor devuelve el token JWT junto con los datos del usuario.

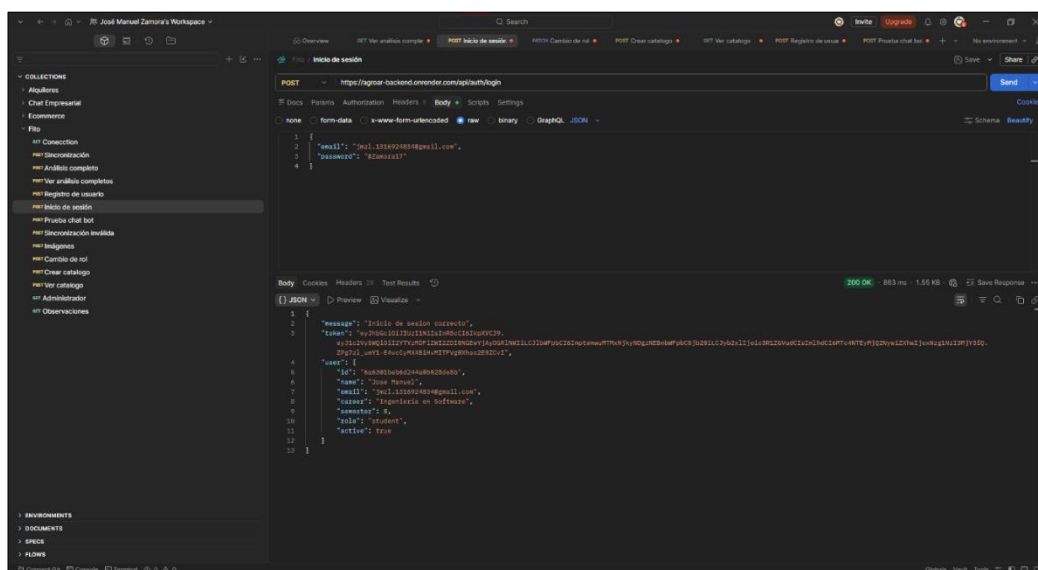


Ilustración 15: Respuesta del endpoint de inicio de sesión en Postman.

La respuesta verifica que las credenciales eran correctas y que el sistema creó un inicio de sesión para el usuario.

Código relacionado

```
authRouter.post(
  "/login",
  asyncHandler(async (request, response) => {
    const result = loginSchema.safeParse(request.body);

    if (!result.success) {
      response.status(400).json({
        message: "Datos de inicio de sesión inválidos",
        errors: result.error.flatten(),
      });
      return;
    }

    const user = await User.findOne({
      email: result.data.email.toLowerCase(),
    });

    if (!user) {
      response.status(401).json({
        message: "Credenciales incorrectas",
      });
      return;
    }

    if (!user.active) {
      response.status(403).json({
        message: "Usuario inactivo",
      });
      return;
    }

    const passwordMatches = await bcrypt.compare(
      result.data.password,
      user.passwordHash,
    );

    if (!passwordMatches) {
      response.status(401).json({
        message: "Credenciales incorrectas",
      });
      return;
    }
  })
);
```

```
const token = jwt.sign(
  {
    userId: user._id.toString(),
    email: user.email,
    role: user.role,
  },
  env.jwtSecret,
  {
    expiresIn: "7d",
  }
);

response.json({
  message: "Inicio de sesión correcto",
  token,
  user: {
    id: user._id,
    name: user.name,
    email: user.email,
    career: user.career,
    semester: user.semester,
    role: user.role,
    active: user.active,
  },
});
```

4.4.8.4 Sprint Review

Esta característica permite que los usuarios registrados accedan a su cuenta utilizando correo electrónico y contraseña, verificando sus credenciales desde el servidor, la implementación abarcó la vinculación entre la aplicación móvil creada en Flutter y el backend construido en Node.js con TypeScript, el backend valida los datos recibidos, consulta la colección `users` en MongoDB, verifica la contraseña cifrada y genera un token JWT para mantener una sesión segura. En la aplicación móvil, la sesión se guarda localmente en SQLite, permitiendo conservar el acceso del usuario en el dispositivo.

Se revisó la colección `users` en MongoDB Atlas para comprobar que los usuarios registrados se almacenan correctamente, durante la revisión se recibió como retroalimentación que la funcionalidad debía diferenciar correctamente el acceso entre estudiantes y administradores, ya que ambos usuarios no deben ingresar a las mismas pantallas. También se

indicó que la sesión debía mantenerse en el dispositivo para evitar que el usuario tenga que iniciar sesión cada vez que abra la aplicación.

A partir de esta retroalimentación se realizaron mejoras en la navegación posterior al inicio de sesión. Se ajustó la validación del rol del usuario para redirigir al administrador hacia su panel correspondiente y al estudiante hacia el panel de escaneo. Con esta revisión se confirmó que la funcionalidad de acceso al sistema cumple con el objetivo planteado para el sprint, ya que permite autenticar usuarios, controlar el acceso por roles y mantener la sesión activa en la aplicación móvil.

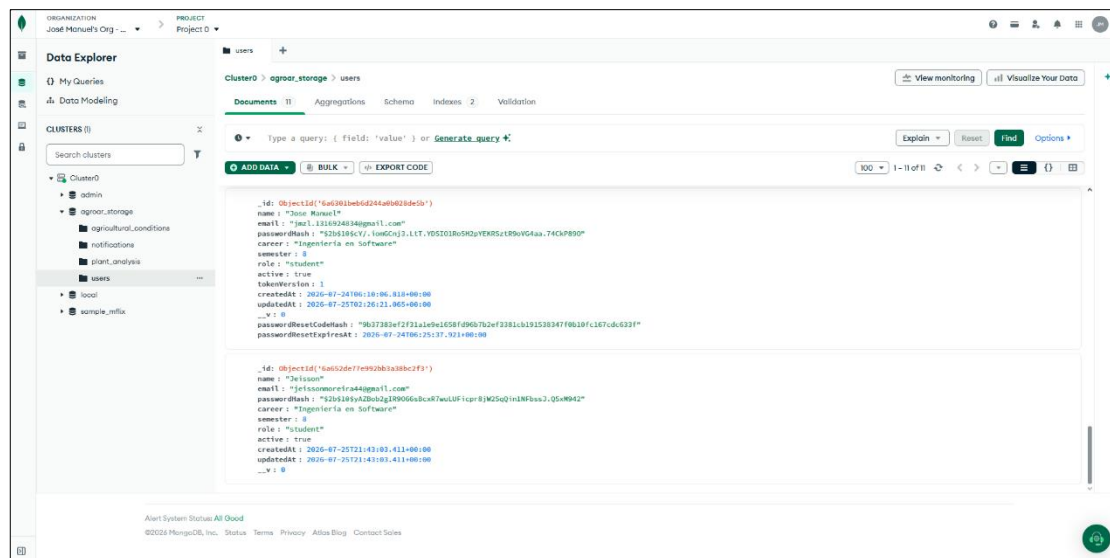


Ilustración 16: Captura de la colección users en MongoDB Atlas.

4.4.8.5 Pruebas del Sprint

4.4.8.5.1 Pruebas de Caja Negra

➤ Prueba de formulario de registro de usuario

Tabla 25: Prueba del formulario de registro de usuario.

Nombre del campo	Tipo de campo	Valor permitido	Observación
Nombre	Texto	Nombre completo del usuario	No debe enviarse vacío.

Nombre del campo	Tipo de campo	Valor permitido	Observación
Correo electrónico	Texto	Correo válido y no registrado previamente	Si el correo ya existe, el backend rechaza el registro.
Contraseña	Password	Texto obligatorio	Debe ingresarse para crear la cuenta.
Confirmar contraseña	Password	Debe coincidir con la contraseña	Si no coincide, el registro no debe continuar.
Carrera	Texto	Nombre de la carrera	Se almacena como dato informativo del usuario.
Semestre	Numérico	Valor entre 1 y 10	Permite identificar el nivel académico del estudiante.

Resultado esperado:

El sistema debe permitir capturar o seleccionar una imagen y generar un resultado de análisis.

Resultado obtenido:

El sistema permitió realizar escaneos desde cámara y galería, mostrando resultados relacionados con plagas, hongos y salud vegetal según el tipo de análisis seleccionado.

○ Prueba del almacenamiento local

Tabla 26: Prueba del almacenamiento local en SQLite.

Elemento probado	Tipo de proceso	Valor permitido	Observación
Registro de análisis	Proceso local	Imagen, fecha, resumen, riesgo y nota	La información se guarda en SQLite.

Elemento probado	Tipo de proceso	Valor permitido	Observación
Estado de sincronización	Texto	pending o synced	Permite conocer si el registro ya fue enviado al backend.
Historial local	Consulta	Registros del usuario autenticado	Muestra solo los análisis asociados al usuario actual.

Resultado esperado:

El sistema debe guardar los análisis localmente, aunque no exista conexión a internet.

Resultado obtenido:

Los análisis se guardaron correctamente en SQLite y quedaron disponibles en el historial local

4.4.8.5.2 Pruebas de Caja Blanca

○ Pruebas de sincronización con MongoDB Atlas

Tabla 27: Pruebas de sincronización con MongoDB Atlas.

Método	Acción esperada	Acción obtenida	Observación
syncPending()	Enviar registros pendientes desde SQLite al backend	Los registros pendientes fueron enviados al endpoint /api/analysis/sync	La sincronización permitió pasar datos locales hacia MongoDB Atlas.
toSyncJson()	Convertir el análisis local en	El JSON incluyó datos como	La estructura enviada coincidió

Método	Acción esperada	Acción obtenida	Observación
	un JSON compatible con el backend	localUuid, imagen, fecha, riesgo, resumen y resultados técnicos	con lo esperado por el backend.
markScanReportAsSynced()	Cambiar el estado local del registro luego de sincronizar	El registro cambió de pending a synced	Esto evita que el mismo análisis se envíe repetidamente.
upsertRemoteScanReport()	Insertar o actualizar registros descargados desde MongoDB	La base local se actualizó sin duplicar registros	Se utilizó localUuid y mongoId para controlar duplicados.

○ **Pruebas de carga de imágenes**

Tabla 28: Pruebas de carga de imágenes en el sistema.

Método	Acción esperada	Acción obtenida	Observación
uploadImage()	Que la imagen registrada llegue al backend.	La transferencia se concretó a través de una petición de tipo multipart.	Gracias a este método fue posible adjuntar la evidencia visual generada durante el análisis.

Método	Acción esperada	Acción obtenida	Observación
Ruta /api/uploads/images	Que el backend reciba la imagen y la procese.	El servidor admitió los archivos con formato correcto y desartó aquellos con extensiones no autorizadas.	La validación del tipo de archivo evitó que se cargaran datos incorrectos al sistema.
Integración con Cloudinary	Que la imagen quede almacenada en la nube.	Se obtuvo como resultado una URL pública asociada a la imagen.	Esta URL hace posible acceder a la evidencia desde cualquier dispositivo.
updateScanReportImageUrl()	Guardar la URL de Cloudinary en SQLite	La URL quedó asociada al análisis local	La imagen sincronizada puede consultarse posteriormente.

4.4.8.6 Incremento y entregables del sprint

Al finalizar el sprint se obtuvo un incremento funcional del sistema AgroAR, enfocado en el acceso al sistema, el registro de usuarios, la gestión de sesión local y la navegación inicial según el rol del usuario. Este incremento permitió que la aplicación cuente con una base de autenticación funcional para diferenciar el acceso entre estudiantes y administradores.

Tabla 29: Incremento funcional obtenido al finalizar el Sprint 1.

Proceso o función implementada	Descripción	Estado
Registro de usuarios	Se implementó el formulario para registrar usuarios con nombre, correo, contraseña, confirmación de contraseña, carrera y semestre.	Completado
Inicio de sesión	Se implementó el acceso mediante correo y contraseña, validando las credenciales desde el backend.	Completado
Validación de credenciales	El backend valida el formato de los datos recibidos y verifica si el usuario existe en MongoDB.	Completado
Cifrado de contraseña	Ninguna contraseña se guarda en texto plano, el sistema aplica bcrypt para cifrarse antes de almacenarlas.	Completado
Generación de token JWT	Una vez que el inicio de sesión validó correctamente, el backend responde generando un token JWT que sostiene la sesión de forma segura.	Completado
Control de roles	Según el tipo de usuario que ingresa, el sistema reconoce su rol y lo dirige automáticamente a la pantalla que le corresponde.	Completado
Sesión local en SQLite	Para que el acceso se mantenga activo dentro del dispositivo, la aplicación conserva localmente los datos de la sesión ya autenticada.	Completado
Validación de usuario activo	El sistema verifica que el usuario se encuentre activo antes de permitir el acceso.	Completado

4.4.8.7 Estado actual del Product Backlog

Tabla 30: Estado actual del Product Backlog.

Código	Historia de usuario	Prioridad	Estado actual	Observación
HU-01	Como usuario, quiero registrarme e iniciar sesión en el sistema para acceder a las funcionalidades de AgroAR Plátano según mi rol.	Alta	Completada	Se implementó el registro, inicio de sesión, validación de credenciales, JWT, sesión local en SQLite y redirección por rol.
HU-02	Como estudiante, quiero escanear una planta de plátano mediante cámara o galería para obtener un análisis de plagas, hongos y salud vegetal.	Alta	Completada	Se implementó el escaneo mediante cámara y galería, junto con la visualización de resultados técnicos del análisis.
HU-03	Como estudiante, quiero que los análisis se guarden localmente cuando no tenga conexión a internet para no perder información en campo.	Alta	Completada	Se implementó el almacenamiento local en SQLite, permitiendo guardar análisis sin conexión y conservar los registros con estado pendiente o sincronizado.

Código	Historia de usuario	Prioridad	Estado actual	Observación
HU-04	En calidad de usuario, necesito que los análisis realizados de forma local se sincronicen con MongoDB Atlas en cuanto se restablezca la conexión, de modo que toda la información quede centralizada en un solo lugar.	Alta	Completada	La sincronización se resolvió a través del backend alojado en Render, un proceso que además contempla el control de registros duplicados, la carga de las imágenes hacia Cloudinary y la actualización correspondiente del estado guardado localmente.
HU-05	En calidad de usuario, necesito acceder al historial de revisiones ya realizadas y contar con la posibilidad de generar reportes PDF, de manera que pueda revisar el trabajo previo y disponer de evidencia	Media	Completada	Se implementó el historial de revisiones, la visualización del detalle del análisis y la generación de reportes PDF con la información registrada.

Código	Historia de usuario	Prioridad	Estado actual	Observación
	técnica del análisis al momento de entregarlo.			

4.4.9 Resultado final del proyecto

4.4.9.1 Funcionalidades implementadas

En esta etapa se implementaron las funcionalidades principales del sistema AgroAR, tomando como referencia las historias de usuario definidas en el Product Backlog. El resultado fue un prototipo capaz de registrar usuarios, analizar plantas, guardar revisiones, sincronizar información y generar evidencias técnicas del proceso realizado.

Tabla 31: Funcionalidades implementadas en el prototipo AgroAR.

Funcionalidad implementada	Descripción	Estado
Registro de usuarios	Permite crear una cuenta de usuario con nombre, correo electrónico, contraseña, carrera y semestre. Los usuarios registrados se almacenan en MongoDB Atlas.	Implementada
Inicio de sesión	Permite el acceso al sistema mediante correo electrónico y contraseña, el backend valida las credenciales y genera un token JWT.	Implementada
Control de roles	Permite diferenciar el acceso entre estudiantes y administradores, el estudiante accede al panel de análisis y el administrador al panel de gestión.	Implementada
Sesión local	Facilita la conservación del acceso activo en el dispositivo al almacenar la sesión del usuario en SQLite.	Implementada

Funcionalidad implementada	Descripción	Estado
Escaneo mediante cámara	Permite capturar una imagen de la planta desde la cámara del dispositivo móvil.	Implementada
Análisis desde galería	Permite seleccionar una imagen existente en el dispositivo para realizar el análisis.	Implementada
Selección de tipo de análisis	El usuario puede decidir si el análisis se orienta a plagas, hongos, salud vegetal o a una evaluación general de la planta.	Implementada
Detección de plagas	A través de un módulo basado en IA, el sistema es capaz de reconocer indicios de picudo negro en las plantas de plátano.	Implementada
Evaluación de salud vegetal	El mismo enfoque de IA se emplea para determinar si la planta se encuentra en buen estado o presenta alguna condición comprometida.	Implementada
Registro de análisis agrícola	Cada escaneo queda documentado junto con su imagen, la fecha en que se realizó, un resumen del caso, el nivel de riesgo detectado, una nota adicional y los resultados técnicos obtenidos.	Implementada
Almacenamiento local en SQLite	Los análisis pueden conservarse directamente en el dispositivo móvil, lo que hace posible seguir trabajando aun sin conexión.	Implementada
Sincronización con MongoDB Atlas	En cuanto se restablece la conexión, los registros guardados localmente se transfieren hacia la base de datos central.	Implementada

Funcionalidad implementada	Descripción	Estado
Almacenamiento de imágenes en Cloudinary	Las imágenes capturadas se cargan a la nube, y la dirección URL resultante queda registrada en MongoDB Atlas.	Implementada
Historial de revisiones	El usuario autenticado puede revisar el conjunto de análisis que ha realizado a lo largo del tiempo.	Implementada
Filtros del historial	Las revisiones pueden organizarse según distintos criterios, aquellas que requieren atención, o las de carácter preventivo.	Implementada
Detalle de revisión	Es posible abrir cualquier análisis y ver su información completa, imagen, resultados técnicos, nota registrada y recomendaciones asociadas.	Implementada
Generación de reporte PDF	A partir del análisis elegido, el sistema construye automáticamente un reporte técnico en formato PDF.	Implementada
Chatbot agrícola	El usuario cuenta con un canal de consulta para resolver dudas sobre el cultivo de plátano y recibir orientación técnica.	Implementada
Historial del chatbot	Permite revisar consultas anteriores realizadas al asistente agrícola.	Implementada
Panel administrativo	Permite al administrador consultar métricas generales y gestionar información del sistema.	Implementada

4.4.9.2 Evidencia final del sistema

- **Evidencia 1. Pantalla de inicio de sesión**

En esta pantalla el usuario ingresa su correo electrónico y contraseña para acceder al sistema.



Ilustración 17: Pantalla de inicio de sesión de la aplicación móvil.

- **Evidencia 2. Pantalla de registro de usuario**

Esta pantalla permite crear una cuenta de usuario estudiante, ingresando datos como nombre, correo electrónico, contraseña, confirmación de contraseña, carrera y semestre. La información registrada se almacena en la colección users de MongoDB Atlas.



Ilustración 18: Pantalla de registro de usuario.

- **Evidencia 3. Panel principal del estudiante**

El panel principal permite al estudiante acceder al análisis de plantas, historial de revisiones, chatbot agrícola y perfil de usuario.



Ilustración 19: Dashboard o pantalla principal del estudiante.

- **Evidencia 4. Pantalla de escaneo de planta**

En esta pantalla el usuario puede realizar un análisis usando la cámara del dispositivo o seleccionar una imagen desde la galería. Además, permite escoger el tipo de análisis: plagas, hongos, salud vegetal o análisis general.



Ilustración 20: Pantalla de cámara o selección de análisis.

- **Evidencia 5. Resultado del análisis agrícola**

Esta pantalla mostró los resultados obtenidos después del análisis de la planta. El sistema presentó información relacionada con plagas, hongos, salud vegetal, nivel de riesgo y acciones recomendadas según la revisión realizada.

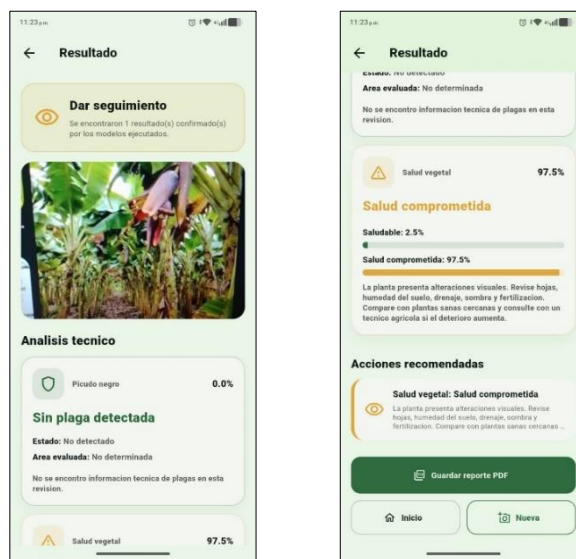


Ilustración 21: Pantalla de resultado del análisis.

- **Evidencia 6. Historial de revisiones**

El historial permite consultar los análisis realizados por el usuario autenticado. Las revisiones pueden filtrarse por todas, atención y preventivas, facilitando la consulta de registros anteriores.



Ilustración 22: Pantalla del historial de revisiones.

- **Evidencia 7. Detalle de revisión guardada**

En esta pantalla se visualiza la información completa de un análisis guardado, incluyendo imagen, fecha, nivel de riesgo, resultados técnicos, nota del usuario y recomendaciones.

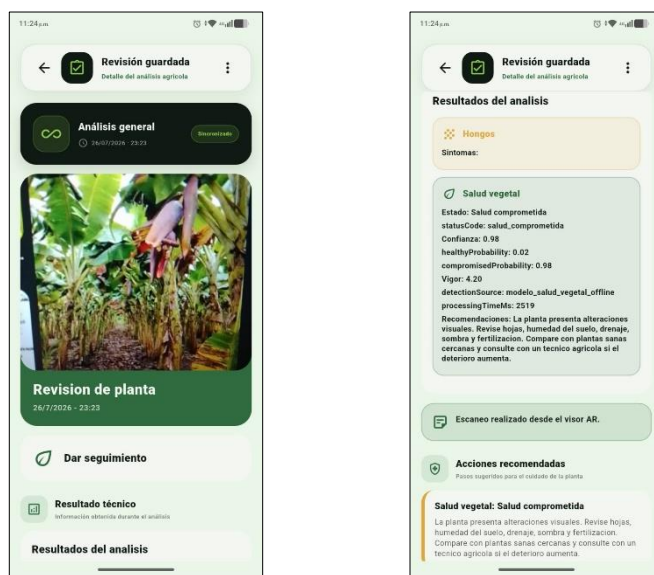


Ilustración 23: Pantalla de detalle de una revisión.

- **Evidencia 8. Reporte PDF generado**

El sistema permite generar un reporte PDF con los datos del análisis agrícola. Este reporte incluye información técnica del escaneo, imagen de evidencia, resultados y recomendaciones.



Ilustración 24: Vista del PDF generado o archivo PDF abierto.

- **Evidencia 9. Chatbot agrícola**

El chatbot permite realizar consultas relacionadas con el cultivo de plátano y recibir recomendaciones técnicas de apoyo. También se guarda un historial de consultas realizadas por el usuario.

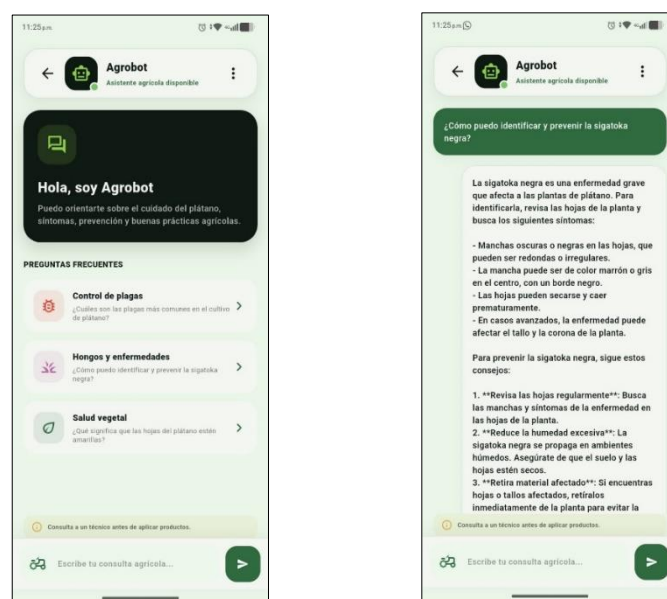


Ilustración 25: Pantalla del chatbot agrícola.

- **Evidencia 10. Panel administrativo**

El panel administrativo permite visualizar información general del sistema y consultar métricas relacionadas con usuarios, análisis registrados y datos almacenados.



Ilustración 26: Pantalla del panel administrativo.

- **Evidencia 11. Registros en MongoDB Atlas**

La base de datos central almacena la información sincronizada desde la aplicación móvil. En MongoDB Atlas se pueden observar las colecciones principales, como usuarios, análisis de plantas y condiciones agrícolas.

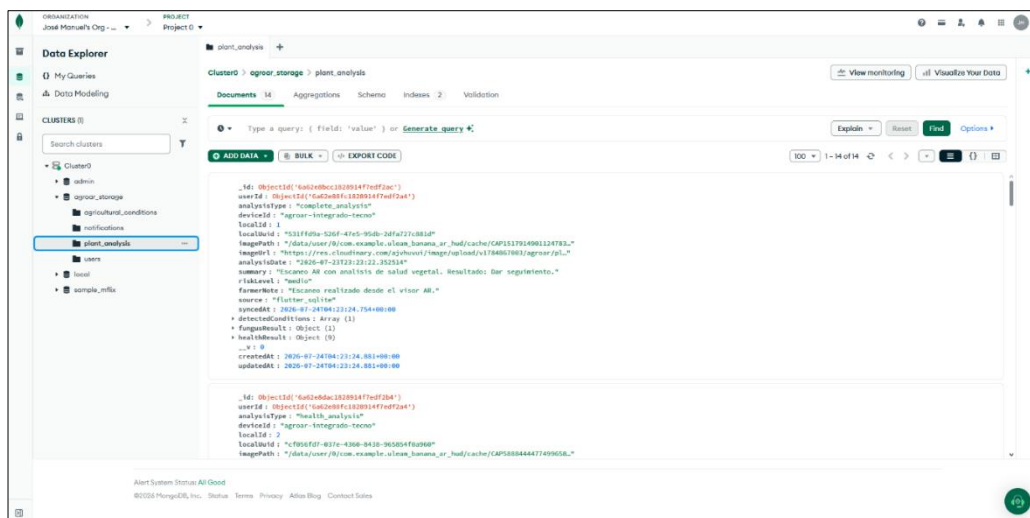


Ilustración 27: MongoDB Atlas mostrando las colecciones y documentos registrados.

- **Evidencia 12. Imagen almacenada en Cloudinary**

Las imágenes capturadas durante los análisis son enviadas a Cloudinary. El sistema guarda la URL generada y la asocia al análisis correspondiente en MongoDB Atlas.

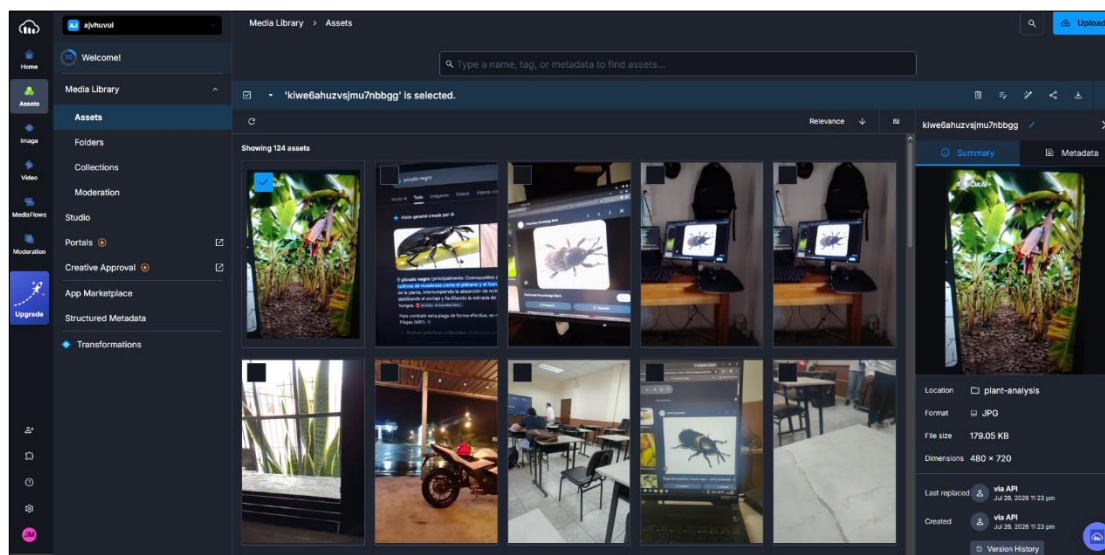


Ilustración 28: Imagen subida en Cloudinary o campo imageUrl dentro de MongoDB Atlas.

- **Evidencia 13. Backend desplegado en Render**

El backend se encuentra desplegado en Render, permitiendo que la aplicación móvil funcione desde diferentes redes sin depender de un servidor local. Desde este servicio se gestionan las rutas de autenticación, análisis, sincronización, chatbot, administración y carga de imágenes.

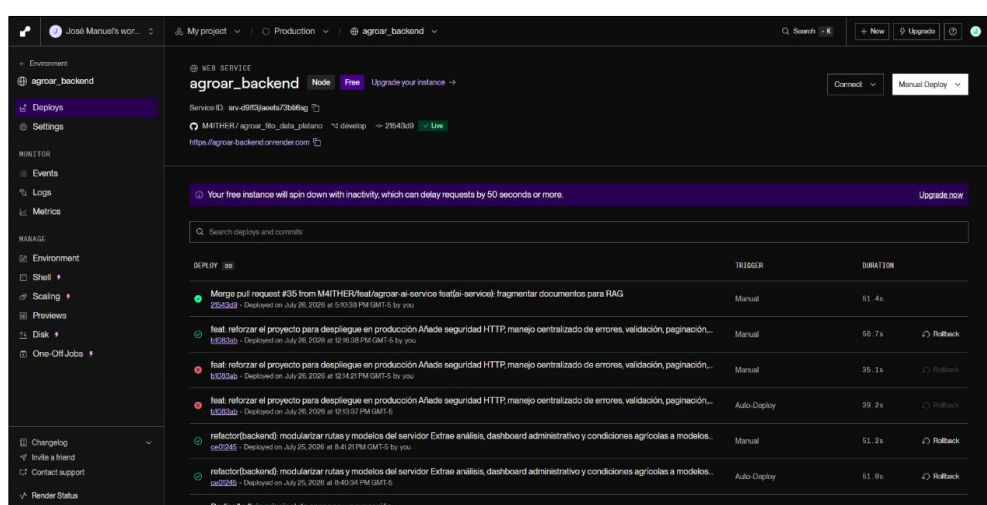


Ilustración 29: Panel de Render mostrando el backend activo o respuesta del endpoint /api/health.

4.4.9.3 Conclusiones finales del sistema

El desarrollo del prototipo AgroAR permitió construir una aplicación móvil funcional orientada al apoyo del cultivo de plátano mediante el registro, análisis y almacenamiento de información agrícola. La aplicación integra autenticación de usuarios, escaneo mediante cámara o galería, análisis de plagas, evaluación de salud vegetal, almacenamiento local, sincronización en la nube, historial de revisiones, chatbot agrícola y generación de reportes PDF.

La arquitectura implementada permitió combinar tecnologías móviles, backend y bases de datos de forma integrada. Flutter fue utilizado para el desarrollo de la aplicación móvil, SQLite para el almacenamiento local sin conexión, Node.js con TypeScript para el backend,

MongoDB Atlas para la base de datos central y Cloudinary para el almacenamiento de imágenes. Esta combinación permitió que el sistema funcione en escenarios con conectividad limitada y posteriormente sincronice la información cuando exista acceso a internet.

El uso de SQLite en el dispositivo móvil permitió que los análisis agrícolas no se pierdan cuando el usuario trabaja en campo sin conexión. Esta característica es importante porque la aplicación está pensada para entornos agrícolas donde la conectividad puede ser inestable. En el momento en que se recupera la conexión a internet, los registros almacenados encuentran su camino hacia MongoDB Atlas, lo que hace posible reunir toda la información en un único punto y conservar, al mismo tiempo, un respaldo confiable de cada análisis realizado.

Detrás de esta flexibilidad se encuentra el uso de esquemas polimórficos dentro de MongoDB, un recurso que hizo posible que distintos tipos de análisis convivieran dentro de una misma estructura de datos. Gracias a ello, la información sobre plagas, hongos, salud vegetal o análisis general puede registrarse sin necesidad de ajustarse a un esquema rígido de tablas predefinidas. Esta decisión de diseño no fue arbitraria: responde directamente a la naturaleza cambiante de los datos fitotécnicos, cuya estructura varía según el tipo de revisión que se esté llevando a cabo.

Por su parte, trabajar bajo la metodología Scrum le dio al desarrollo un orden claro, organizado en torno a historias de usuario, tareas específicas, un sprint backlog, ciclos de prueba e incrementos funcionales sucesivos. Este enfoque hizo posible avanzar de manera progresiva, someter cada funcionalidad a validación por etapas y corregir los errores conforme iban apareciendo durante el proceso. A ello se suma un beneficio adicional: mantener en todo momento una visión clara de cuánto se había avanzado en el prototipo y qué funcionalidades seguían pendientes.

CAPÍTULO V

5 EVALUACIÓN DE RESULTADOS

5.1 Introducción

En este capítulo se comprobó el comportamiento de las funcionalidades implementadas en el prototipo AgroAR, la evaluación se enfocó en comprobar si los procesos clave del sistema funcionaron adecuadamente durante su utilización, desde el inicio de sesión hasta la sincronización y revisión de una consulta agrícola.

Se tomaron en cuenta pruebas funcionales y técnicas aplicadas a la aplicación móvil, el backend y las bases de datos empleadas, se probó en dispositivos Android evaluando el comportamiento de la autenticación, el escaneo mediante cámara o galería, el almacenamiento local en SQLite y sincronización en MongoDB Atlas, asimismo se confirmó la carga de imágenes hacia Cloudinary y el correcto funcionamiento implementado de backend desplegado en Render, la evaluación constató que el prototipo opera como una solución móvil offline, lo que significa que puede guardar información local sin conectividad y luego sincronizar con la base de datos central cuando hay conectividad, esta propiedad es relevante para el ámbito agrícola, ya que en áreas rurales la falta de internet es prominente.

También se revisaron funciones complementarias como el chatbot agrícola y la generación de reportes PDF. Estas pruebas permitieron comprobar que el sistema no solo muestra pantallas funcionales, sino que mantiene un flujo completo de trabajo, el usuario inicia sesión, realiza un análisis, guarda la revisión, sincroniza la información con MongoDB Atlas y puede generar un reporte como evidencia del proceso realizado.

Con esta evaluación se deja constancia de cómo respondió el prototipo frente a los procesos principales que fueron desarrollados. Para ellos se compararon las acciones esperadas con los resultados obtenidos en la aplicación móvil, el backend y las bases de datos utilizadas.

5.2 Presentación y monitoreo de resultados

Para la presentación y monitoreo de resultados se definieron varios procesos principales del sistema AgroAR. Cada proceso fue evaluado considerando el comportamiento esperado, la forma en que fue probado y el resultado obtenido durante el uso del prototipo. La evaluación se centró en comprobar que la aplicación funcione correctamente desde el punto de vista del usuario y desde el punto de vista técnico. Por esta razón, se revisaron tanto las pantallas de la aplicación móvil como los datos almacenados en SQLite, MongoDB Atlas y Cloudinary.

Los procesos evaluados fueron los siguientes

Tabla 32: Procesos principales evaluados en el sistema AgroAR.

Proceso evaluado	Descripción	Elementos verificados
Autenticación y control de acceso	Validación del registro, inicio de sesión y redirección por rol.	Formulario de login, formulario de registro, token JWT, sesión local y roles de usuario.
Escaneo y análisis de planta	Captura o selección de imagen para realizar el análisis agrícola.	Cámara, galería, tipo de análisis, resultados de plagas, hongos y salud vegetal.
Almacenamiento local offline	Guardado de análisis en SQLite cuando el dispositivo no tiene conexión.	Tabla <code>scan_reports</code> , estado <code>pending</code> , datos del usuario, imagen, nota y riesgo.
Sincronización con la nube	Los registros generados en el dispositivo viajan hacia el backend, desde donde quedan	Endpoint encargado de la sincronización, MongoDB Atlas, Cloudinary, marca de estado

Proceso evaluado	Descripción	Elementos verificados
	almacenados de forma permanente en MongoDB Atlas.	synced y mecanismo de control frente a duplicados.
Historial de revisiones	Espacio donde el usuario autenticado puede revisar el conjunto de análisis que ha generado previamente.	Lista de todas las revisiones con alternativas de filtrado, vista detallada de cada análisis e imagen correspondiente.
Generación de reporte PDF	A partir de una revisión ya almacenada, el sistema construye un documento técnico que resume dicho análisis.	Datos del análisis, imagen relacionada, resultados técnicos, recomendaciones generadas y el archivo PDF resultante.
Chatbot agrícola	Consulta de recomendaciones relacionadas con el cultivo de plátano.	Preguntas del usuario, respuesta del chatbot e historial de consultas.

El monitoreo de estos procesos permitió comprobar el estado funcional del prototipo y detectar posibles ajustes durante el desarrollo. En las pruebas se verificó que los datos ingresados desde la aplicación móvil se guarden correctamente, que los registros puedan consultarse posteriormente y que la sincronización con la nube mantenga la información centralizada.

5.2.1 Planificación de la evaluación

La evaluación del sistema AgroAR Plátano se planificó tomando como base las funcionalidades principales desarrolladas durante el prototipo. Para esta etapa se consideraron pruebas manuales realizadas desde la aplicación móvil, el backend desplegado en la nube, la

base de datos MongoDB Atlas, el almacenamiento local SQLite y los servicios complementarios utilizados por el sistema.

El propósito de esta planificación fue comprobar que cada proceso funcione de acuerdo con lo esperado, especialmente en los flujos relacionados con autenticación, escaneo de plantas, almacenamiento local, sincronización, historial, generación de reportes y asistencia mediante chatbot. También se verificó que la información registrada por el usuario se mantenga correctamente tanto en el dispositivo móvil como en la base de datos central.

Tabla 33: Planificación de la evaluación del sistema.

Proceso evaluado	Método de validación	Resultado esperado
Autenticación y control de acceso	Ingresar con usuario estudiante y usuario administrador. Probar credenciales válidas e inválidas.	El sistema debe permitir el acceso solo a usuarios registrados y dirigirlos al panel correspondiente según su rol.
Registro de usuario	Crear una cuenta con nombre, correo, contraseña, carrera y semestre.	El usuario debe almacenarse correctamente en MongoDB y permitir el inicio de sesión posterior.
Escaneo y análisis de planta	Hacer un escaneo mediante la cámara y experimentar con los modelos de análisis que hay.	El sistema debe producir un resultado del análisis que incluya el nivel de riesgo y datos fitotécnicos de acuerdo con la IA utilizada.
Almacenamiento local SQLite	Registrar evaluaciones sin depender del enlace a internet.	La revisión debe almacenarse localmente con un estado de sincronización pendiente.

Proceso evaluado	Método de validación	Resultado esperado
Sincronización con MongoDB Atlas	Conectar el móvil a internet para realizar la sincronización de manera automática o manual.	Los registros que están en estado pendientes llegan al backend para guardar los datos la base de datos central.
Almacenamiento de imágenes	Registrar una revisión con imagen tomada desde la aplicación.	Cloudinary recibe la imagen y queda asociada al registro guardado en MongoDB.
Historial de revisiones	Consultar el historial de revisiones guardadas desde su debida pantalla.	Solo se muestran los registros del usuario que se encuentra autenticado.
Generación de reporte PDF	Crear un reporte desde la pantalla de resultados.	Toda la información del análisis va detallada en el reporte PDF.
Chatbot agrícola	Consultar sobre el hongo, plagas o el cultivo del plátano.	El chatbot debe responder recomendaciones agrícolas y guardar la consulta en el historial.
Panel de administrador	Iniciar sesión con las credenciales de administrador.	El administrador debe acceder a su panel de gestión y no al panel de escaneo del estudiante.

Para la ejecución de estas pruebas se utilizó un dispositivo Android con la aplicación instalada, el backend desplegado en Render, MongoDB Atlas como base de datos principal y Cloudinary para el almacenamiento de imágenes. Además, se realizaron pruebas con conexión activa y sin conexión para validar el comportamiento offline-first del sistema.

5.2.2 Ejecución del monitoreo

La ejecución del monitoreo se realizó probando los procesos principales del prototipo AgroAR desde la aplicación móvil y verificando el comportamiento de los servicios que intervienen en el sistema. Para esta actividad se tomó en cuenta el flujo completo de uso: inicio de sesión, análisis de la planta, almacenamiento local, sincronización con la nube, consulta del historial y generación de reportes.

Durante las pruebas se observó el comportamiento del sistema tanto con conexión a internet como sin conexión, debido a que una de las características principales del prototipo es permitir el registro de información en campo, aunque el dispositivo no tenga acceso inmediato a la red. Posteriormente, al recuperar la conexión, se verificó que los datos pendientes se enviaran al backend y quedaran almacenados en MongoDB Atlas.

Tabla 34: Ejecución del monitoreo de procesos del sistema.

Proceso monitoreado	Acción realizada	Resultado obtenido
Inicio de sesión	Se ingresó con un usuario registrado desde la aplicación móvil.	El sistema permitió el acceso y mantuvo la sesión activa en el dispositivo.
Control por rol	Se inició sesión con usuario estudiante y usuario administrador.	El estudiante accedió al panel de escaneo y el administrador al panel de gestión.
Escaneo de planta	Se realizó un análisis utilizando la cámara del dispositivo.	La aplicación generó un resultado con nivel de riesgo e información del análisis.

Proceso monitoreado	Acción realizada	Resultado obtenido
Selección de tipo de análisis	Se probó el análisis de plagas, salud vegetal y análisis general.	El sistema ejecutó el flujo correspondiente según el tipo de análisis seleccionado.
Registro sin conexión	Se desactivó la conexión a internet y se registró una nueva revisión.	La revisión quedó almacenada en SQLite con estado pendiente para después sincronizarse.
Sincronización automática	Se restableció la conexión a internet después de registrar información sin conexión.	Los registros pendientes se sincronizaron automáticamente con el backend.
Almacenamiento en MongoDB	Se verificó que los registros enviados desde la aplicación se almacenaran en MongoDB Atlas.	La información quedó registrada correctamente en la colección correspondiente.
Almacenamiento de imagen	Se verificó la URL generada para la imagen del análisis.	La imagen quedó almacenada en Cloudinary y asociada al registro.
Historial de revisiones	Se accedió a la pantalla de revisiones desde la aplicación.	Se visualizaron correctamente las revisiones del usuario autenticado.
Reporte PDF	Se generó un reporte desde una revisión almacenada.	El sistema generó un archivo PDF con los datos del análisis realizado.

Proceso monitoreado	Acción realizada	Resultado obtenido
Chatbot agrícola	Se realizó una consulta sobre el cultivo de plátano.	El chatbot proporcionó una recomendación y la consulta quedó en el historial.

El monitoreo confirmó que el registro de una revisión pasa por las capas principales del sistema sin perder información. Primero se guarda en SQLite, luego se envía al backend cuando existe conexión, después se almacena en MongoDB Atlas y, en el caso de las imágenes, se genera una URL en Cloudinary que queda asociada al análisis.

5.2.2.1 Proceso de autenticación y control de acceso

En este proceso se evaluó el acceso al sistema mediante el formulario de inicio de sesión y el formulario de registro de usuarios. La finalidad de la prueba fue comprobar que la aplicación permitiese ingresar únicamente a usuarios registrados y que, después del inicio de sesión, el sistema redirija al usuario según el rol asignado.

La autenticación se validó utilizando el backend desplegado en Render y la base de datos MongoDB Atlas. Cuando el usuario inició sesión correctamente, el backend genera un token JWT y devuelve los datos principales del usuario. Posteriormente, la aplicación guarda la sesión de forma local para evitar que el usuario tenga que iniciar sesión cada vez que abre la aplicación.

Tabla 35: Evaluación del proceso de autenticación y control de acceso.

Elemento evaluado	Prueba realizada	Resultado esperado	Resultado obtenido
Registro de usuario	Se ingresaron nombre, correo, contraseña, confirmación de contraseña, carrera y semestre.	El sistema debe crear el usuario en MongoDB Atlas.	El usuario se registró correctamente y quedó almacenado en la base de datos.
Validación de campos vacíos	Se intentó registrar un usuario sin completar todos los campos obligatorios.	El sistema debe mostrar mensajes de validación.	La aplicación impidió continuar hasta completar los datos requeridos.
Confirmación de contraseña	Se ingresaron contraseñas diferentes en el formulario de registro.	El sistema debe rechazar el registro.	La aplicación mostró la validación correspondiente y no envió datos inválidos.
Inicio de sesión correcto	Se ingresó correo y contraseña válidos.	El sistema debe permitir el acceso a la aplicación.	El usuario ingresó correctamente y la sesión quedó activa.
Inicio de sesión incorrecto	Se ingresó una contraseña inválida.	El sistema debe rechazar el acceso.	El sistema no permitió iniciar sesión con credenciales incorrectas.
Control por rol	Se ingresó con un usuario estudiante y	El estudiante debe ir al panel principal y el	El sistema redirigió correctamente según el rol del usuario.

Elemento evaluado	Prueba realizada	Resultado esperado	Resultado obtenido
	con un usuario administrador.	administrador al panel administrativo.	
Sesión local	Se cerró y volvió a abrir la aplicación después de iniciar sesión.	La sesión debe mantenerse activa en el dispositivo.	La aplicación conservó la sesión local sin solicitar nuevamente las credenciales.

Como resultado de esta prueba, se comprobó que el proceso de autenticación funciona correctamente. El sistema valida los datos ingresados por el usuario, evita registros incompletos, permite iniciar sesión con credenciales válidas y aplica control de acceso según el rol registrado en la base de datos.

5.2.2.2 Proceso de escaneo y análisis de la planta

En este proceso se evaluó la funcionalidad principal de la aplicación, correspondiente al escaneo y análisis de una planta de plátano. La prueba se realizó desde la pantalla de cámara de la aplicación móvil, utilizando la opción de analizar planta y los modos de revisión disponibles dentro del prototipo.

Tabla 36: Evaluación del proceso de escaneo y análisis de la planta.

Elemento evaluado	Prueba realizada	Resultado esperado	Resultado obtenido
Acceso a cámara	Se ingresó a la pantalla de escaneo desde el panel principal.	La cámara debe abrirse correctamente y permitir iniciar el análisis.	La cámara se abrió correctamente en el dispositivo móvil.

Elemento evaluado	Prueba realizada	Resultado esperado	Resultado obtenido
Captura de imagen	Se presionó la opción de analizar planta.	El sistema debe capturar una imagen para procesarla.	La aplicación capturó la imagen y la utilizó para generar el análisis.
Análisis de plagas	Se realizó una prueba con el modo de plagas.	El sistema debe indicar si existe o no detección de plaga.	El resultado mostró la información correspondiente al análisis de plaga.
Análisis de salud vegetal	Se realizó una prueba con el modo de salud vegetal.	El sistema debe indicar el estado de salud de la planta.	El sistema presentó el resultado de salud vegetal con su nivel de confianza.
Análisis general	Se realizó una prueba con el modo general.	El sistema debe combinar los resultados disponibles de los módulos integrados.	La aplicación generó una revisión general con resultados técnicos agrupados.
Presentación del resultado	Se revisó la pantalla de resultado después del escaneo.	El usuario debe visualizar el riesgo, resumen y recomendaciones.	La pantalla mostró el resultado del análisis de forma ordenada.
Registro del análisis	Se guardó la revisión	El análisis debe quedar registrado para consulta posterior.	La revisión fue guardada en el historial local de la aplicación.

Elemento evaluado	Prueba realizada	Resultado esperado	Resultado obtenido
	generada por el escaneo.		

5.2.2.3 Proceso de sincronización con MongoDB Atlas

En este proceso se evaluó la sincronización de información entre la aplicación móvil y MongoDB Atlas, que funciona como base de datos principal del sistema. Esta prueba es importante porque AgroAR Plátano utiliza un enfoque offline-first, permitiendo que las revisiones se guarden primero en el dispositivo móvil y luego se envíen a la nube cuando existe conexión a internet.

Tabla 37: Evaluación del proceso de sincronización con MongoDB Atlas.

Elemento evaluado	Prueba realizada	Resultado esperado	Resultado obtenido
Registro local sin conexión	Se realizó un análisis de planta y se guardó la revisión en la aplicación.	Se espera que el registro se almacene en SQLite con estado pendiente de sincronización.	El registro se almacenó localmente y apareció como pendiente de sincronización.
Conservación del registro local	El registro se almacenó localmente y se mostró como un pendiente de sincronización.	El registro no procesado debe permanecer accesible en el historial local.	La evaluación permaneció accesible en el registro de la aplicación.

Elemento evaluado	Prueba realizada	Resultado esperado	Resultado obtenido
Reconexión a internet	Se recuperó la conexión del dispositivo móvil.	El sistema tiene que identificar la conexión disponible.	La aplicación identificó la conexión y permitió proseguir con la sincronización.
Envío al backend	Se actualizó el registro pendiente a través de la aplicación móvil.	El backend tiene que recibir la información enviada desde Flutter.	La API manejó adecuadamente la solicitud que envió la aplicación.
Almacenamiento en MongoDB Atlas	Se examinó la colección pertinente en MongoDB Atlas.	El almacenamiento debe incluir los datos del análisis.	La revisión se mostró adecuadamente en MongoDB Atlas.
Almacenamiento de imagen	Se comprobó la imagen asociada con el análisis.	La imagen tiene que ser cargada a Cloudinary y su enlace debe almacenarse en MongoDB.	La imagen se guardó en Cloudinary y se asoció al registro del análisis.
Cambio de estado local	Se examinó el historial tras la sincronización.	El registro tiene que pasar de pendiente a sincronizado.	La aplicación presentó la revisión como sincronizada.

5.2.2.4 Evaluación cuantitativa de los módulos de inteligencia artificial

Con el fin de complementar la evaluación cualitativa presentada en los apartados anteriores, se realizó una evaluación cuantitativa de los módulos de inteligencia artificial integrados en el sistema. Para ello, se aplicó cada módulo sobre un conjunto de imágenes de prueba con resultado previamente conocido, contrastando el resultado obtenido por la IA contra el resultado esperado, con el fin de calcular la precisión de cada módulo.

Tabla 38: Precisión de los módulos de inteligencia artificial evaluados.

Módulo de IA evaluado	N° de imágenes evaluadas	Aciertos	Precisión
Detección de plagas	15	14	93 %
Detección de hongos	15	13	86 %
Análisis de salud vegetal	15	13	86 %

Los resultados obtenidos muestran que los módulos de inteligencia artificial alcanzan un nivel de precisión aceptable para un prototipo funcional, aunque no exento de errores. Estas variaciones se atribuyen principalmente a factores externos como la calidad de la imagen, la iluminación y el ángulo de captura, tal como se detalla en el siguiente apartado.

5.2.2.4.1 Planificación de pruebas de los módulos de Inteligencia Artificial

Para garantizar la fiabilidad del diagnóstico asistido, se estructuró un plan de pruebas específico orientado a medir el rendimiento cuantitativo de los modelos de Inteligencia Artificial integrados en el sistema. El objetivo de estas pruebas fue someter cada módulo a un lote de imágenes con diagnósticos reales previamente conocidos por el experto agrónomo, permitiendo así medir la exactitud del algoritmo.

Tabla 39: Planificación de pruebas de ejecución de Inteligencia Artificial.

Proceso evaluado	Método de validación	Resultado esperado
Detección de plagas (Picudo negro) mediante IA	Procesar un lote controlado de 15 imágenes fotográficas (con presencia y ausencia confirmada de la plaga) a través del módulo de escaneo de la aplicación móvil.	El modelo debe identificar correctamente la presencia de la plaga frente a los falsos positivos, documentando el número de aciertos para calcular su porcentaje de exactitud.
Detección de hongos (enfermedades foliares)	Ingresar 15 imágenes de prueba pre-clasificadas con síntomas de afecciones fúngicas (ej. Sigatoka) en el motor de inferencia del sistema.	El algoritmo debe generar alertas fitosanitarias congruentes con la evidencia visual, permitiendo medir cuantitativamente su capacidad de detección.
Análisis de salud vegetal y vigor	Capturar y procesar 15 imágenes de plantas sanas y enfermas, contrastando la salida de la IA con el diagnóstico agronómico real del lote.	El sistema debe clasificar correctamente el estado de la planta (ej. "Salud comprometida"), estableciendo una métrica de precisión basada en el total de imágenes evaluadas.

5.3 Interpretación objetiva de resultados

El monitoreo llevado a cabo durante esta etapa arroja una conclusión clara: el prototipo AgroAR cumplió con los procesos centrales que se habían definido para esta fase del proyecto.

A través de la aplicación, fue posible registrar el análisis de una planta, conservar esa información en el dispositivo, enviarla posteriormente hacia la nube y, más adelante, recuperarla desde el historial cuando se necesite. Este comportamiento revela un aspecto clave, el sistema no necesita conectividad continua a internet para operar, un requisito esencial considerando el tipo de ambiente agrícola en el que se empleará.

Entre los descubrimientos más relevantes de esta etapa se halla la verificación de manera exitosa del guardado local a través de SQLite, en las pruebas efectuadas, las revisiones almacenadas sin conexión se mantuvieron en el dispositivo, reteniendo su condición de pendientes hasta que la conectividad se restablezca, el resultado apoya la noción de que la aplicación puede continuar funcionando en el campo, incluso cuando la red experimenta interrupciones, adicionalmente, se evaluó la comunicación entre la aplicación móvil y el servidor, los datos enviados desde Flutter llegan sin inconvenientes a la API desarrollada en Node.js, utilizando TypeScript, y posteriormente se guardaron en MongoDB Atlas, asimismo, las imágenes vinculadas a los registros se guardaron a través de Cloudinary, lo que permite que no sean dependientes exclusivamente de una ruta local del móvil.

En cuanto a la organización de los datos, el uso de MongoDB permitió guardar estructuras flexibles para los distintos tipos de análisis del cultivo de plátano. Esto resulta útil porque no todos los escaneos generan exactamente la misma información, algunos pueden incluir resultados de plagas, otros de salud vegetal y otros un análisis general, esta adaptabilidad se alinea con el enfoque del sistema, puesto que el prototipo incorpora varios módulos de inteligencia artificial.

En el plano funcional, el sistema respondió de manera satisfactoria a lo largo de sus flujos principales autenticación, escaneo, registro, sincronización, historial, generación de PDF y consulta al chatbot. No obstante, las pruebas también dejaron al descubierto un matiz importante: la precisión de los modelos de inteligencia artificial no es constante, sino que

fluctúa en función de variables como la calidad de la imagen capturada, las condiciones de iluminación, el enfoque logrado por la cámara y las características propias del dispositivo empleado. Justamente por esta razón, lo que arroja la IA debe entenderse como un apoyo técnico que orienta la revisión, nunca como un diagnóstico cerrado o definitivo.

En cuanto a su aporte real, el mayor valor del prototipo radica en cómo organiza la información agrícola. Frente al escenario tradicional donde la información quedaba dispersa entre apuntes manuales o archivos independientes, la aplicación reúne en un único flujo la imagen capturada, la fecha del registro, el nivel de riesgo identificado, la observación del usuario y el resultado del análisis correspondiente, y esto permite examinar el estado de una planta para mantener futuras revisiones en la Granja Experimental de la Uleam.

CAPÍTULO VI

6 CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

Se concluye que el análisis del contexto agrícola y tecnológico a nivel macro, meso y micro permitió determinar que la falta de digitalización en el registro de datos fitotécnicos es un problema recurrente, evidenciado tanto en la desactualización de los procesos productivos a nivel nacional como en la dispersión de la información en documentos de Excel y registros manuales dentro de la Granja Experimental Uleam El Carmen, lo cual estableció los fundamentos y la dirección de la presente investigación.

La revisión de literatura especializada permitió fundamentar teóricamente que las arquitecturas de bases de datos NoSQL, particularmente mediante la aplicación de esquemas polimórficos, constituyen una alternativa más flexible que los modelos relacionales tradicionales para representar variables biológicas heterogéneas, sustentando la viabilidad técnica del modelado de datos fitotécnicos propuesto.

El análisis de la situación actual mediante la encuesta aplicada a los estudiantes y la entrevista al docente especialista permitió identificar que el registro manual de las variables fitotécnicas genera dificultades de trazabilidad y pérdida de información, confirmando la necesidad de un modelado de datos flexible capaz de adaptarse a la diversidad de variables biológicas observadas en campo.

Se logró crear el sistema informático AgroAR fundamentado en esquemas polimórficos, mediante la aplicación de la metodología ágil Scrum, integrando una aplicación móvil desarrollada en Flutter, un backend en Node.js con TypeScript y una base de datos documental en MongoDB Atlas, lo que permitió una gestión efectiva y escalable de la información biológica del cultivo de plátano.

Las pruebas de aceptación de usuario y la validación técnica de los esquemas polimórficos confirmaron que el sistema cumple con los requerimientos funcionales identificados en la Granja Experimental, aunque también evidenciaron un margen de mejora en la precisión de los módulos de inteligencia artificial, cuyo rendimiento depende de la cantidad y variedad de datos reales disponibles para su entrenamiento.

6.2 Recomendaciones

A partir de las pruebas realizadas se recomienda continuar con el entrenamiento de los modelos de inteligencia artificial utilizando imágenes reales del cultivo de plátano en diferentes condiciones de campo, con el fin de mejorar la precisión de los análisis y reducir resultados poco confiables.

Se recomienda, como siguiente paso, realizar nuevas rondas de pruebas de la aplicación con estudiantes, docentes y otras personas relacionadas con las actividades diarias de la Granja Experimental Uleam El Carmen, este ejercicio ayudaría a verificar lo fácil que es utilizar la aplicación y a partir de eso, ajustar funciones para que se adapten a las características reales del entorno.

Es recomendable mantener y fortalecer el enfoque offline-first del sistema, enfocándose especialmente en mejorar la sincronización local y central, además de enriquecer el contenido de los reportes en formato PDF, de este modo, la información levantada en campo podrá no solo conservarse de manera confiable, sino también presentarse con solidez como evidencia técnica del trabajo realizado.

Se recomienda a las autoridades de la Granja Experimental Uleam Extensión El Carmen evaluar la adopción formal del prototipo AgroAR como herramienta de apoyo en las prácticas agrícolas, así como gestionar los recursos necesarios para su mantenimiento y eventual escalamiento a otros cultivos gestionados por la institución.

REFERENCIAS

- Arias, J., & Covinos, M. (2021). *Diseño y metodología de la investigación*. https://gc.scalahed.com/recursos/files/r161r/w26022w/Arias_S2.pdf
- Balcázar Daza, A. M., & Arbeláez, J. E. (2025). Necesidades biotecnológicas de los potenciales agrícolas del departamento de Cundinamarca para el uso de las microfinanzas. *Criterio Libre*, 23(42), 74-97.
- Bhuiyan, Md. A. B., Abdullah, H. M., Arman, S. E., Saminur Rahman, S., & Al Mahmud, K. (2023). BananaSqueezeNet: A very fast, lightweight convolutional neural network for the diagnosis of three prominent banana leaf diseases. *Smart Agricultural Technology*, 4, 100214. <https://doi.org/10.1016/j.atech.2023.100214>
- Carvalho, H. F. de S., Bezerra, L. M. C., & Rosa, F. de F. (2025). Database Models in Agriculture: A systematic review. *Cadernos de Ciência & Tecnologia*, e27757-e27757.
- Casciaro, M., & Mammino, L. (2020). *Node.js Design Patterns: Design and implement production-grade Node.js applications using proven patterns and techniques*. Packt Publishing Ltd.
- Castro, L. M., Calvas, B., & Knoke, T. (2015). Ecuadorian Banana Farms Should Consider Organic Banana with Low Price Risks in Their Land-Use Portfolios. *PLOS ONE*, 10(3), e0120384. <https://doi.org/10.1371/journal.pone.0120384>
- Chávez Abad, R. (2015). *Introducción a la Metodología de la Investigación*. Repositorio UTMACHALA (Universidad Técnica de Machala).
- Cortazar, S. M. U., Wolf, E. D., & González, I. A. (2017). Effect of plant density on growth and yield in Barraganete plantain (*Musa paradisiaca* (L.) AAB cv. Curare enano) for a single harvest cutting in Provincia de Los Ríos, Ecuador. *Acta Agronómica*, 66(3), 367-372. <https://doi.org/10.15446/acag.v66n3.52198>

- Crescente, J. M., Guidobaldi, F., Demichelis, M., Formica, M. B., Helguera, M., & Vanzetti, L. S. (2017). Phenobook: An open source software for phenotypic data collection. *GigaScience*, 6(4), giw019. <https://doi.org/10.1093/gigascience/giw019>
- F, L., R. ., Peralta, E. L. ., López, V. ., López, D. ., Brunel, S. ., Dusunceli. (2022). *Estrategia y Plan de Acción Regional para la Preparación, Prevención, Detección, Respuesta y Recuperación de América Latina y el Caribe a la Marchitez por Fusarium de las Musáceas – Raza 4 Tropical*. Food & Agriculture Org.
- Fernández, F., Pico, J. T., & Avellán Cedeño, B. A. (2021). *Guía para la producción y manejo integrado del cultivo de plátano*. Lago Agrio. EC:INIAP-Estación Experimental de la Amazonia, 2021. <http://repositorio.iniap.gob.ec/handle/41000/5825>
- Ghiwari, S., Sambrekar, K., & Rajpurohit, V. S. (2017). Hierarchical Storage for Agro Informatics System Using NoSQL Technology. *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, 1-5. <https://doi.org/10.1109/ICCUBEA.2017.8463693>
- Giamas, A. (2022). *Mastering MongoDB 6.x: Expert techniques to run high-volume and fault-tolerant database solutions using MongoDB 6.x*. Packt Publishing Ltd.
- Guía técnica para el cultivo del plátano (Musa sp.). Estación Experimental «Dean Padgett B.»* (s. f.). IICA Biblioteca Venezuela.
- GUIA_CULTIVO_PLATANO_2011-libre.pdf*. (s. f.). Recuperado 11 de mayo de 2026, de https://d1wqtxts1xzle7.cloudfront.net/81203573/GUIA_CULTIVO_PLATANO_2011-libre.pdf?1645506468=&response-content-disposition=inline%3B+filename%3DGUIA_CULTIVO_PLATANO.pdf&Expires=1778555347&Signature=JiErtJ-3SrbbPsIxzH37rvw344yOQxVwVE9zyWbfR6WzFmx3F4ApegVXauVIzmDZErX1OauPGgvNNH1UiJ6ASuEJNkPgZ-

Hw~25McKteZdrPvhuR9lyLwc3VX1lF1dskGrZfhMBE93HQRD8TGqLA6tVSDbx
PInyA0jd6619pNUh0jMSusurOx7dcIkG6TI-
eZUR1~WOx8936HQ7WJj6yXOxDBOImRJNb1eguYJBd4c3aOE2uQmPNrOKFm
Jk4CNjfXRPyxoXt7SOs-0Hwhx93tdmErdKd6JP6W4ky5Fxpr9youeZl9PVOMy-
PJz26lpiAD4uPWCNXluy-o3t5orOCA__&Key-Pair-
Id=APKAJLOHF5GGSLRBV4ZA

- Hidalgo-Larrea, J., Bermúdez, M. J. V., Vera, M. del P. A., Bravo, L., & Miño, J. (2026). Marco arquitectónico para sistemas de predicción aplicados a la producción del banano en Ecuador. *Ecuadorian Science Journal*, 10(1), 7-14. <https://doi.org/10.46480/esj.10.1.280>
- Kepner, C. D., & Soothill, J. H. (2022). *El imperio del banano: Las compañías bananeras contra la soberanía de las naciones del caribe*. Ediciones AKAL.
- Laso Bayas, J. C., Gardeazabal, A., Karner, M., Folberth, C., Vargas, L., Skalský, R., Balkovič, J., Subash, A., Saad, M., Delerce, S., Crespo Cuaresma, J., Hlouskova, J., Molina-Maturano, J., See, L., Fritz, S., Obersteiner, M., & Govaerts, B. (2020). AgroTutor: A Mobile Phone Application Supporting Sustainable Agricultural Intensification. *Sustainability*, 12(22), 9309. <https://doi.org/10.3390/su12229309>
- Majumdar, J., Naraseeyappa, S., & Ankalaki, S. (2017). Analysis of agriculture data using data mining techniques: Application of big data. *Journal of Big Data*, 4(1), 20. <https://doi.org/10.1186/s40537-017-0077-4>
- Munhoz, T., Vargas, J., Teixeira, L., Staver, C., & Dita, M. (2024). Fusarium Tropical Race 4 in Latin America and the Caribbean: Status and global research advances towards disease management. *Frontiers in Plant Science*, 15. <https://doi.org/10.3389/fpls.2024.1397617>

- Phaltankar, A., Ahsan, J., Harrison, M., Nedov, L., & Anderson, S. (2020). *MongoDB Fundamentals: A hands-on guide to using MongoDB and Atlas in the real world*. Packt Publishing Ltd.
- Quijije, M. J. L., Loor, L. M. S., Lopez, C. S. V., Centeno, M. E. H., Guzmán, J. A. M., García, J. L. R., Hidrovo, C. A. M., & Ramos, J. R. O. (2025). Efecto sobre el desarrollo agronómico del plátano Dominico Hartón sometido a condiciones de sequía e inundación: Agricultura y Silvicultura. *La Técnica. Revista de las Agrociencias*. ISSN 2477-8982, 15(2), 134-145. <https://doi.org/10.33936/latecnica.v15i2.7567>
- Röckel, F., Schreiber, T., Schüler, D., Braun, U., Krukenberg, I., Schwander, F., Peil, A., Brandt, C., Willner, E., Gransow, D., Scholz, U., Kecke, S., Maul, E., Lange, M., & Töpfer, R. (2022). PhenoApp: A mobile tool for plant phenotyping to record field and greenhouse observations. *F1000Research*, 11, 12. <https://doi.org/10.12688/f1000research.74239.2>
- Rozentals, N. (2021). *Mastering TypeScript: Build enterprise-ready, modular web applications using TypeScript 4 and modern frameworks*. Packt Publishing Ltd.
- Saiz-Rubio, V., & Rovira-Más, F. (2020). From Smart Farming towards Agriculture 5.0: A Review on Crop Data Management. *Agronomy*, 10(2), 207. <https://doi.org/10.3390/agronomy10020207>
- Sarasa, A. (2021). *Introducción a las bases de datos NSQL clave-valor usando Redis*. Editorial UOC.
- Sarasa, A., & Cabezuelo, A. S. (2021). *Introducción a las bases de datos NoSQL usando MongoDB*. Editorial UOC.
- Tummers, J., Kassahun, A., & Tekinerdogan, B. (2019). Obstacles and features of Farm Management Information Systems: A systematic literature review. *Computers and*

Electronics in Agriculture, 157, 189-204.

<https://doi.org/10.1016/j.compag.2018.12.044>

Vieira, D. (2022). *Designing Hexagonal Architecture with Java: An architect's guide to building maintainable and change-tolerant applications with Java and Quarkus*. Packt Publishing Ltd.

Wu, S. H., & Mueller, T. A. (2024). A user-friendly NoSQL framework for managing agricultural field trial data. *Scientific Reports*, 14(1), 29819. <https://doi.org/10.1038/s41598-024-81609-2>

Zúñiga, P. I. V., Cedeño, R. J. C., & Palacios, I. A. M. (2023). Metodología de la investigación científica: Guía práctica. *Ciencia Latina Revista Científica Multidisciplinar*, 7(4), 9723-9762. https://doi.org/10.37811/cl_rcm.v7i4.7658

ANEXOS

Anexo A: Asignación de tutor

 
Universidad Laica Eloy Alfaro de Manabí
Periodo 2025-2 - Notificación de tutor asignado - SOFTWARE 2024 - NS - (EL CARMEN)
Estimad@ Docente y Estudiante Uleam En cumplimiento de lo establecido en la Ley, el Reglamento de Régimen Académico y las disposiciones estatutarias de la Uleam, por medio de la presente se oficializa la dirección y tutoría en el desarrollo del Trabajo de Integración curricular / Trabajo de Titulación del siguiente estudiante: Tema: SISTEMA INFORMÁTICO CON ESQUEMAS POLIMÓRFICOS PARA EL MODELADO DE DATOS FITOTÉCNICOS EN LA GRANJA EXPERIMENTAL ULEAM EL CARMEN Estado de aprobación: Aprobado Tipo de titulación: Trabajo de Integración Curricular Tipo de proyecto: Trabajo de Integración Curricular / Trabajo de titulación se articula con proyectos y programas de Investigación. Apellidos y nombres del tutor asignado: LOPEZ RODRIGUEZ CARLOS VINICIO Apellidos y nombres del estudiante: ZAMORA LOOR JOSE MANUEL Carrera: SOFTWARE 2024 - NS - (EL CARMEN) Periodo de inducción: Periodo 2025-2

Anexo B: Reporte del sistema antiplagio


Certificado de análisis
 Compilatio Magister+ | ULEAM-ECU

ZAMORA LOOR JOSE comp
 ID : fd1cd4b13f7b3a822eb089b351fcb47000a34

9%
 Textos sospechosos

Nombre del fichero : ZAMORA LOOR JOSE comp.txt
 Tamaño del archivo original : 4.48 MB
 Número de palabras : 26.527

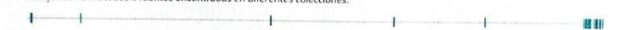
Depositante : CARLOS VINICIO LOPEZ RODRIGUEZ
 Fecha de depósito : 31 de julio de 2026
 Tipo de carga : interface
 Fecha de fin de análisis : 31 de julio de 2026

Resumen (sección 1/2)

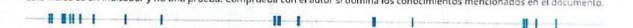
Localización de los textos sospechosos en el documento :
 

Incluido en el porcentaje de textos sospechosos :

1%

Similitudes
 Pasajes con similitudes a fuentes encontradas en diferentes colecciones.
 

6%

Detección de IA
 Textos estilísticamente próximos a un texto generado por una IA.
 Este índice es un indicador y no una prueba. Comprueba con el autor si domina los conocimientos mencionados en el documento.
 

2%

Idiomas no reconocidos
 Pasajes en los que parte del vocabulario utilizado no forma parte del diccionario de la lengua.
 Puede tratarse de un intento del autor de modificar el texto para evitar ser detectado.
 

No incluido en el porcentaje de textos sospechosos :

<1%

Textos entre comillas
 Pasajes entre comillas, a menudo indicativos de una cita.
 

Fuentes de similitudes (sección 2/2)

1%

Similitudes
 Pasajes con similitudes a fuentes encontradas en diferentes colecciones.
 

Fuente principal detectada

Nº	Descripciones	Similitudes	Ubicaciones
1	DESARROLLO DE UN SISTEMA DE MONITORIO... Viene de mi grupo	<1%	
2	Effect of planting density and complementary irrigation on the... doi.org/10.17268/ci.agropecu.2020.04.03	<1%	
3	Estrategia y Plan de Acción Regional para la Preparación... www.fao.org/3/cb674es/cb674es.pdf	<1%	
4	Effect of plant density on growth and yield in Barrangete plantai... doi.org/10.15446/acag.v6n3.52198	<1%	

Fuente mencionada (sin similitudes detectadas)

Nº	Descripciones
1	https://gc.scalahed.com/recursos/files/r161r/w26022w/Arias_S2.pdf
2	https://doi.org/10.1016/j.atech.2023.100214
3	https://doi.org/10.35977/0104-1096.cct2025.v42.27757
4	https://doi.org/10.1093/igascience/giw019
5	http://repositorio.iniaip.gob.ec/handle/41000/5825
6	https://doi.org/10.1109/ICCUBE.2017.8463693
7	https://doi.org/10.46480/tesj.10.1.280
8	https://doi.org/10.3390/su12229309
9	https://doi.org/10.33936/latecnica.v15i2.7567
10	https://doi.org/10.3390/agronomy10020207
11	https://doi.org/10.1016/j.compag.2018.12.044
12	https://doi.org/10.37811/rl_rcm.v7i4.7658



Anexo C: Fotografías

Ilustración 30: Validación del inicio de sesión en Postman, con respuesta exitosa del backend y generación del token JWT.

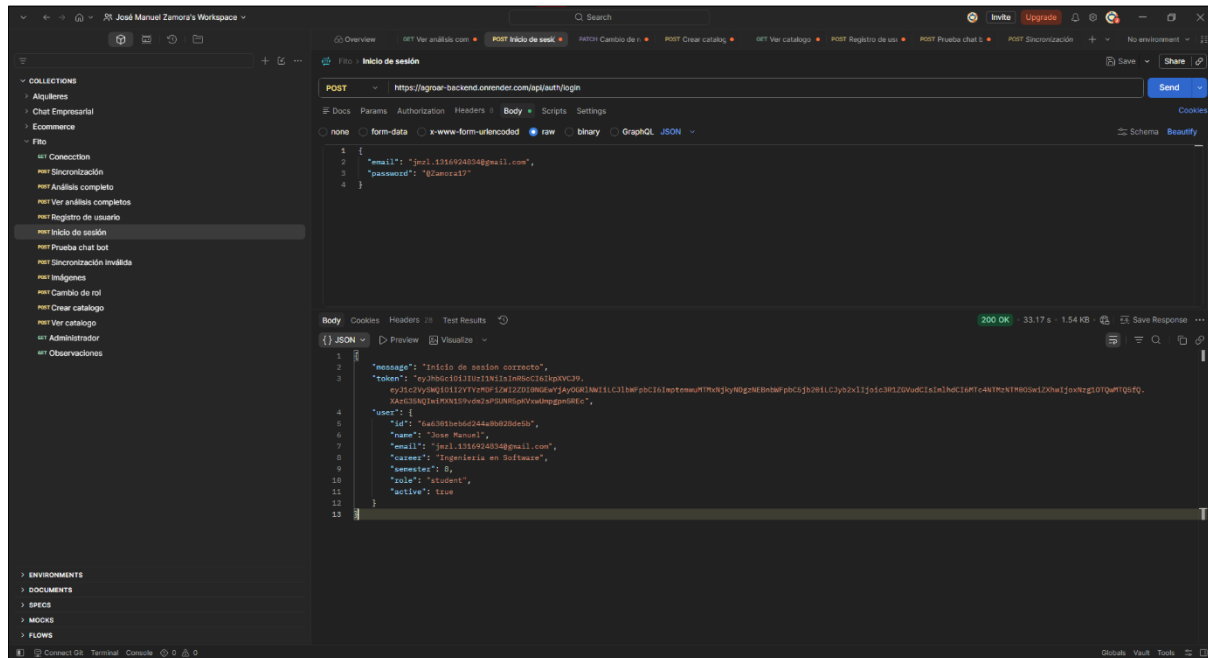


Ilustración 31: Pantalla de inicio de sesión del prototipo AgroAR Plátano ejecutado en entorno de pruebas.

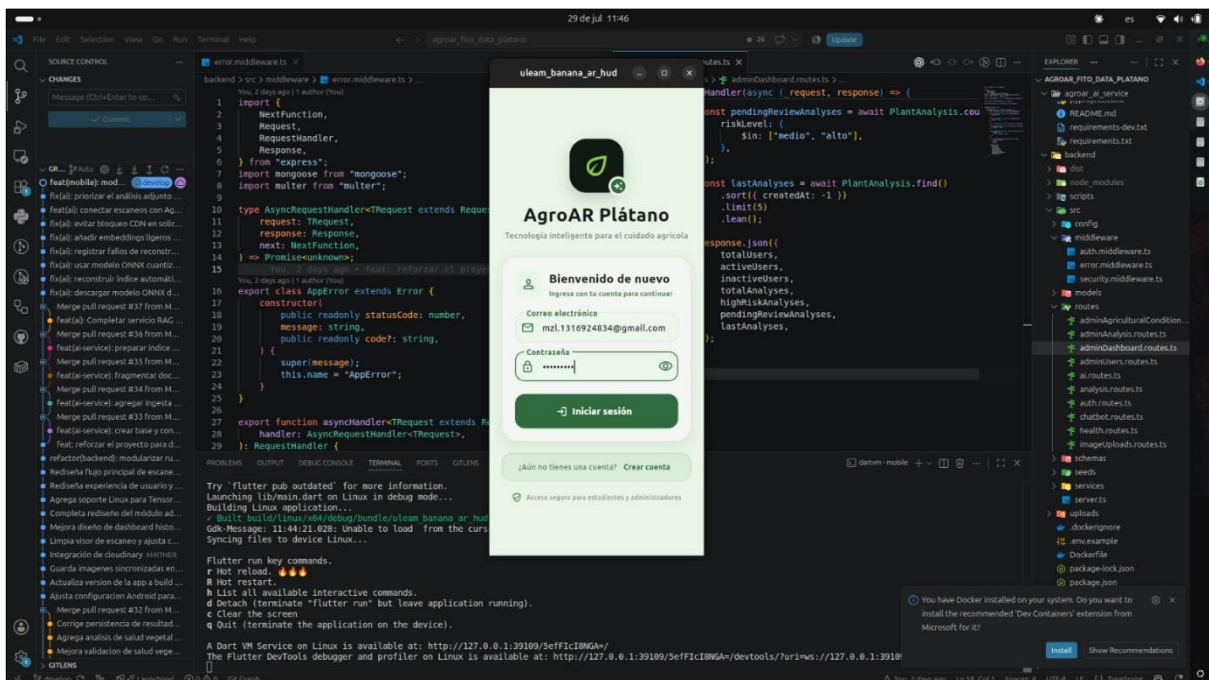


Ilustración 32: Panel principal del usuario estudiante con resumen de revisiones y acceso al análisis de plantas.

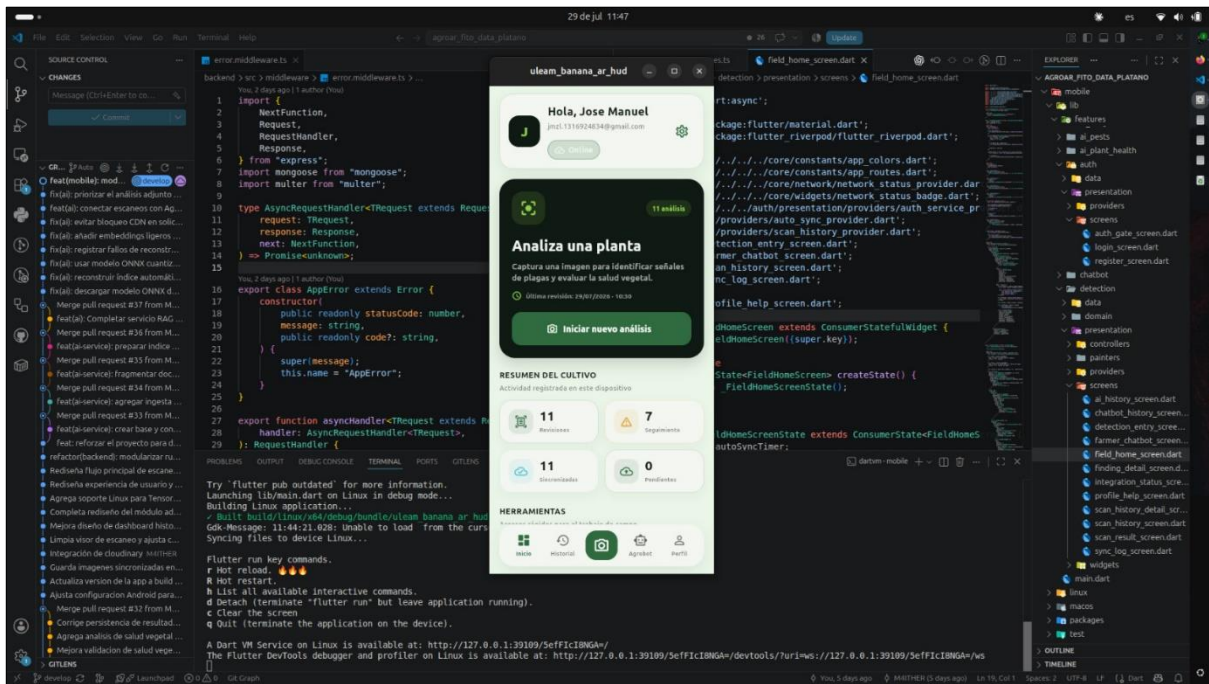


Ilustración 33: Historial de revisiones sincronizadas filtrada por usuario entre la aplicación móvil y MongoDB Atlas.

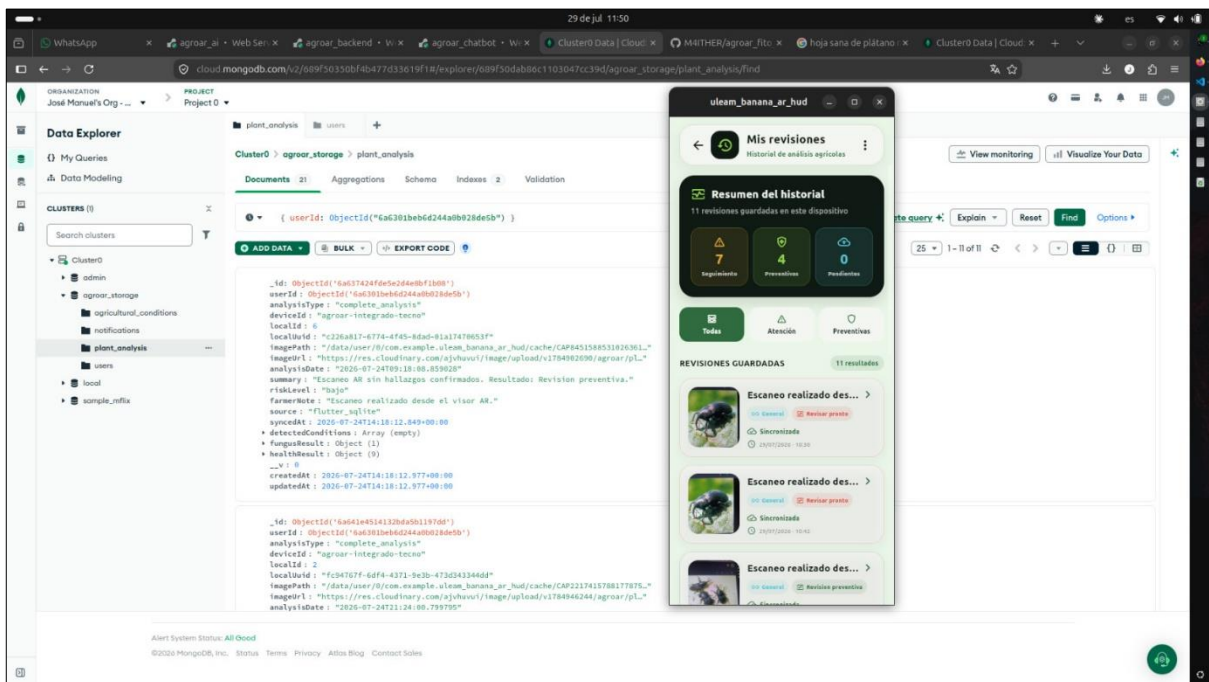


Ilustración 34: Panel administrativo para la gestión de usuarios, análisis recibidos y resumen general del sistema.

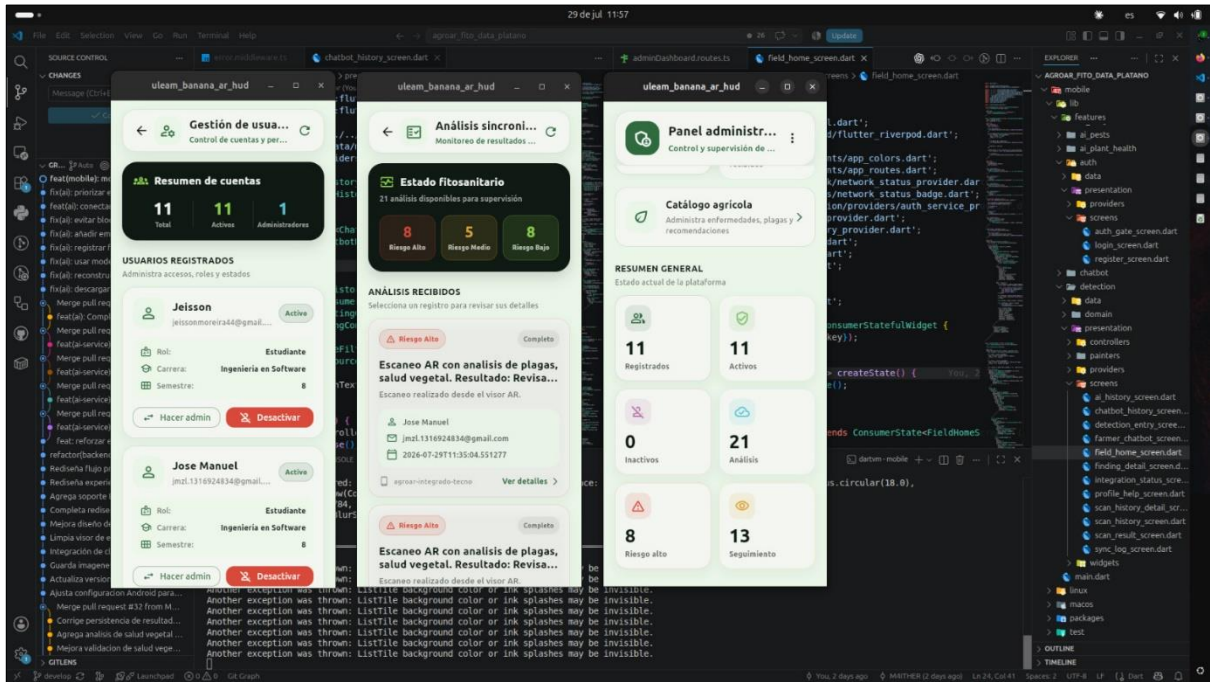
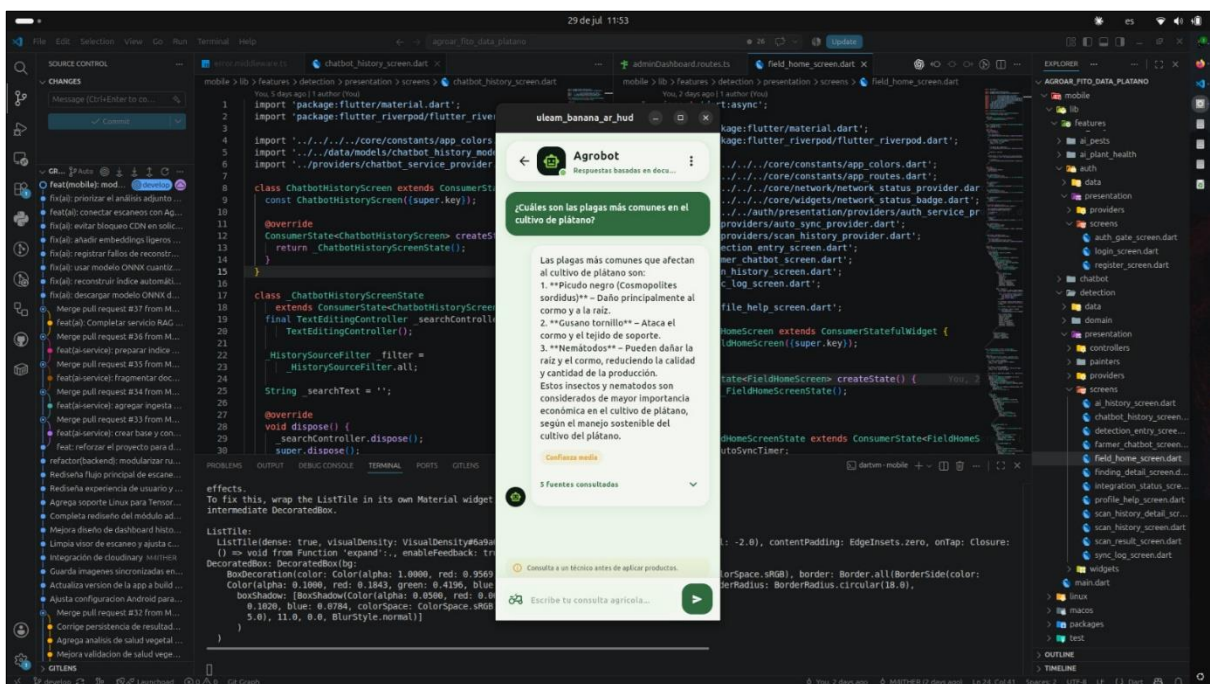


Ilustración 35: Respuesta del chatbot agrícola sobre plagas comunes en el cultivo de plátano.



Anexo D: Evidencia de entrevista y encuestas

Ilustración 36: Evidencia de la entrevista semiestructurada aplicada al docente especialista de la Granja Experimental ULEAM El Carmen.



Ilustración 37: Resultados de la encuesta sobre la aceptación de tecnologías de agricultura de precisión en la Granja Experimental ULEAM El Carmen.

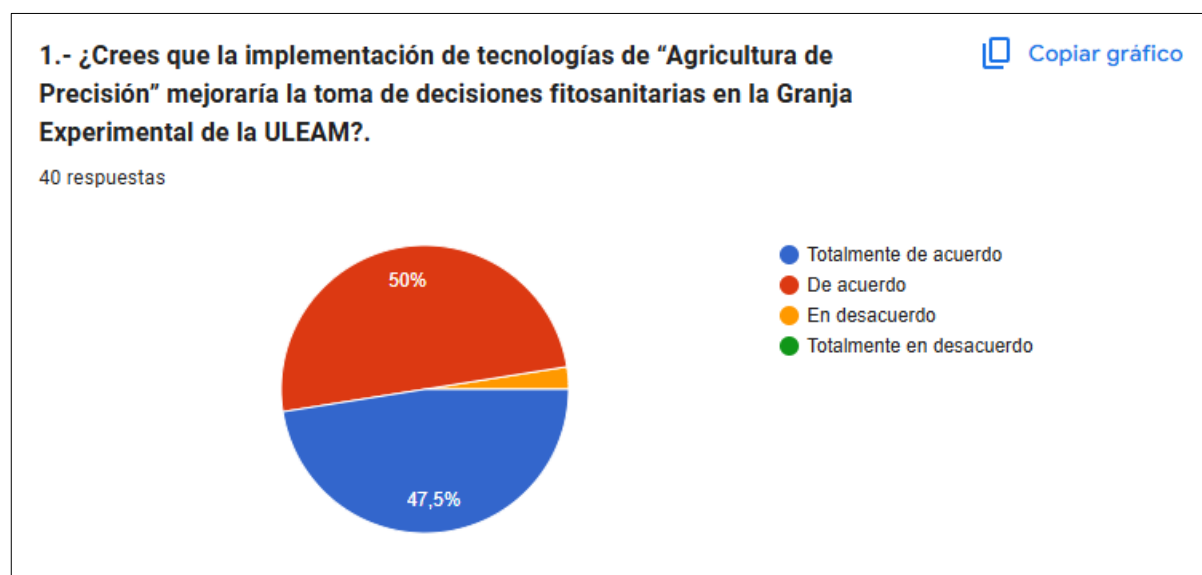


Ilustración 38: Resultados de la encuesta sobre los principales obstáculos tecnológicos para el uso de una aplicación agrícola en El Carmen.

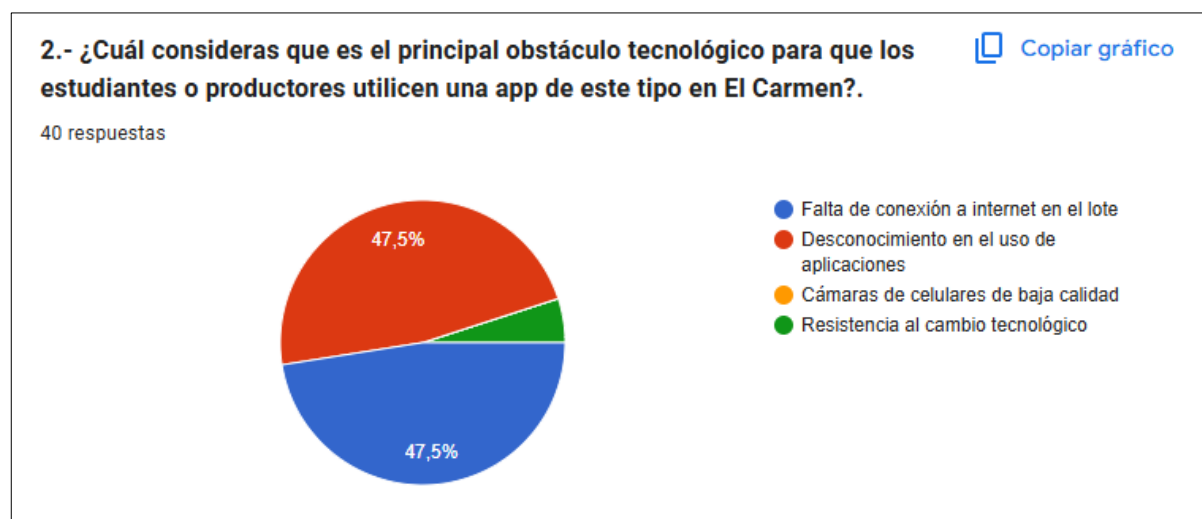


Ilustración 39: Resultados de la encuesta sobre la aceptación de recomendaciones generadas por un sistema tecnológico a partir del análisis de imágenes.

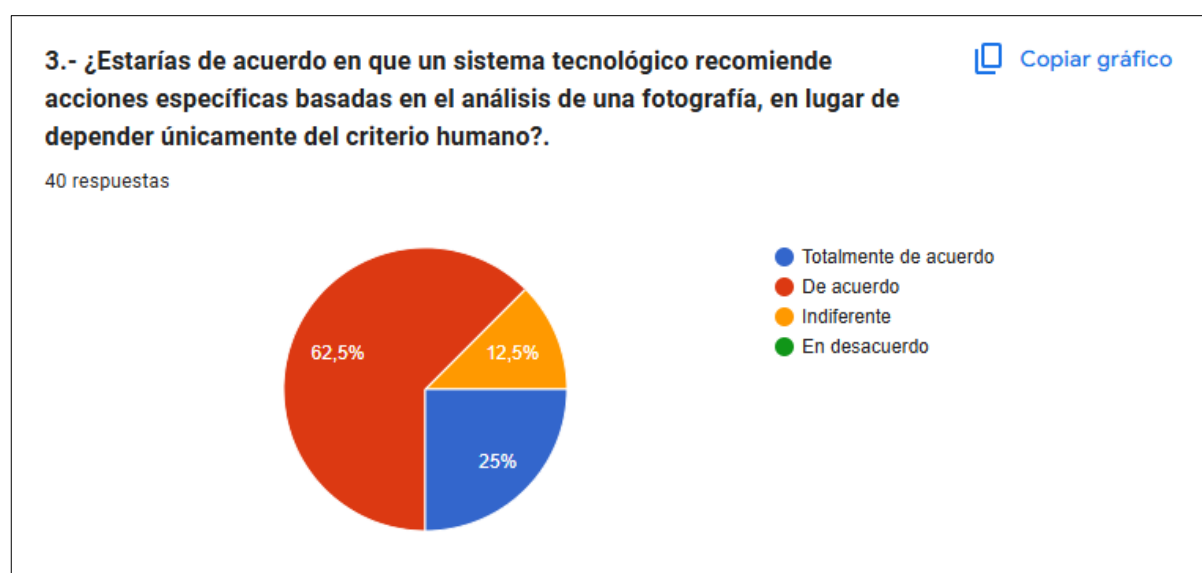
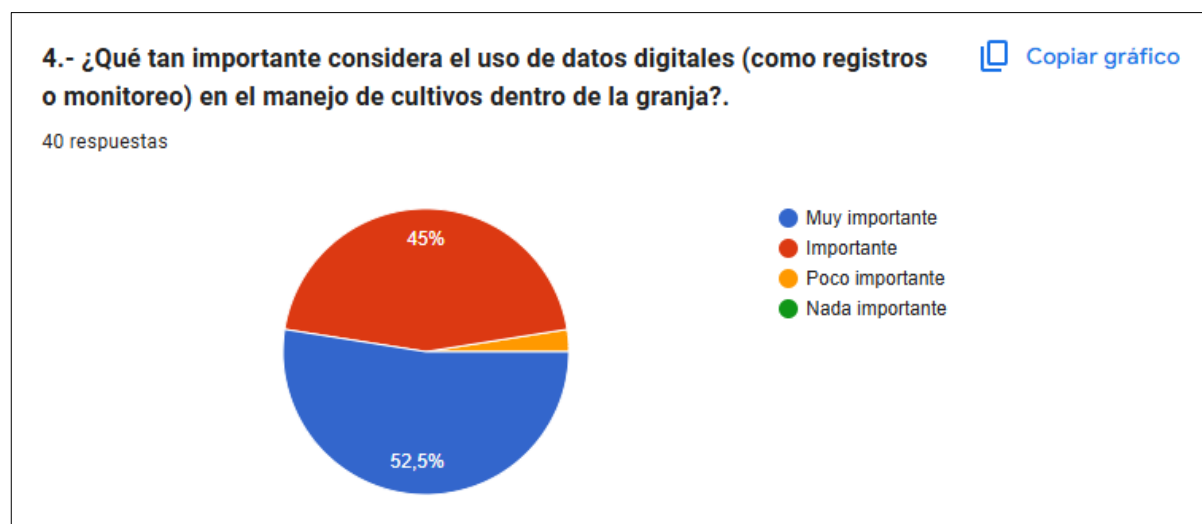


Ilustración 40: Resultados de la encuesta sobre la importancia del uso de datos digitales en el manejo de cultivos dentro de la granja.



GLOSARIO

AgroAR Plátano: Prototipo de aplicación móvil desarrollado para apoyar el registro, análisis y seguimiento de revisiones agrícolas en el cultivo de plátano.

Almacenamiento local: Proceso mediante el cual la aplicación guarda información directamente en el dispositivo móvil, permitiendo conservar datos, aunque no exista conexión a internet.

Backend: Parte del sistema encargada de recibir, validar, procesar y enviar información entre la aplicación móvil y la base de datos.

Base de datos NoSQL: Modelo de almacenamiento flexible que permite guardar información con estructuras variables, sin depender de tablas rígidas como en las bases de datos relacionales.

Cloudinary: Servicio en la nube utilizado para almacenar imágenes asociadas a los análisis agrícolas realizados desde la aplicación.

Datos fitotécnicos: Información técnica relacionada con el desarrollo, estado y seguimiento de los cultivos, como observaciones, imágenes, riesgos, plagas o condiciones de salud vegetal.

Esquemas polimórficos: Estructuras de datos flexibles que permiten guardar registros con campos diferentes dentro de una misma colección, según el tipo de análisis realizado.

Flutter: Framework utilizado para desarrollar la aplicación móvil del sistema AgroAR Plátano.

Historial de revisiones: Módulo de la aplicación donde se almacenan y consultan los análisis agrícolas realizados por el usuario.

Inteligencia artificial: Tecnología utilizada como apoyo para analizar imágenes del cultivo y generar resultados relacionados con plagas o salud vegetal.

JWT: Token de seguridad generado por el backend para validar la sesión del usuario después del inicio de sesión.

MongoDB Atlas: Base de datos NoSQL en la nube utilizada para almacenar la información sincronizada desde la aplicación móvil.

Node.js: Entorno utilizado para desarrollar el backend del sistema y gestionar las peticiones enviadas desde la aplicación móvil.

Offline-first: Enfoque de diseño que permite que la aplicación funcione sin conexión a internet, guardando la información localmente y sincronizándola cuando la red esté disponible.

Reporte PDF: Documento generado por el sistema con la información de una revisión agrícola, incluyendo datos del análisis, observaciones e imagen de evidencia.

Render: Plataforma utilizada para desplegar el backend y permitir que la aplicación móvil se conecte al servidor desde internet.

SQLite: Base de datos local utilizada en el dispositivo móvil para almacenar revisiones cuando no existe conexión a internet.

Sincronización: Proceso mediante el cual los datos guardados localmente en SQLite se envían al backend y luego se almacenan en MongoDB Atlas.

Usuario administrador: Tipo de usuario encargado de supervisar información general del sistema, como usuarios registrados, análisis recibidos y datos administrativos.

Usuario estudiante: Tipo de usuario que utiliza la aplicación para realizar análisis de plantas, guardar revisiones, consultar historial y generar reportes.