



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 CONVERSACIONAL INTELIGENTE DE
PARÁMETROS PRODUCTIVOS DEL CULTIVO DE MUSA PARADISIACA EN LA
GRANJA EXPERIMENTAL ULEAM EL CARMEN

AUTOR:

VERA ANDRADE ANGELO ANDRÉS

TUTOR:

ING. CARLOS VINICIO LÓPEZ RODRÍGUEZ

AÑO:

2026

CERTIFICACIÓN DEL TUTOR

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

CERTIFICACIÓN

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

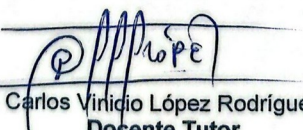
Haber dirigido, revisado y aprobado preliminarmente el Trabajo de Integración Curricular bajo la autoría del estudiante **ANGELO ANDRÉS VERA ANDRADE**, legalmente matriculado/a en la carrera de Tecnologías de la Información período académico 2026-1, cumpliendo el total de 288 horas, cuyo tema del proyecto o núcleo problémico es **"SISTEMA INFORMÁTICO CONVERSACIONAL INTELIGENTE DE PARÁMETROS PRODUCTIVOS DE MUSA PARADISIACA 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 Viridilio López Rodríguez, Mg
Docente Tutor
Área: Tecnologías de la Información

TRIBUNAL DE SUSTENTACIÓN



Universidad Laica Eloy Alfaro de Manabí

Extensión El Carmen

Carrera de Ingeniería en Software

TRIBUNAL DE SUSTENTACIÓN

Título del Trabajo de Titulación:

**SISTEMA INFORMÁTICO CONVERSACIONAL INTELIGENTE DE
PARÁMETROS PRODUCTIVOS DEL CULTIVO DE MUSA PARADISIACA EN LA
GRANJA EXPERIMENTAL ULEAM EL CARMEN.**

Modalidad:

Proyecto Integrador

Autor:

Vera Andrade Angelo Andrés

Tutor:

Ing. Carlos Vinicio López Rodríguez, Mg.

Tribunal de Sustentación:

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

Fecha de Sustentación:

26 de agosto del 2026.

UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ

EXTENSIÓN EL CARMEN



DECLARACIÓN DE AUTORÍA

La responsabilidad del contenido de este Trabajo de Titulación, cuyo tema es "SISTEMA INFORMÁTICO CONVERSACIONAL INTELIGENTE DE PARÁMETROS PRODUCTIVOS DEL CULTIVO DE MUSA PARADISIACA EN LA GRANJA EXPERIMENTAL ULEAM EL CARMEN", corresponde exclusivamente a Vera Andrade Angelo Andrés con Cédula de Identidad N° 1317437505, y los derechos patrimoniales de la misma corresponden a la Universidad Laica "Eloy Alfaro" de Manabí.

Vera Andrade Angelo Andrés

C.I. 1317437505

DEDICATORIA

Dedico este trabajo, ante todo, a Dios, por darme la fortaleza y la sabiduría necesarias para culminar esta etapa universitaria.

A mi madre y a mi padre, por su amor incondicional, por el esfuerzo constante que hicieron posible mi formación y por confiar siempre en mis capacidades, incluso en los momentos más difíciles del camino.

A toda mi familia y amigos, por su apoyo, su paciencia y sus palabras de aliento en cada etapa de este proceso. Este logro no es solo mío, sino también de quienes me acompañaron y creyeron en mí desde el inicio.

Vera Andrade Angelo Andrés

AGRADECIMIENTOS

Agradezco a Dios por darme salud, perseverancia y claridad para completar este Trabajo de Titulación. A mis padres y a mi familia, por su respaldo constante y por ser el motor que impulsó cada esfuerzo realizado durante esta investigación. Agradezco de manera especial a mi tutor, Ing. Carlos Vinicio López Rodríguez, Mg., por su guía, su exigencia académica y su disposición permanente para orientar el desarrollo de este proyecto. Extiendo mi gratitud a la Universidad Laica “Eloy Alfaro” de Manabí, Extensión El Carmen, y a la Carrera de Ingeniería en Software, por la formación académica recibida a lo largo de estos años. Asimismo, agradezco al personal técnico y a los estudiantes de la Granja Experimental ULEAM El Carmen por su colaboración durante el levantamiento de información y la validación del sistema desarrollado, y a mis compañeros y amigos, por su apoyo a lo largo de esta etapa.

Vera Andrade Angelo Andrés

ÍNDICE GENERAL

CERTIFICACIÓN DEL TUTOR.....	II
------------------------------	----

TRIBUNAL DE SUSTENTACIÓN.....	III
DECLARACIÓN DE AUTORÍA.....	IV
DEDICATORIA.....	V
AGRADECIMIENTOS.....	VI
ÍNDICE DE TABLAS.....	XI
ÍNDICE DE GRÁFICOS E ILUSTRACIONES.....	XII
ÍNDICE DE ANEXOS.....	XIII
RESUMEN.....	XIV
ABSTRACT.....	XV
1 INTRODUCCIÓN.....	1
1.1 Introducción.....	1
1.2 Presentación del tema.....	1
1.3 Contextualización de la problemática.....	2
1.4 Planteamiento del problema.....	2
1.4.1 Problematización.....	2
1.4.2 Génesis del problema.....	3
1.4.3 Estado actual del problema.....	3
1.5 Diagrama causa-efecto.....	4
1.6 Objetivos.....	4
1.6.1 Objetivo general.....	4
1.6.2 Objetivos específicos.....	5
1.7 Justificación.....	5
1.8 Impactos esperados.....	6
1.8.1 Impacto tecnológico.....	6
1.8.2 Impacto social.....	6
1.8.3 Impacto ecológico.....	6
2 MARCO TEÓRICO.....	7

2.1	Antecedentes históricos.....	7
2.1.1	Evolución de los sistemas conversacionales e inteligencia artificial.....	7
2.1.2	La inteligencia artificial aplicada al sector agrícola.....	7
2.2	Investigaciones relacionadas.....	8
2.3	Definiciones conceptuales y categorías clave.....	9
2.3.1	Inteligencia artificial conversacional.....	9
2.3.2	Arquitectura de sistemas conversacionales.....	10
2.3.3	Procesamiento de lenguaje natural (PLN).....	10
2.3.4	Comprensión y generación del lenguaje natural.....	11
2.3.5	Tecnologías de reconocimiento y síntesis de voz.....	11
2.3.6	Modelos de lenguaje de gran escala (LLM).....	12
2.3.7	Diseño conversacional e interfaces multimodales.....	12
2.3.8	Frameworks y plataformas de desarrollo para chatbots.....	13
2.3.9	Tecnología del desarrollo del sistema.....	13
2.3.10	Generalidades del cultivo de Musa paradisiaca.....	14
2.3.11	Requerimientos edafoclimáticos del sector.....	14
2.3.12	Fenología y fases de desarrollo biológico.....	15
2.3.13	Parámetros biométricos y variables fitotécnicas.....	15
2.3.14	Control fitosanitario y principales enfermedades.....	16
2.3.15	Granjas experimentales y su rol en la investigación agrícola.....	16
2.3.16	Metodología de desarrollo ágil (Scrum).....	17
2.3.17	Lenguaje de programación Python.....	17
2.3.18	Base de datos para sistemas conversacionales.....	17
2.4	Conclusiones del marco teórico.....	18
3	MARCO INVESTIGATIVO.....	19
3.1	Introducción.....	19
3.2	Tipos de investigación.....	19

3.2.1	Investigación explicativa.....	19
3.2.2	Investigación de desarrollo.....	20
3.2.3	Investigación mixta.....	20
3.3	Métodos de investigación.....	20
3.3.1	Método inductivo-deductivo.....	20
3.3.2	Método analítico y experimental.....	21
3.4	Fuentes de información.....	21
3.4.1	Fuentes primarias.....	21
3.4.2	Fuentes secundarias.....	21
3.5	Estrategia de recolección de datos.....	22
3.5.1	Población.....	22
3.5.2	Muestra.....	22
3.5.3	Técnicas e instrumentos de recolección.....	22
3.6	Análisis e interpretación de resultados.....	23
3.6.1	Resultados de encuesta a estudiantes:.....	24
3.6.2	Resultado de la entrevista.....	27
3.6.3	Informe final del análisis de datos.....	29
4	MARCO PROPOSITIVO.....	31
4.1	Introducción.....	31
4.2	Descripción de la propuesta.....	31
4.3	Determinación de recursos.....	32
4.3.1	Recursos humanos.....	32
4.3.2	Recursos tecnológicos.....	32
4.3.3	Recursos económicos.....	33
4.4	Desarrollo según metodología Scrum.....	34
4.4.1	Descripción del producto.....	34
4.4.2	Condiciones de éxito del producto.....	35

4.4.3	Historias de usuario.....	36
4.4.4	Diseño del sistema (enfoque lógico y backend).....	39
4.4.5	Descripción técnica y arquitectura del sistema.....	42
4.4.6	Roles y responsabilidades del equipo Scrum.....	45
4.4.7	Planificación de Sprints.....	46
4.4.8	Backlog del producto.....	51
4.4.9	Interfaces de integración.....	53
4.4.10	Definición de Hecho (DoD).....	54
4.4.11	Eventos Scrum.....	54
4.4.12	Proceso de pruebas.....	57
5	EVALUACIÓN DE RESULTADOS.....	60
5.1	Introducción.....	60
5.2	Presentación y monitoreo de resultados.....	60
5.2.1	Planificación de la evaluación.....	60
5.2.2	Ejecución del monitoreo.....	60
5.3	Interpretación objetiva.....	62
5.4	Evaluación de aceptación y usabilidad (pruebas piloto con usuarios).....	62
6	CONCLUSIONES Y RECOMENDACIONES.....	64
6.1	Conclusiones.....	64
6.2	Recomendaciones.....	64
7	BIBLIOGRAFÍA.....	66
8	ANEXOS.....	70
9	GLOSARIO.....	83

ÍNDICE DE TABLAS

Tabla 1. Resultados de la encuesta aplicada a los estudiantes de la Granja Experimental ULEAM El Carmen.....	25
Tabla 2. Matriz de interpretación de la entrevista al docente experto de la Granja Experimental	29
Tabla 3. Recursos Humanos.....	34
Tabla 4. Recursos Tecnológicos.....	35
Tabla 5. Presupuesto del proyecto.....	35
Tabla 6. HU-01 – Procesamiento de consulta en lenguaje natural (RAG).....	38
Tabla 7. HU-02 – Actualización y vectorización de la Base de Conocimiento.....	39
Tabla 8. HU-03 – Evaluación de certeza y derivación.....	40
Tabla 9. Caso de Uso 1: Procesamiento RAG y Generación de Respuesta.....	42
Tabla 10. Caso de Uso 002 – Indexación y Vectorización de Documentos.....	43
Tabla 11. Caso de Uso 003 – Gestión de Caché y Consultas Factuales.....	44
Tabla 12. Requerimientos Funcionales del Servidor e IA.....	46
Tabla 13. Requerimientos No Funcionales del Servidor e IA.....	47
Tabla 14. Roles y Responsabilidades del Equipo Scrum.....	47
Tabla 15. Resumen de Planificación de Sprints.....	48
Tabla 16. Planificación Sprint 1.....	49
Tabla 17. Planificación Sprint 2.....	50
Tabla 18. Planificación Sprint 3.....	51
Tabla 19. Planificación Sprint 4.....	53
Tabla 20. Backlog Inicial del Producto (Backend e IA).....	54
Tabla 21. Endpoints Principales del Sistema (Backend).....	56
Tabla 22. Sprint Review 1 – Entorno y Base de Conocimiento.....	57
Tabla 23. Sprint Review 2 – Desarrollo del Backend y API.....	58
Tabla 24. Sprint Review 3 – Inteligencia Artificial y Pipeline RAG.....	58
Tabla 25. Sprint Review 4 – Optimización y Pruebas Finales.....	59
Tabla 26. Pruebas de Caja Negra – Endpoints de la API.....	60
Tabla 27. Pruebas de Caja Blanca – Lógica Python y Modelos.....	61
Tabla 28. Resultados de latencia del servidor.....	64
Tabla 29. Resultados de clasificación de intenciones.....	64
Tabla 30. Fiabilidad técnica y seguridad agronómica.....	65

ÍNDICE DE GRÁFICOS E ILUSTRACIONES

Ilustración 1. Diagrama causa-efecto del problema de investigación.....	4
Ilustración 2. Diagrama de Casos de Uso: Procesamiento RAG y Generación.....	41
Ilustración 3. Diagrama de Casos de Uso: Procesamiento RAG y generación de respuesta...	76
Ilustración 4. Diagrama de Casos de Uso: Gestión de caché y consultas factuales.....	77
Ilustración 5. Diagrama de Casos de Uso: Actualización de la base de conocimiento.....	77
Ilustración 6. Evidencia de ejecución del modelo Llama 3.1 en servidor local (Ollama).....	78
Ilustración 7. Código de inicialización de la base de datos vectorial ChromaDB.....	79
Ilustración 8. Estructura de carpetas evidenciando la persistencia local de ChromaDB.....	79
Ilustración 9. Terminal de ejecución confirmando la carga de manuales y activación del pipeline.....	80
Ilustración 10. Interfaz de documentación automática Swagger UI de la API REST.....	81
Ilustración 11. Repositorio en GitHub con el código fuente del proyecto.....	82
Ilustración 12. Interfaz web del chatbot interactuando en entorno de escritorio.....	83
Ilustración 13. Historial de conversaciones y visualización de respuestas del sistema.....	84
Ilustración 14. Interfaz web del chatbot adaptada a la vista de dispositivos móviles.....	85
Ilustración 15. Visualización del menú lateral en dispositivos móviles.....	85
Ilustración 16. Entrevistando al docente.....	86
Ilustración 17. Estudiantes respondiendo las encuestas.....	86

ÍNDICE DE ANEXOS

Anexo A. Asignación de tutor.....	73
Anexo B. Reporte del sistema antiplagio.....	73
Anexo C. Arquitectura Lógica del Sistema.....	75
Anexo D. Evidencia de Ejecución del Modelo de Inteligencia Artificial (Ollama).....	78
Anexo E. Base de Datos Vectorial (ChromaDB) y Generación Aumentada por Recuperación (RAG).....	79
Anexo F. Documentación de la API REST (FastAPI).....	81
Anexo G. Repositorio de Código Fuente.....	81
Anexo H. Evidencia del chatbot funcionando.....	82
Anexo I. Fotos realizando las encuestas y entrevista.....	86

RESUMEN

En la Granja Experimental ULEAM El Carmen, los 40 estudiantes y el personal técnico vinculados al cultivo de *Musa paradisiaca* enfrentan limitaciones para acceder de forma oportuna a información técnica actualizada, ya que los talleres presenciales y los manuales impresos no ofrecen disponibilidad continua ni respuestas inmediatas ante dudas agronómicas urgentes. Frente a esta problemática, el presente trabajo, titulado "Sistema Informático Conversacional Inteligente de parámetros productivos de *Musa paradisiaca* en la Granja Experimental ULEAM El Carmen", desarrolla la arquitectura lógica y el motor de inteligencia artificial de un asistente virtual pensado para superar la lentitud de los métodos tradicionales de capacitación agrícola. El trabajo se ejecutó con un enfoque mixto y bajo la metodología ágil Scrum, distribuido en cuatro sprints. La población estuvo conformada por los 40 estudiantes de la Granja Experimental y un docente especializado en el cultivo, con muestreo censal y no probabilístico intencional respectivamente, mediante las técnicas de encuesta y entrevista. Como base del sistema se adoptó una arquitectura orientada a servicios con una API REST construida en FastAPI que integra el modelo Llama 3.1 ejecutado localmente mediante Ollama. Asimismo, se incorporó un pipeline de Generación Aumentada por Recuperación en Python respaldada por MongoDB, ChromaDB y SQLite, de modo que recupera fragmentos agronómicos verificados y evita así respuestas inventadas.

En las pruebas el clasificador de intenciones alcanzó 91 % de precisión. Además, el 97 % de los usuarios consideró útil la interacción por voz y las consultas complejas se resolvieron en 5.1 segundos. Por consiguiente, estos resultados confirman que el sistema mejora la autonomía técnica del productor incorpora alertas de bioseguridad para prácticas más seguras y moderniza la forma en que la Granja Experimental gestiona su conocimiento.

Palabras clave: Inteligencia Artificial, Procesamiento de Lenguaje Natural, RAG, Ollama, *Musa paradisiaca*, Backend.

ABSTRACT

At the ULEAM El Carmen Experimental Farm, the 40 students and technical staff involved in the *Musa paradisiaca* crop face limitations in accessing up-to-date technical information in a timely manner, since in-person workshops and printed manuals do not provide continuous availability or immediate answers to urgent agronomic questions. In response to this problem, this degree project, "Intelligent Conversational Information System for productive parameters of *Musa paradisiaca* at the ULEAM El Carmen Experimental Farm", develops the logical architecture and AI engine of a virtual assistant that optimizes the transfer of technical agricultural knowledge, overcoming the lack of immediacy of traditional training methods. Methodologically, the project was executed under a mixed approach and the Agile Scrum framework, in four sprints. The population consisted of the 40 students of the Experimental Farm and one specialized crop-management instructor, selected through census and purposive non-probability sampling respectively, using survey and interview techniques. Technological innovation significantly boosts the autonomy of agricultural producers. The system is built on a service-oriented architecture featuring a REST API developed in FastAPI, which integrates the Llama 3.1 model running locally via Ollama. Additionally, a Python-based Retrieval-Augmented Generation (RAG) pipeline supported by MongoDB, ChromaDB and SQLite was implemented to retrieve verified agronomic information, thereby preventing the generation of fabricated responses.

Testing showed that the intent classifier achieved 91% accuracy. Furthermore, 97% of users found the voice interaction useful, and complex queries were resolved in 5.1 seconds. Consequently, these results confirm that the system enhances the producer's technical autonomy, incorporates biosecurity alerts to promote safer practices, and modernizes the Experimental Farm's knowledge management.

Keywords: Artificial Intelligence, Natural Language Processing, RAG, Ollama, *Musa paradisiaca*, Backend.

CAPÍTULO I

1 INTRODUCCIÓN

1.1 Introducción

Los sistemas informáticos conversacionales, conocidos también como chatbots o asistentes virtuales, ganan relevancia porque permiten interactuar con los usuarios mediante lenguaje natural, ofreciendo una comunicación más accesible, rápida y eficiente que facilita el acceso a información especializada sin exigir conocimientos técnicos avanzados por parte de quien consulta.

Los talleres presenciales y los manuales impresos, que son la forma tradicional de capacitación agrícola, no ofrecen disponibilidad continua ni respuestas inmediatas. Los sistemas conversacionales sí: se adaptan a lo que el usuario necesita en el momento y responden al instante, algo clave en el campo, donde decisiones como el control de plagas, la fertilización o el manejo del cultivo no pueden esperar.

El plátano, nombre común de la *Musa paradisiaca*, es uno de los cultivos más importantes del Ecuador por su volumen de producción y por la demanda que genera tanto dentro como fuera del país. El manejo exitoso del cultivo exige sin embargo conocimientos técnicos precisos en cada etapa del proceso productivo. Asimismo, se requieren saberes especializados en fertilización control fitosanitario y manejo hídrico entre otros aspectos fundamentales para lograr resultados óptimos.

La producción de plátano impulsa el desarrollo económico del cantón El Carmen en la provincia de Manabí. A pesar de esa relevancia, los estudiantes técnicos y productores vinculados a la Granja Experimental de la ULEAM no siempre acceden a tiempo a información técnica actualizada por lo que se complica la correcta aplicación de las prácticas agronómicas.

Ante ese panorama, un sistema informático conversacional inteligente aparece como una alternativa concreta: agiliza la transferencia de conocimiento técnico, apoya la toma de decisiones y, con eso, fortalece la productividad del cultivo de *Musa paradisiaca* en la Granja Experimental ULEAM.

1.2 Presentación del tema

La propuesta de desarrollar un sistema conversacional inteligente nace de la necesidad de modernizar la forma en que se comparte el conocimiento agronómico, ya que la capacitación

sigue dependiendo de métodos tradicionales que muchas veces impiden que las recomendaciones técnicas lleguen a tiempo. El sistema funcionará como una plataforma especializada que usa un motor de Inteligencia Artificial Conversacional junto con Procesamiento de Lenguaje Natural (PLN) para brindar asesoría eficiente y clara sobre los parámetros productivos del cultivo de Musa paradisiaca, centrado en la Granja Experimental de la Universidad Laica Eloy Alfaro de Manabí (ULEAM), cantón El Carmen.

La idea es que el sistema actúe como un asistente virtual experto al que cualquiera pueda consultar desde su celular, funcionando tanto conectado a internet como en zonas con poca señal, de manera que sirva realmente en el campo, que es donde se toman las decisiones críticas, y con una interfaz multimodal que acepte consultas tanto por texto como por voz.

1.3 Contextualización de la problemática

La investigación se desarrolla en la Granja Experimental de la ULEAM, ubicada en el cantón El Carmen, provincia de Manabí, zona reconocida por su alta producción de plátano, que constituye un punto estratégico para el desarrollo de estudios agrícolas y es conocida popularmente como “La Puerta de Oro de Manabí”.

La digitalización agrícola multiplica a escala mundial el acceso a información técnica y biológica para el manejo de cultivos. Sin embargo, en países en desarrollo como Ecuador los canales de transmisión de ese conocimiento hacia productores y estudiantes aún presentan serias brechas. Los métodos convencionales de capacitación, como las visitas técnicas periódicas, los manuales impresos y los talleres presenciales, no logran satisfacer la demanda de información en tiempo real que exige el manejo eficiente de cultivos de alta variabilidad como la Musa paradisiaca.

La Granja Experimental cumple un doble papel, académico y productivo, ya que funciona como espacio de práctica para los estudiantes de Ingeniería en Software e Ingeniería Agropecuaria. Eso la convierte en un lugar apropiado para probar soluciones tecnológicas nuevas. Al mismo tiempo, la falta de herramientas digitales especializadas en ese entorno deja abierta una oportunidad clara de intervención, con impacto tanto académico como productivo.

1.4 Planteamiento del problema

1.4.1 Problematización

La manera en que se gestiona y se transmite el conocimiento técnico en las granjas experimentales universitarias deja ver la falta de sistemas especializados. Los métodos tradicionales de capacitación resultan lentos y poco adaptados a cada necesidad. Asimismo, los

talleres presenciales y las guías impresas generan baja eficiencia porque cuando el estudiante o el agricultor requiere la información esta suele estar desactualizada o simplemente no disponible. Por consiguiente, esa desconexión entre el conocimiento disponible y su aplicación real en el campo termina frenando la producción.

De esta necesidad surge la pregunta principal de esta investigación: ¿De qué manera la implementación de un sistema informático conversacional inteligente puede mejorar la eficiencia de la capacitación agrícola y fomentar la adopción de buenas prácticas en el cultivo de Musa paradisíaca en la Granja Experimental ULEAM? La falta de herramientas digitales modernas ha creado barreras que dificultan el acceso a información vital para investigar, aprender y producir de forma sostenible, y un sistema conversacional resulta la mejor opción por ofrecer una interfaz amigable que rompe la barrera del miedo a la tecnología.

1.4.2 Génesis del problema

El origen de este problema está en la distancia histórica entre cómo se genera el conocimiento agronómico y cómo se aplica en la práctica. Por condiciones estructurales y tecnológicas que afectan tanto a la Granja Experimental de la ULEAM como a otros entornos rurales similares, los agricultores han terminado confiando más en su experiencia, en lo que se comenta en la zona o en consultas ocasionales a algún experto. Mientras los centros de investigación siguen generando y actualizando datos técnicos, los canales para difundirlos no han avanzado al mismo ritmo, y se sigue dependiendo de formatos manuales o digitales que no están conectados entre sí.

1.4.3 Estado actual del problema

Actualmente, en la Granja Experimental ULEAM el soporte técnico y la capacitación se realizan sobre todo con métodos convencionales, los cuales presentan varias limitaciones claras:

Capacitación esporádica y teórica: la transmisión de conocimientos depende de cuándo se realizan talleres o visitas técnicas, lo cual deja un vacío de soporte continuo, ya que no existe ayuda en tiempo real para resolver los problemas cotidianos del cultivo.

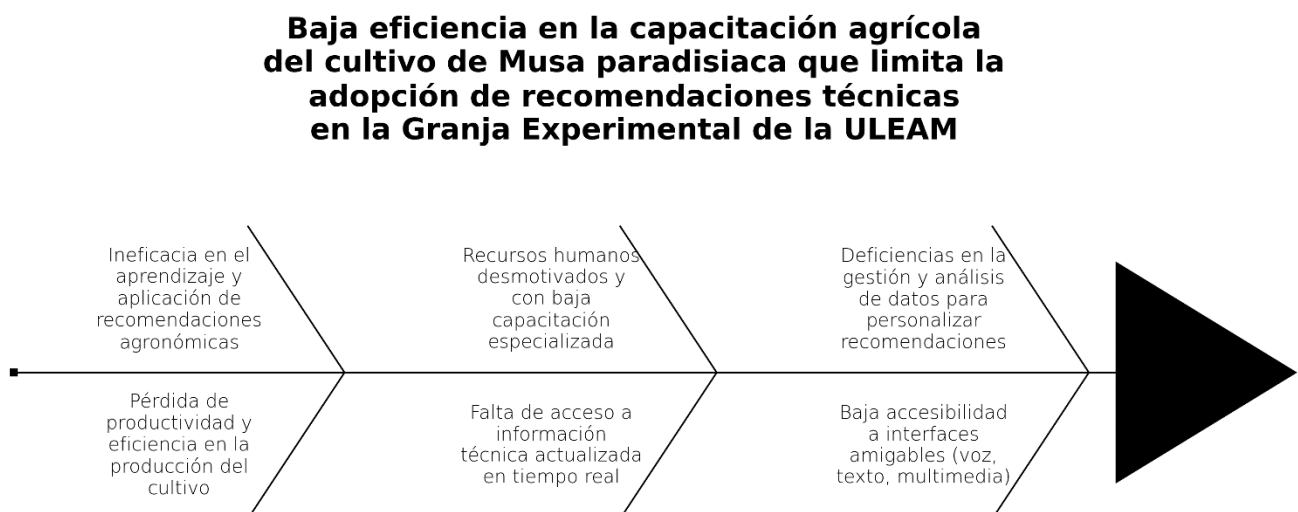
Agricultores y estudiantes carecen de un canal inmediato para aclarar dudas sobre plagas enfermedades o fertilización. Por lo tanto, esa falta retrasa las decisiones y con el tiempo puede generar pérdidas de producción.

El sector agropecuario carece todavía de herramientas digitales diseñadas realmente para el usuario final. Por otra parte, faltan sistemas de consulta por voz o texto simple que se adapten a personas con distintos niveles de conocimiento técnico y digital.

Recomendaciones no personalizadas: la asesoría suele ser muy general, ya que rara vez considera las condiciones específicas de una parcela, la etapa del cultivo o su historial de manejo, lo que reduce la efectividad de las recomendaciones.

1.5 Diagrama causa-efecto

Ilustración 1. Diagrama causa-efecto del problema de investigación



1.6 Objetivos

1.6.1 Objetivo general

Desarrollar un sistema informático conversacional inteligente que mejore la gestión de los parámetros productivos del cultivo de Musa paradisiaca en la Granja Experimental ULEAM El Carmen.

1.6.2 Objetivos específicos

Analizar las tecnologías y los fundamentos teóricos de la inteligencia artificial conversacional mediante la revisión de investigaciones previas sobre chatbots aplicados al sector agrícola, con el fin de sentar las bases teóricas y tecnológicas del sistema propuesto.

Diagnosticar los requerimientos técnicos y funcionales del entorno productivo de la Granja Experimental ULEAM El Carmen mediante un estudio de campo, con el fin de diseñar una arquitectura conversacional que garantice respuestas técnicas precisas y contextualizadas.

Desarrollar un prototipo funcional del sistema informático conversacional inteligente, para lo cual se configuró el motor de Procesamiento de Lenguaje Natural junto con las interfaces de usuario necesarias para consultar información técnica productiva.

Evaluar la eficiencia y aceptación del sistema mediante pruebas piloto con usuarios finales, estudiantes y personal docente de la Granja Experimental, con el fin de validar su funcionalidad en un entorno agrícola real.

1.7 Justificación

El proyecto moderniza la capacitación agrícola mediante tecnologías accesibles y eficientes. Asimismo, un sistema conversacional inteligente facilita el acceso al conocimiento técnico mejora la toma de decisiones agronómicas y optimiza el manejo del cultivo de Musa paradisiaca en la Granja Experimental ULEAM atendiendo así a necesidades reales del sector agrícola de El Carmen.

Desde el ámbito académico esta investigación aporta datos empíricos sobre el uso de inteligencia artificial en agronomía un campo todavía poco documentado en el contexto ecuatoriano. Asimismo, el diseño conversacional empleado como método de capacitación agrícola constituye un aporte metodológico propio porque muestra cómo estructurar conocimiento técnico complejo para transmitirlo mediante una interacción natural con una máquina.

El proyecto fortalece además el desarrollo académico y tecnológico de la institución. Por una parte promueve el uso de inteligencia artificial en contextos reales del sector agrícola y por otra forma profesionales de ingeniería de software capaces de resolver problemas concretos de su entorno.

1.8 Impactos esperados

1.8.1 Impacto tecnológico

El proyecto alcanza como aporte central un prototipo funcional validable en un entorno real. Se trata de una herramienta de inteligencia artificial conversacional que supera las limitaciones de disponibilidad y actualización propias de los métodos tradicionales de capacitación. Para la Granja Experimental ULEAM esto representa una innovación tecnológica relevante puesto que introduce herramientas modernas en la gestión del conocimiento agrícola y además abre camino a futuras tecnologías de precisión.

1.8.2 Impacto social

El impacto social esperado se centra en los usuarios diarios del sistema. Se prevé que estudiantes, docentes y personal técnico de la Granja Experimental puedan disponer de una herramienta de consulta disponible en cualquier momento. De forma indirecta, también podría beneficiar a los productores de plátano de El Carmen y de Manabí en la medida en que accedan a información técnica validada por la universidad. En consecuencia, se espera que la mejora del flujo del conocimiento impulse la calidad de las prácticas agronómicas, la productividad del sector y la sostenibilidad del desarrollo local.

1.8.3 Impacto ecológico

El proyecto genera un efecto ambiental esperado, aunque se centre en el desarrollo de software. Al ofrecer recomendaciones más precisas sobre el manejo del cultivo, especialmente en control de plagas y uso de insumos, podría reducirse el riesgo de aplicar productos de forma inadecuada o excesiva. Por consiguiente, se espera que el sistema favorezca un uso más responsable de los recursos, podría contribuir a disminuir la contaminación del suelo y empujaría hacia prácticas agrícolas más sostenibles en la región de El Carmen. Cabe precisar que este estudio no midió directamente niveles de contaminación del suelo, por lo que este impacto se plantea como una expectativa razonable y no como un resultado comprobado.

CAPÍTULO II

2 MARCO TEÓRICO

2.1 Antecedentes históricos

2.1.1 Evolución de los sistemas conversacionales e inteligencia artificial

La búsqueda de que las máquinas se comuniquen con los humanos mediante lenguaje natural se origina en la década de 1960. Uno de los primeros avances fue ELIZA, desarrollado por Joseph Weizenbaum en el MIT, que simulaba conversaciones reconociendo patrones de palabras clave y emitiendo respuestas predefinidas. Aunque no comprendía el significado real de los mensajes, evidenció la capacidad de las personas para interactuar con programas que imitaban el diálogo humano, sentando las bases de la investigación en PLN (Jurafsky & Martin, 2024).

Durante las décadas de 1970 y 1980, los sistemas conversacionales evolucionaron hacia modelos basados en reglas y representaciones simbólicas del conocimiento, apoyados en árboles de decisión y sistemas expertos que exigían que los usuarios se adaptaran a formatos de entrada rígidos, por lo que resultaban poco flexibles ante la variabilidad natural del lenguaje humano y esto limitó su adopción masiva (Jurafsky & Martin, 2024).

La verdadera inflexión se produjo a partir de 2010, gracias a la confluencia de tres factores: la disponibilidad masiva de datos textuales (Big Data), el incremento de la capacidad de procesamiento mediante GPU, y el desarrollo de arquitecturas de aprendizaje profundo como los Transformers (Vaswani et al., 2017). Estos avances permitieron el surgimiento de los Modelos de Lenguaje de Gran Escala (LLM), capaces de generar respuestas coherentes, mantener el contexto conversacional y comprender matices del lenguaje con precisión sin precedentes.

Actualmente, los sistemas conversacionales inteligentes forman parte de múltiples aplicaciones en sectores como la educación, la salud y la agricultura, facilitando el acceso inmediato a información especializada, y su nivel de madurez tecnológica los posiciona como herramientas viables para transferir conocimiento técnico en entornos de alta complejidad, como el manejo de cultivos tropicales (Sommerville, 2011).

2.1.2 La inteligencia artificial aplicada al sector agrícola

La incorporación de la inteligencia artificial al sector agrícola ha transformado radicalmente la gestión de la información productiva: desde los primeros sistemas de soporte a

la decisión basados en reglas lógicas simples, hasta los sistemas de aprendizaje automático capaces de procesar imágenes satelitales y datos sensoriales en tiempo real. Hoy, la agricultura de precisión demanda sistemas capaces de integrar múltiples fuentes de información y comunicarlas a los usuarios de forma clara y accesible (FAO, 2024).

Los sistemas conversacionales abren una vía esencial de acceso al conocimiento agronómico. A diferencia de las plataformas tradicionales que obligan a navegar manualmente por menús y bases de datos, los chatbots permiten preguntar en lenguaje cotidiano y recibir una respuesta personalizada en tiempo real. En un entorno rural como El Carmen, donde el nivel de alfabetización digital es desigual, esa inmediatez puede ser justamente lo que separa una intervención fitosanitaria a tiempo de la pérdida de una cosecha.

Diversos estudios recientes confirman que los sistemas conversacionales han mejorado la interacción entre usuarios y plataformas tecnológicas, porque acercan la información técnica a través del lenguaje natural e integran fuentes de datos en tiempo real para apoyar la toma de decisiones en los cultivos (Dwivedi, 2023).

2.2 Investigaciones relacionadas

En el ámbito de la inteligencia artificial conversacional aplicada al agro, diversas investigaciones han validado la efectividad de estos sistemas para transferir conocimiento técnico, por lo que a continuación se analizan dos antecedentes que sustentan y contextualizan el presente proyecto de titulación.

En Ecuador, Morocho-Cayamcela, González-Palacio y Luzuriaga-Quichimbo (2022) desarrollaron un sistema de lógica difusa para diagnosticar enfermedades foliares en *Musa paradisiaca*. Era técnicamente sólido, pero su interfaz tradicional limitaba la usabilidad para productores con poca experiencia digital, una barrera de acceso evidente.

Este antecedente confirma que los sistemas conversacionales basados en PLN son viables técnicamente y bien aceptados por usuarios del sector agrícola latinoamericano. Con todo, su alcance se limitó a un solo cultivo y no consideró ni la multimodalidad de acceso ni la operación sin conexión. Ahí es donde el sistema propuesto para la Granja Experimental ULEAM El Carmen va más allá: al combinar texto y voz con capacidad de funcionar en zonas de baja conectividad, ofrece una solución más robusta y más ajustada a las condiciones reales del campo ecuatoriano.

Un segundo antecedente relevante es Farmer.Chat, un asistente conversacional basado en inteligencia artificial diseñado para escalar servicios agrícolas a pequeños productores, el cual demuestra que los sistemas conversacionales facilitan el acceso a recomendaciones

técnicas personalizadas y mejoran la adopción tecnológica cuando el lenguaje y el canal de interacción se ajustan al contexto del usuario (Singh et al., 2024; Sharma et al., 2023). El sistema conversacional desarrollado para la Granja Experimental ULEAM El Carmen retoma ese mismo principio de accesibilidad conversacional y lo aplica al dominio específico de Musa paradisiaca, de modo que la interacción resulta intuitiva sin necesidad de conocer términos técnicos de navegación.

Combinar los datos captados en el campo con el conocimiento experto codificado dentro del sistema conversacional produce, en la práctica, un valor superior al de las soluciones aisladas. Esa premisa orienta la arquitectura híbrida que este proyecto aplica al cultivo de plátano.

Al llevar esas conclusiones al entorno de la Granja Experimental ULEAM se confirma la pertinencia tecnológica de la propuesta, pues los sistemas conversacionales rinden mejor cuando integran una base de conocimiento experto junto con datos contextuales actualizados. Por esa razón el asistente virtual no se limitará a recuperar información agronómica estática, sino que enriquecerá cada respuesta con los parámetros productivos registrados en la base de datos de la propia granja, generando así recomendaciones verdaderamente personalizadas.

2.3 Definiciones conceptuales y categorías clave

2.3.1 Inteligencia artificial conversacional

La Inteligencia Artificial Conversacional une el Procesamiento del Lenguaje Natural con algoritmos de aprendizaje automático para crear sistemas capaces de mantener diálogos coherentes con las personas. A diferencia de los sistemas de respuesta automatizada tradicionales, estos comprenden el contexto de la conversación, retienen la memoria del intercambio y generan respuestas originales a partir del conocimiento almacenado, ya que no se limitan a responder automáticamente, sino que interpretan el contexto del intercambio y generan respuestas adaptadas a la necesidad específica del usuario, capacidad de adaptación contextual que diferencia fundamentalmente a un chatbot basado en inteligencia artificial de un sistema de respuesta automática por menús (Dwivedi, 2023).

Aplicado al cultivo de Musa paradisiaca, un sistema conversacional inteligente podrá interpretar preguntas como “¿qué le pasa a mi planta si las hojas se ponen amarillas?” y ofrecer un diagnóstico diferencial que considere la fase fenológica del cultivo, las condiciones climáticas del sector y las enfermedades más comunes de la zona, sin que el usuario necesite conocer la terminología técnica exacta para formular la consulta.

2.3.2 Arquitectura de sistemas conversacionales

Un sistema conversacional moderno no es una pieza tecnológica única, sino un ecosistema de módulos interconectados que trabajan de manera sincronizada para garantizar una experiencia de usuario óptima, y su arquitectura típica se compone de los siguientes elementos fundamentales (Sommerville, 2011):

Interfaz de usuario (UI): es el punto de contacto entre el sistema y el usuario, ya sea mediante una pantalla de chat, una aplicación móvil o un canal de voz (McTear, 2020), y en el caso de la Granja Experimental ULEAM deberá ser accesible desde dispositivos móviles de gama media, dadas las condiciones tecnológicas del entorno agrícola.

El motor de Inteligencia Artificial constituye el núcleo del sistema (Russell & Norvig, 2020). Procesa lo que recibe del usuario y analiza cada consulta mediante modelos de PLN, por lo que su precisión determina en gran parte la calidad de las respuestas generadas.

El diseño conversacional establece la lógica del diálogo. Define cómo responde el sistema ante distintos tipos de preguntas cómo maneja las ambigüedades y cómo guía al usuario hasta la información que busca.

La integración con bases de datos permite al sistema consultar información técnica actualizada sobre el cultivo de Musa paradisiaca almacenada en repositorios estructurados. De este modo, se asegura la precisión agronómica de cada respuesta.

El módulo de análisis y mejora continua evalúa el rendimiento del sistema a partir de métricas de satisfacción del usuario, precisión al identificar intenciones y tiempo de respuesta; con esos datos se pueden hacer ajustes periódicos, siguiendo el mismo principio de mejora continua basada en retroalimentación que documentan Ouyang et al. (2022).

Todo este proceso debe ejecutarse en milisegundos para que la experiencia del usuario sea fluida y natural, de modo que, desde la interfaz donde el estudiante escribe o habla, pasando por el motor que entiende la pregunta, hasta la conexión con la base de conocimiento agronómico que devuelve la respuesta técnica, cada componente debe estar optimizado para garantizar una latencia mínima.

2.3.3 Procesamiento de lenguaje natural (PLN)

El Procesamiento de Lenguaje Natural es el pilar tecnológico que permite a las máquinas interpretar y generar lenguaje humano, ya que combina lingüística computacional con modelos estadísticos y de aprendizaje profundo para que los sistemas informáticos procesen, analicen y generen texto o voz de manera similar a como lo hace una persona (Sommerville, 2011).

En un sistema conversacional agrícola, el PLN se enfrenta a un desafío particular: los usuarios describen problemas fitosanitarios o condiciones del cultivo con terminología local, regionalismos y expresiones coloquiales. El sistema debe interpretar con precisión expresiones cotidianas como "la planta se está secando por arriba" o "las hojas tienen manchas negras" y vincularlas con condiciones agronómicas específicas tales como el estrés hídrico o la Sigatoka Negra.

Las arquitecturas Transformer como BERT o GPT elevan el rendimiento de las técnicas actuales de PLN por encima de los antiguos modelos limitados al reconocimiento de palabras clave. Gracias a ello logran captar el contexto sintáctico y semántico completo de cada consulta. En consecuencia, el sistema entrega respuestas acertadas incluso si el usuario no emplea un lenguaje técnico preciso, factor decisivo para que la herramienta sea adoptada en los entornos agrícolas rurales. Estas arquitecturas basadas en mecanismos de atención han elevado significativamente el estado del arte en comprensión y generación de lenguaje natural (Devlin et al., 2019).

2.3.4 Comprensión y generación del lenguaje natural

Desde el punto de vista funcional, el PLN se organiza en dos procesos complementarios que actúan juntos para interpretar la interacción del usuario:

La Comprensión del Lenguaje Natural o NLU constituye el proceso de entrada encargado de transformar el texto o la voz del usuario en datos estructurados y manejables. Este proceso abarca la tokenización, que separa el texto en unidades con sentido; el Reconocimiento de Entidades Nombradas o NER, que localiza conceptos como Sigatoka Negra o fertilización foliar; el análisis de intención, que clasifica la consulta como diagnóstico, recomendación o información general; y el análisis de sentimiento que identifica urgencia o preocupación en el tono del usuario (Sommerville, 2011).

La Generación del Lenguaje Natural (NLG) es el proceso complementario: convierte los datos estructurados o el conocimiento recuperado de las bases de datos en respuestas coherentes y comprensibles. En el sistema propuesto, esto significa traducir información técnica agronómica a un lenguaje accesible, ajustando el nivel técnico según el perfil de quien pregunta.

2.3.5 Tecnologías de reconocimiento y síntesis de voz

Para garantizar la accesibilidad del sistema en contextos rurales, donde los usuarios pueden tener limitaciones de lectura o las manos ocupadas durante el trabajo de campo, el

sistema incorporará capacidades de interacción por voz, tecnologías determinantes para que productores con distintos niveles de alfabetización digital adopten la herramienta (FAO, 2022).

El Reconocimiento Automático de Voz (ASR) convierte señales de audio en texto procesable (Radford et al., 2022). En aplicaciones agrícolas debe enfrentar desafíos como el ruido ambiental, las variaciones dialectales regionales y el vocabulario técnico especializado, y debe operar con alta precisión incluso con acentos regionales del Ecuador.

La Síntesis de Texto a Voz (TTS) vocaliza las respuestas del chatbot, convirtiendo el texto generado por el sistema en audio comprensible, y como la selección de voces naturales y el ajuste de la velocidad de locución son críticos para usuarios con menor nivel de lectura, la tecnología actual permite generar respuestas de voz que despiertan mayor empatía y confianza en el usuario rural (McTear, 2020).

2.3.6 Modelos de lenguaje de gran escala (LLM)

Sin duda uno de los avances más determinantes que ha alcanzado la inteligencia artificial conversacional durante la última década corresponde a los Modelos de Lenguaje de Gran Escala (LLM), ya que se trata de sistemas de aprendizaje profundo (Goodfellow et al., 2016) entrenados con corpus de texto descomunales (Brown et al., 2020), capaces de comprender y generar lenguaje natural con una coherencia que hasta hace poco resultaba impensable; incluso modelos como GPT-4, LLaMA (Touvron et al., 2023) o Claude han terminado demostrando capacidades avanzadas de razonamiento, síntesis y generación de contenido especializado.

Dentro del sistema conversacional pensado para la Granja Experimental ULEAM estos modelos asumen dos funciones al mismo tiempo, pues por un lado entienden las consultas mediante NLU mientras que por otro generan respuestas ajustadas al contexto mediante NLG, de manera que la técnica de Generación Aumentada por Recuperación conocida como RAG (Lewis et al., 2020; Gao et al., 2024) termina uniendo la capacidad generativa de los LLM con una base de conocimiento agronómico específica y así consigue que las respuestas sobre Musa paradisiaca resulten técnicamente precisas en lugar de simples generalidades.

Este enfoque adquiere una relevancia particular dentro del dominio agrícola puesto que una recomendación equivocada, ya sea de fertilización o de control fitosanitario, termina acarreando consecuencias económicas y ecológicas bastante serias.

2.3.7 Diseño conversacional e interfaces multimodales

El diseño conversacional es la disciplina que define cómo debe fluir el diálogo entre el usuario y el sistema para que la experiencia resulte natural, eficiente y satisfactoria, de manera

que un buen diseño no solo estructura las respuestas, sino que también anticipa los flujos de consulta más frecuentes, define cómo maneja el sistema las preguntas ambiguas y establece mecanismos de clarificación cuando la intención del usuario no queda suficientemente clara.

Son varios los escenarios de uso que el diseño conversacional debe cubrir dentro de la Granja Experimental ULEAM, entre ellos las consultas sobre identificación de plagas y enfermedades, las recomendaciones de fertilización según la fase fenológica, los protocolos de cosecha y poscosecha, así como la información sobre los requerimientos hídricos del cultivo, de tal forma que cada uno de estos escenarios termina exigiendo un flujo de diálogo distinto que el sistema debe manejar sin perder la coherencia en ningún momento.

Comunicarse con el sistema a través de múltiples canales, como texto, voz, imágenes y eventualmente video. En la interacción humano-computadora, la multimodalidad incrementa la accesibilidad del sistema al eliminar la barrera de la escritura y permitir que usuarios con diferentes habilidades digitales interactúen eficazmente, por lo cual es un requisito indispensable dado el perfil heterogéneo de los usuarios de la granja experimental.

2.3.8 Frameworks y plataformas de desarrollo para chatbots

Para construir el sistema conversacional no es necesario partir desde cero, ya que existen frameworks especializados que proporcionan la infraestructura básica para el desarrollo de chatbots, reduciendo significativamente los tiempos de implementación y mejorando la calidad del producto final.

Mientras que plataformas propietarias como Dialogflow (Google) o Azure Bot Framework (Microsoft) ofrecen herramientas visuales y APIs robustas con modelos de PLN preentrenados de alta precisión, los frameworks de código abierto, entre ellos Rasa NLU, ofrecen en cambio control total sobre los datos y la arquitectura, aspecto fundamental a la hora de manejar información sensible de producción agrícola institucional, de modo que la elección final dependerá tanto de los recursos disponibles como de los requisitos de privacidad propios de la Granja Experimental ULEAM.

2.3.9 Tecnología del desarrollo del sistema

En cuanto al lenguaje de programación y el entorno de ejecución del backend, Python se posiciona como la opción más adecuada para este tipo de proyectos gracias a su amplio ecosistema de bibliotecas especializadas en PLN, como NLTK (Bird et al., 2009), spaCy y Transformers de Hugging Face, y a su adopción generalizada en el desarrollo de sistemas de inteligencia artificial (Jurafsky & Martin, 2024).

2.3.10 Generalidades del cultivo de *Musa paradisiaca*

El cultivo de *Musa paradisiaca*, conocido comúnmente como plátano, es una de las especies herbáceas de mayor envergadura a nivel mundial y, desde la perspectiva botánica y de seguridad alimentaria, se clasifica como un cultivo fundamental para el sustento de poblaciones en zonas tropicales y subtropicales, caracterizado por su alta demanda de nutrientes y su sistema radicular superficial (Belalcázar Carvajal, 1991; FAO, 2022).

Históricamente, la consolidación de este cultivo respondió tanto a factores biológicos como a un profundo proceso de expansión agroindustrial, ya que, como detallan Kepner y Soothill (1935), el desarrollo de las plantaciones de musáceas en América Latina transformó la dinámica agraria, convirtió territorios enteros en ejes centrales de exportación y exigió un control cada vez más riguroso sobre la administración de las plantaciones.

Esta categorización general es el punto de partida del modelo de datos, pues, 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, lo que permite a los investigadores segmentar la información recolectada y evitar mezclarla con la de otras especies cultivadas en la misma región experimental.

2.3.11 Requerimientos edafoclimáticos del sector

El óptimo desarrollo de la *Musa paradisiaca* depende estrictamente de los factores edafoclimáticos del entorno, ya que la planta requiere suelos con excelente drenaje, textura franco-arenosa o franco-arcillosa, un pH entre 5.5 y 7.0, temperaturas de entre 22°C y 30°C, y precipitación bien distribuida a lo largo del año para mantener el vigor foliar (INIAP, 2023; Vivas Cedeño et al., s.f.).

El cantón El Carmen, por estar en la transición entre la Costa y la Amazonía, tiene condiciones climáticas favorables para el cultivo de plátano. Asociada al fenómeno de El Niño, la variabilidad climática genera periodos de estrés hídrico que terminan afectando las variables productivas, tal como lo demuestran Loo Quijije et al. (2025) en un estudio de carácter local, donde señalan que dicho estrés hídrico altera el desarrollo agronómico del plátano e impacta directamente en variables críticas como la altura de la planta y el diámetro del pseudotallo.

Dentro del sistema conversacional los requerimientos edafoclimáticos constituyen un tipo de conocimiento fundamental puesto que el chatbot deberá orientar al usuario sobre las condiciones óptimas del cultivo al mismo tiempo que interpreta descripciones de síntomas vinculados a deficiencias nutricionales o estrés ambiental sin dejar de recomendar acciones de manejo ajustadas a las condiciones propias del sector de El Carmen.

2.3.12 Fenología y fases de desarrollo biológico

La fenología del plátano estudia las distintas fases de desarrollo biológico que atraviesa la planta durante su ciclo vital, dividido técnicamente en tres etapas principales: la fase vegetativa, de crecimiento de hojas y pseudotallo; la fase reproductiva, de emisión de la inflorescencia o bellota; y la fase productiva, de desarrollo y maduración del racimo (FAO, 2022).

El monitoreo riguroso de estas transiciones es indispensable para la planificación agrícola moderna, ya que identificar con precisión el estado fenológico del cultivo permite a los técnicos aplicar intervenciones oportunas y eficaces.

A nivel de ingeniería de software, la gestión de estas etapas justifica plenamente el uso de esquemas polimórficos, de modo que, siguiendo las recomendaciones de Wu y Mueller (2024) para marcos de recolección NoSQL, el backend cambiará dinámicamente el esquema de validación según la fase fenológica seleccionada: si la planta está en fase vegetativa exigirá el número de hojas, y si entra en fase reproductiva solicitará el conteo de manos del racimo.

2.3.13 Parámetros biométricos y variables fitotécnicas

Los parámetros biométricos y las variables fitotécnicas son los indicadores numéricos del estado de salud y vigor de la planta, por lo que, en el manejo agronómico de musáceas, la evaluación integral incluye medir sistemáticamente la circunferencia del pseudotallo a una altura estándar, la tasa de emisión foliar, la longitud y el ancho de las hojas, y el peso del racimo (Tilahun et al., 2022; Loor Quijije et al., 2025).

La estandarización de estas medidas responde a normativas de control y calidad que garantizan un manejo productivo adecuado, y establecer métricas exactas y libres de ambigüedad es un requisito documentado por INIAP (2023) para detectar a tiempo anomalías de crecimiento que puedan indicar la presencia de patógenos o deficiencias nutricionales severas en el área de siembra.

Digitalizar este volumen de variables numéricas en campo plantea un desafío de latencia y almacenamiento, por lo que, aprovechando las ventajas de las jerarquías documentales propias de las bases de datos NoSQL (Kleppmann, 2017), la base de datos de la universidad procesará en tiempo real las entradas biométricas enviadas desde los dispositivos móviles, garantizando que el análisis estadístico de los parámetros fitotécnicos permanezca siempre estructurado.

2.3.14 Control fitosanitario y principales enfermedades

El control fitosanitario abarca las medidas preventivas y curativas destinadas a proteger el cultivo de plagas y enfermedades, y en el caso de las musáceas, patologías de alto riesgo como la marchitez por *Fusarium* o la Sigatoka Negra representan amenazas biológicas críticas capaces de diezmar plantaciones enteras si no se monitorean bajo estrictos protocolos de bioseguridad (Zhang et al., 2024; FAO, 2022).

El impacto de estas patologías trasciende el ámbito biológico y afecta directamente la estabilidad socioeconómica de las regiones productoras, por lo que la vulnerabilidad fitosanitaria exige una constante modernización de las prácticas para proteger la inversión agrícola y sostener los niveles de producción a largo plazo.

En este escenario, la digitalización del estado de salud de la planta resulta innegociable, por lo que, con base 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 así la respuesta técnica ante cualquier brote infeccioso en la granja.

2.3.15 Granjas experimentales y su rol en la investigación agrícola

Las granjas experimentales universitarias son ecosistemas de innovación donde convergen la investigación, la docencia y la producción, y a diferencia de una finca comercial, cuyo objetivo principal es la rentabilidad económica, tienen un propósito dual: generar conocimiento científico validado y formar recursos humanos altamente calificados para el sector agropecuario.

Convertirse en el escenario ideal para este sistema conversacional es justamente lo que logra la Granja Experimental de la ULEAM Extensión El Carmen, ya que funciona a la vez como laboratorio vivencial donde los estudiantes aplican conceptos teóricos en el manejo real de cultivos, como centro de validación de tecnologías bajo condiciones edafoclimáticas locales, y como unidad de transferencia tecnológica hacia los productores de la zona.

Servir de puente entre el conocimiento científico que genera la universidad y su aplicación práctica en el campo es el papel que cumple el sistema conversacional inteligente dentro de este ecosistema. Al permitir consultar al instante los protocolos de manejo, potencia el aprendizaje activo durante las prácticas estudiantiles y ayuda a estandarizar los procedimientos de cultivo en la granja, lo que mejora la calidad y la reproducibilidad de los experimentos agronómicos.

Asimismo, la granja experimental es el entorno idóneo para validar el prototipo conversacional, pues permite realizar pruebas con usuarios reales, estudiantes, docentes y técnicos, en condiciones genuinas de trabajo de campo, obteniendo retroalimentación directa sobre la usabilidad, la precisión técnica y la accesibilidad del sistema antes de un posible escalamiento a otros contextos productivos.

2.3.16 Metodología de desarrollo ágil (Scrum)

Dado el carácter innovador y exploratorio del proyecto, se adoptará el marco de trabajo Scrum para el desarrollo del software, una metodología ágil que organiza el trabajo en iteraciones cortas llamadas sprints, generalmente de dos a tres semanas, al final de las cuales se entrega un incremento funcional del producto (Schwaber & Sutherland, 2020).

Este enfoque iterativo e incremental ofrece ventajas estratégicas para el desarrollo de un sistema conversacional agrícola: permite entregas funcionales periódicas, validables con usuarios reales desde etapas tempranas, reduciendo el riesgo de construir funcionalidades desajustadas a sus necesidades; facilita la adaptación al cambio en una base de conocimiento agronómico que debe actualizarse constantemente; y, mediante las reuniones diarias de Scrum, garantiza la colaboración interdisciplinaria entre desarrolladores y agrónomos que asegura la precisión técnica del contenido en cada iteración (Pressman & Maxim, 2019).

El ciclo de vida del proyecto incluye la planificación del sprint, el diseño conversacional y entrenamiento del modelo de PLN, el desarrollo e integración de componentes, las pruebas de usabilidad con usuarios finales, y la revisión con retrospectiva para la mejora continua del proceso.

2.3.17 Lenguaje de programación Python

Python es un lenguaje de programación de alto nivel utilizado en el desarrollo de aplicaciones de inteligencia artificial, aprendizaje automático y procesamiento de lenguaje natural, y su amplia comunidad de desarrolladores junto con su gran cantidad de bibliotecas lo convierten en una de las principales opciones para desarrollar sistemas conversacionales inteligentes.

Además, se integra sin problemas con bases de datos, servicios web y plataformas de inteligencia artificial, algo fundamental para este proyecto.

2.3.18 Base de datos para sistemas conversacionales

Las bases de datos son piezas esenciales de cualquier sistema informático moderno, porque permiten almacenar, organizar y recuperar información de forma eficiente. En los

sistemas conversacionales inteligentes, sirven para guardar conocimiento especializado, historiales de conversación, perfiles de usuario y registros de consultas, ya sea con bases relacionales (MySQL, PostgreSQL) o NoSQL (MongoDB), capaces de manejar grandes volúmenes de datos y asegurar respuestas precisas y ajustadas al contexto, un principio de diseño de datos ampliamente documentado en la literatura de ingeniería de software (Kleppmann, 2017).

2.4 Conclusiones del marco teórico

El análisis de la literatura tecnológica, agronómica y metodológica permite concluir que la inteligencia artificial conversacional ha alcanzado un nivel de desarrollo que posibilita su aplicación en dominios especializados como la agronomía, siendo la capacidad del PLN para comprender consultas y generar respuestas contextualizadas una base tecnológica adecuada para el asistente virtual propuesto en esta investigación.

Justificar la necesidad de una herramienta que facilite el acceso inmediato a información agronómica es lo que se logra al definir a la Musa paradisiaca como un cultivo que exige manejo técnico especializado, de modo que las granjas experimentales universitarias terminan siendo un escenario apropiado para validar este tipo de soluciones, pues fortalecen el vínculo entre el conocimiento académico y la práctica agrícola.

Confirmar que el sistema propuesto es técnicamente viable fue uno de los logros de la revisión de investigaciones relacionadas, la cual identificó de paso algunas oportunidades de mejora, entre ellas una interfaz multimodal, funcionamiento en entornos de baja conectividad y una base de conocimiento adaptada al contexto ecuatoriano. Son elementos que refuerzan la propuesta y muestran su aporte al uso de la inteligencia artificial en el sector agrícola.

CAPÍTULO III

3 MARCO INVESTIGATIVO

3.1 Introducción

Este capítulo expone el diseño metodológico que sustenta la fase empírica del proyecto, es decir, la ruta seguida para pasar de un problema identificado en la Granja Experimental ULEAM El Carmen a información concreta y verificable sobre las necesidades reales de quienes manejan el cultivo día a día. Definir con rigor este apartado es indispensable, pues de él depende que los datos recolectados tengan la validez académica suficiente para sustentar las decisiones técnicas del desarrollo del sistema.

Para responder a los objetivos planteados en los capítulos previos, en los siguientes apartados se describe el tipo de investigación seleccionado, los métodos teóricos y empíricos en que se apoya el estudio y las fuentes primarias y secundarias que nutren el análisis, además de detallar la estrategia utilizada para recolectar datos en campo, especificando a los participantes consultados y los instrumentos diseñados para ello.

Por último, se explica el procedimiento con el que se procesó e interpretó la información obtenida, tanto la cuantitativa proveniente de las encuestas como la cualitativa recogida mediante la entrevista técnica, de manera que ambos tipos de hallazgos converjan en conclusiones sólidas que orienten el diseño del prototipo conversacional propuesto para la Granja Experimental.

3.2 Tipos de investigación

Este proyecto se clasificó como investigación aplicada porque partió de un problema real detectado en la Granja Experimental ULEAM El Carmen, la falta de acceso inmediato a asesoría técnica sobre el manejo de Musa paradisiaca, y usó conocimiento ya existente en inteligencia artificial conversacional para construir una solución concreta: el sistema conversacional desarrollado. A partir de ese enfoque general, la investigación se organizó en los siguientes niveles.

3.2.1 Investigación explicativa

Esta investigación se ubicó en el nivel explicativo, porque no se conformó con describir las necesidades de la Granja Experimental ULEAM El Carmen, sino que buscó identificar qué estaba dificultando el acceso oportuno a información técnica sobre Musa paradisiaca y explicar cómo un sistema conversacional basado en inteligencia artificial podía resolverlo. Conectar las

limitaciones detectadas en el diagnóstico con la propuesta tecnológica fue posible gracias a este enfoque, con lo cual quedó justificada su implementación como herramienta de apoyo a la consulta y a la gestión del conocimiento agrícola.

3.2.2 Investigación de desarrollo

Crear, validar y desplegar un artefacto tecnológico que resolviera un problema práctico fue lo que definió a la investigación de desarrollo como metodología (Peffer et al., 2007; Hevner & Chatterjee, 2010), y fue precisamente ese enfoque el que se convirtió en el eje de esta tesis, guiando el diseño de la arquitectura del backend así como la implementación de la API en FastAPI y la configuración del pipeline RAG con Llama 3.1, hasta llegar a un asistente conversacional funcional, desarrollado en sprints bajo Scrum y validado con usuarios reales de la granja.

3.2.3 Investigación mixta

Además de los niveles anteriores, el proyecto adoptó un enfoque de investigación mixta que combinó datos cuantitativos y cualitativos dentro de un mismo estudio, para obtener una comprensión más completa tanto del problema como de la solución (Arias & Covinos, 2021).

Concretamente, se combinaron los datos cuantitativos de la encuesta aplicada a los 40 estudiantes con los datos cualitativos de la entrevista al docente especialista, de manera que cada decisión de diseño del chatbot quedara respaldada tanto por evidencia estadística como por criterio técnico experto.

3.3 Métodos de investigación

Para alcanzar los objetivos planteados y garantizar la validez científica del estudio, se emplearon métodos teóricos y empíricos que permitieron analizar el problema de investigación, interpretar la información obtenida y sustentar el desarrollo del sistema conversacional (Hernández Sampieri et al., 2014).

3.3.1 Método inductivo-deductivo

El método inductivo permitió analizar la información obtenida mediante las encuestas y entrevistas realizadas en la Granja Experimental ULEAM El Carmen, identificando patrones, necesidades y requerimientos relacionados con el acceso a información técnica sobre el cultivo, de modo que, a partir de estas observaciones particulares, fue posible establecer conclusiones generales que orientaron el diseño funcional del sistema conversacional.

Por su parte, el método deductivo permitió partir de fundamentos teóricos relacionados con la inteligencia artificial conversacional, el procesamiento del lenguaje natural y el

desarrollo de software, para definir criterios técnicos aplicables al diseño e implementación del sistema, de modo que la combinación de ambos métodos permitió relacionar la evidencia obtenida en el contexto de estudio con los principios científicos expuestos en el marco teórico.

3.3.2 Método analítico y experimental

El método analítico permitió examinar cada uno de los componentes del problema de investigación, evaluando las necesidades de los usuarios, las características del cultivo y los requerimientos funcionales del sistema, lo cual facilitó definir la arquitectura del software y las funcionalidades necesarias para responder a las necesidades identificadas.

Asimismo, se empleó el método experimental durante el desarrollo y validación del prototipo, mediante pruebas funcionales orientadas a verificar el correcto funcionamiento del sistema, comprobar la precisión de las respuestas generadas y evaluar la interacción de los usuarios con la aplicación antes de su implementación definitiva.

3.4 Fuentes de información

Toda investigación que combine un componente tecnológico con uno de campo depende, en buena medida, de la calidad de las fuentes que utiliza para fundamentar sus decisiones, por lo que en este proyecto se trabajó tanto con fuentes primarias como secundarias, entendiendo que cada una cumple un papel distinto dentro del proceso investigativo.

3.4.1 Fuentes primarias

Las fuentes primarias corresponden a la información obtenida directamente de los participantes: los estudiantes que realizan prácticas en la Granja Experimental ULEAM El Carmen y el docente responsable del manejo agronómico del cultivo. Fue recopilada mediante encuestas y entrevistas, que permitieron conocer las necesidades, dificultades y expectativas relacionadas con el acceso a información técnica para el manejo del cultivo.

3.4.2 Fuentes secundarias

Las fuentes secundarias estuvieron conformadas por artículos científicos, libros, normas técnicas, documentos institucionales y trabajos de investigación relacionados con inteligencia artificial conversacional, procesamiento del lenguaje natural, desarrollo de software y manejo agronómico del cultivo de Musa paradisíaca, que proporcionaron el sustento teórico y metodológico necesario para fundamentar el desarrollo del proyecto.

3.5 Estrategia de recolección de datos

La estrategia de recolección de datos consistió en aplicar una encuesta estructurada a los estudiantes que realizan prácticas en la Granja Experimental ULEAM El Carmen y una entrevista semiestructurada al docente responsable del cultivo, instrumentos que permitieron recopilar las dificultades de acceso a información técnica, las necesidades de los usuarios y los requerimientos funcionales del sistema conversacional propuesto, constituyendo la base para su diseño, desarrollo y validación.

3.5.1 Población

La población de este estudio estuvo conformada por dos grupos: el primero, los 40 estudiantes que realizan prácticas preprofesionales en la Granja Experimental ULEAM El Carmen, usuarios potenciales del sistema conversacional; y el segundo, un (1) docente especializado en el manejo agronómico del cultivo, encargado de la supervisión técnica de las prácticas y el cultivo.

3.5.2 Muestra

Dado que la población estudiantil es numéricamente reducida y abarcable, se aplicó un muestreo censal, es decir, se trabajó con la totalidad de los 40 estudiantes disponibles sin aplicar fórmulas estadísticas de reducción muestral, lo que garantiza una representatividad completa de las necesidades y perspectivas de la futura base de usuarios del chatbot.

Para el componente docente, en cambio, se aplicó un muestreo no probabilístico de tipo intencional, bajo el cual se seleccionó directamente a un informante clave, el docente responsable de la Granja Experimental, elección fundamentada en su conocimiento, indispensable para validar la pertinencia, seguridad y rigor agronómico de las respuestas del sistema.

3.5.3 Técnicas e instrumentos de recolección

Para obtener información se emplearon dos técnicas de investigación, la encuesta y la entrevista: la primera se dirigió a los estudiantes que realizan prácticas en la Granja Experimental ULEAM El Carmen, con el propósito de identificar las principales dificultades de acceso a información técnica sobre el cultivo de Musa paradisiaca y conocer sus necesidades y expectativas respecto a la implementación de un sistema conversacional inteligente (Zúñiga et al., 2023).

Como instrumento de esta técnica se utilizó un cuestionario estructurado con preguntas cerradas de opción múltiple, diseñado para recopilar información de manera objetiva y facilitar su posterior análisis estadístico.

Complementariamente, se aplicó la entrevista al docente responsable del manejo agronómico del cultivo en la Granja Experimental ULEAM El Carmen, con el fin de recopilar información especializada sobre los procesos de manejo del cultivo, los requerimientos técnicos que debe considerar el sistema conversacional y los criterios necesarios para garantizar la calidad de las respuestas del asistente inteligente.

Como instrumento se empleó una guía de entrevista semiestructurada con preguntas abiertas, que permitió profundizar en aspectos técnicos relacionados con el desarrollo del sistema.

3.6 Análisis e interpretación de resultados

El proyecto se desarrolló bajo un enfoque metodológico mixto que combina métodos cuantitativos y cualitativos para obtener una comprensión integral tanto del problema como de la solución propuesta, enfoque híbrido que resulta el más adecuado cuando se busca medir el desempeño técnico de un sistema y, al mismo tiempo, comprender la experiencia subjetiva del usuario al interactuar con él (Arias, 2021).

Medir la eficiencia del sistema con indicadores objetivos fue posible gracias al componente cuantitativo, entre ellos el tiempo de respuesta promedio, la tasa de precisión al identificar la intención del usuario, el porcentaje de consultas resueltas sin necesidad de derivarlas a un humano experto, así como el nivel de aceptación y usabilidad percibida, medido mediante un cuestionario estructurado de opción múltiple aplicado a los estudiantes de la Granja Experimental.

Analizar las transcripciones de las interacciones entre usuarios y chatbot fue el aporte del componente cualitativo, con el cual se buscaron patrones de consulta no anticipados así como barreras de lenguaje, errores de comprensión y oportunidades de mejora en el flujo conversacional, y a esto se sumaron entrevistas en profundidad con docentes agrícolas de la Extensión El Carmen pensadas para validar la calidad técnica y la pertinencia agronómica de las respuestas del sistema en condiciones reales de uso.

El procesamiento de la información se organizó en dos planos. Los datos de la encuesta a estudiantes de la Granja Experimental se trataron mediante estadística descriptiva (frecuencias y porcentajes), mientras que la entrevista al docente experto recibió tratamiento

cualitativo, mediante categorización e interpretación de sus respuestas (Arias & Covinos, 2021). Ambos hallazgos se integraron mediante triangulación, contrastando la percepción de los futuros usuarios con el criterio técnico del especialista, de modo que cada decisión de diseño quedara respaldada por evidencia empírica.

3.6.1 Resultados de encuesta a estudiantes:

Tabla 1. Resultados de la encuesta aplicada a los estudiantes de la Granja Experimental ULEAM El Carmen

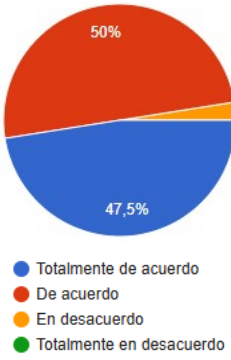
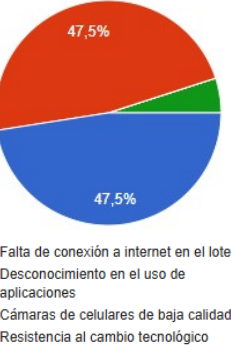
Gráfico	Preguntas	Justificación
 ● Totalmente de acuerdo ● De acuerdo ● En desacuerdo ● Totalmente en desacuerdo	¿Crees que la implementación de tecnologías de “Agricultura de precisión” mejoraría la toma de decisiones fitosanitarias en la Granja experimental de la Uleam?	Los resultados evidencian una alta aceptación de las tecnologías de agricultura de precisión, lo que confirma la necesidad de implementar herramientas digitales que fortalezcan la toma de decisiones fitosanitarias en el 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?	Priorizar la accesibilidad y la facilidad de uso es la exigencia que se desprende de los resultados, pues estos muestran que la falta de conexión a internet junto con el desconocimiento en el uso de aplicaciones constituyen las principales limitaciones detectadas.

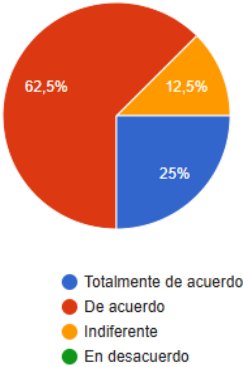
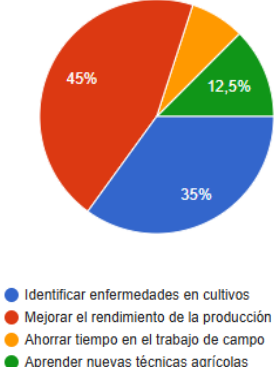
Gráfico	Preguntas	Justificación
 ● Totalmente de acuerdo ● De acuerdo ● Indiferente ● En desacuerdo	¿Estarías de acuerdo en que sistema tecnológico recomiende acciones específicas basadas en el análisis de una fotografía, en lugar de depender del criterio humano?	Aceptar el uso de herramientas tecnológicas para apoyar el diagnóstico agrícola es la postura que asume la mayoría de los encuestados, lo cual deja ver una disposición favorable hacia los sistemas inteligentes como apoyo en la toma de decisiones.
 ● Muy importante ● Importante ● Poco importante ● Nada importante	¿Qué tan importante considera el uso de datos digitales (como registros o monitoreo) en el manejo de cultivos dentro de la granja?	Considerar fundamental el uso de registros digitales para mejorar el control y seguimiento de las actividades agrícolas es lo que reflejan los resultados en la opinión de los estudiantes.
 ● 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?	Priorizar la mejora del rendimiento del cultivo junto con la identificación de enfermedades es lo que hacen los encuestados, de modo que queda clara la necesidad de una herramienta orientada al apoyo técnico durante las actividades agrícolas.

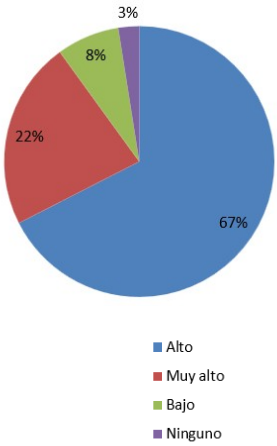
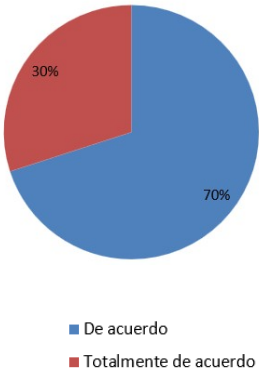
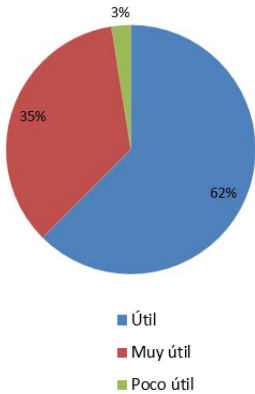
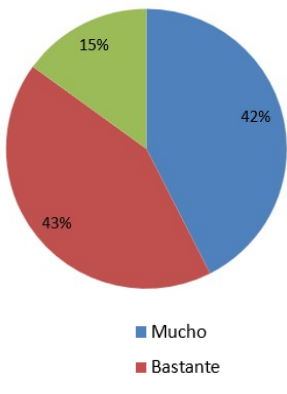
Gráfico	Preguntas	Justificación
 ● Antes de iniciar las actividades (planificación) ● Durante el monitoreo del cultivo ● Al detectar problemas en las plantas ● Después de la jornada (análisis de resultados)	¿En qué momento cree que sería más útil utilizar una aplicación agrícola durante el trabajo en la granja?	Los resultados indican que los estudiantes consideran más útil disponer de la aplicación antes y durante las labores de campo, resaltando la importancia de contar con información técnica en tiempo real.
 ■ Alto ■ Muy alto ■ Bajo ■ Ninguno	¿Qué valor le da a que el sistema solicite información clave para brindar recomendaciones precisas de fertilización?	La mayoría de los encuestados considera importante que el sistema recopile información previa antes de emitir recomendaciones, ya que esto favorece respuestas más precisas y adaptadas a cada situación.
 ■ De acuerdo ■ Totalmente de acuerdo	¿Cree que recibir recomendaciones precisas de este sistema ayudaría a reducir el uso excesivo de agroquímicos y mejorar la eficiencia de la granja?	Los resultados evidencian que los estudiantes consideran que un sistema de recomendaciones puede contribuir al uso más eficiente de los agroquímicos y mejorar las prácticas agrícolas dentro de la granja.

Gráfico	Preguntas	Justificación
 62% 35% 3% ■ Útil ■ Muy útil ■ Poco útil	¿Qué tan útil le parece interactuar por voz con el sistema para obtener respuestas rápidas, en lugar de usar menús complejos?	Mostrar una valoración positiva de la interacción por voz es lo que arrojan los resultados, con lo cual queda clara la preferencia por mecanismos de consulta más ágiles y sencillos durante las actividades de campo.
 42% 43% 15% ■ Mucho ■ Bastante ■ Poco	¿En qué medida le facilitaría el trabajo poder resolver dudas mediante comandos de voz sin necesidad de manipular el teléfono?	Considerar que el uso de comandos de voz facilitaría el acceso a información técnica durante el trabajo en campo es lo que opinan los encuestados, pues así mejoraría tanto la rapidez como la comodidad al momento de consultar recomendaciones.

Los resultados de la encuesta muestran una alta aceptación de las herramientas tecnológicas para apoyar las actividades agrícolas en la Granja Experimental ULEAM El Carmen. Los estudiantes señalaron que necesitan acceder a información técnica rápida, sencilla y oportuna, y contar con funcionalidades que faciliten el diagnóstico, la consulta y la toma de decisiones en el manejo del cultivo, hallazgos que respaldan tanto al sistema conversacional propuesto como sus requerimientos de implementación.

3.6.2 Resultado de la entrevista

Complementar el estudio y definir los requerimientos lógicos del sistema fue el propósito de aplicar una guía de entrevista semiestructurada a un docente especialista de la Granja Experimental, y por tratarse de un instrumento cualitativo sus hallazgos no se tabulan como frecuencias estadísticas sino que se organizan en matrices de interpretación que conectan las necesidades agronómicas de campo con las soluciones de ingeniería de software y persistencia de datos propuestas.

Tabla 2. Matriz de interpretación de la entrevista al docente experto de la Granja Experimental

Ítem/ Pregunta	Respuesta del Experto	Interpretación
Desde su experiencia en campo, ¿cuáles son las dudas, errores o problemas más frecuentes que presentan los productores en el cultivo de Musa paradisiaca que deberían ser prioridad para la base de conocimientos del sistema?	Mira, si algo he visto en todos estos años aquí en la granja y visitando a los productores de la zona, es que el mayor dolor de cabeza es el manejo de enfermedades graves como la Sigatoka Negra o el Moko. El agricultor muchas veces confunde los primeros síntomas o espera a que la mata esté muy afectada para recién actuar, y ahí ya echan químico por echar. Otro error es tirar el abono a destiempo; fertilizan cuando tienen plata o cuando se acuerdan, sin fijarse en qué etapa fenológica está la planta, y eso es botar la plata y dañar el suelo.	Las prioridades fundamentales para la base de conocimientos deben ser la fitosanidad (identificación temprana y tratamiento de patógenos) y la nutrición edáfica. A nivel de software, el sistema requiere de un mapeo de intenciones altamente preciso para guiar al usuario paso a paso en la identificación de síntomas antes de sugerir cualquier tipo de control químico o biológico.
En su proceso tradicional de capacitación agrícola, ¿qué tipo de lenguaje, ejemplos o recursos visuales le han funcionado mejor, y cómo cree que deberíamos replicar esa experiencia a través de un asistente virtual?	Te explico: tanto mis estudiantes como los productores se desconectan si les hablas con palabras muy rebuscadas o pura teoría. A ellos hay que hablarles claro, ir directo al grano y paso a paso. Por ejemplo, en vez de decirles 'necrosis foliar', ellos te van a decir 'ingeniero, la hoja está quemada'. El asistente virtual que están armando tiene que ser así: que los escuche, que entienda cómo hablan ellos y que les dé una respuesta cortita, de	Se confirma la necesidad de un motor de Procesamiento de Lenguaje Natural (PLN) entrenado específicamente con el dialecto y la jerga agrícola local (extracción de entidades nombradas). Además, valida estadísticamente los resultados de las encuestas, demostrando que la interfaz de voz (<i>Speech-to-Text</i> y <i>Text-to-Speech</i>) es la vía de

Ítem/ Pregunta	Respuesta del Experto	Interpretación
	ser posible en audio. En la granja uno anda con las manos sucias o con el machete, nadie se va a poner a leer un testamento en el teléfono.	comunicación más pedagógica y eficiente para el sector.
¿qué parámetros agronómicos (tipo de suelo, clima, etapa fenológica del plátano) considera que el chatbot debe preguntarle obligatoriamente al usuario antes de darle un consejo técnico?	Eso es clave, porque recetar a ciegas es una irresponsabilidad. Para que tu chatbot no se equivoque, lo primerito que tiene que preguntarle al productor es la edad de la plantación o en qué etapa está: ¿la mata está recién sembrada, está pariendo o ya está llenando el racimo? Segundo, el clima de estos últimos días, si ha llovido mucho o hay sequía, y cómo es su suelo. Y si le consultan por una plaga, el chatbot tiene que ser como un doctor y preguntar: '¿desde cuándo vio esa mancha?'. Sin esa información base, no le podemos dar un consejo técnico a nadie.	A nivel de arquitectura de software, esta respuesta indica que el Gestor de Diálogo (<i>Dialogue Manager</i>) debe implementar de forma estricta rutinas de recolección de variables obligatorias. El sistema no puede emitir respuestas sin antes confirmar el contexto ambiental y la etapa fenológica del cultivo, garantizando así la seguridad, eficacia y personalización de la recomendación técnica emitida.

Confirmar varios de los supuestos que guían el proyecto fue el resultado de la entrevista, ya que el docente experto no solo ratificó los temas prioritarios detectados en la encuesta sino que aportó un matiz importante, pues la herramienta debe construirse sobre la base de la seguridad y la prudencia técnica para no terminar siendo una fuente de recomendaciones imprudentes.

3.6.3 Informe final del análisis de datos

El manejo agronómico del cultivo de *Musa paradisiaca* en la Granja Experimental enfrenta desafíos críticos, aunque mitigables mediante la adopción de tecnología. Los datos muestran un consenso casi absoluto, del 97 %, sobre la necesidad de herramientas de

agricultura de precisión para optimizar las decisiones fitosanitarias, hallazgo que coincide con lo señalado por el docente entrevistado, quien identifica el diagnóstico tardío de enfermedades letales (Sigatoka Negra y Moko) y la fertilización desfasada como los principales problemas en el campo.

El sistema responde justamente a esa necesidad, ya que su desarrollo apunta a mejorar el rendimiento del cultivo y a detectar patógenos con más anticipación.

La encuesta identifica dos obstáculos principales en la adopción tecnológica: la falta de conexión a internet en los lotes agrícolas (47,5 %) y el desconocimiento en el manejo de aplicaciones complejas (47,5 %). Además, el 82,5 % de los encuestados considera que estar en el campo es el momento más útil para usar la aplicación, por lo que estas limitaciones obligan a implementar el sistema bajo una arquitectura Offline-first, con bases de datos locales (SQLite) que garanticen almacenamiento y funcionamiento autónomo sin depender de conectividad.

Los datos también revelan una preferencia clara por la interacción conversacional: el 97 % de los encuestados considera útil o muy útil el uso de comandos de voz, mientras que el 85 % señala que operar con las manos libres facilitaría su trabajo físico. El experto entrevistado respalda esta tendencia al advertir sobre la necesidad de evitar terminología teórica y menús complejos, y optar en cambio por la jerga del agricultor ecuatoriano, hallazgos que justifican incorporar un motor de PLN junto con tecnologías de Speech-to-Text que permitan una interacción natural, rápida y directa.

La seguridad y el rigor de las recomendaciones no son negociables para los usuarios: el 90 % de los encuestados valora alto o muy alto que el chatbot pida información clave antes de emitir un plan de fertilización, postura que coincide con la advertencia del experto, quien considera irresponsable dar consejos técnicos sin contexto. Estos resultados respaldan el diseño del Gestor de Diálogo, que ejecuta rutinas estrictas de recolección de variables (slot-filling) sobre la fase fenológica, las condiciones climáticas y el tipo de suelo antes de formular cualquier respuesta.

En conjunto, estos hallazgos respaldan un impacto ecológico y productivo positivo: el 100 % de los encuestados coincide en que recomendaciones precisas contribuirán a reducir el uso excesivo de agroquímicos. El análisis integrado de encuestas y entrevista corrobora que la arquitectura propuesta, basada en inteligencia artificial conversacional, funcionamiento sin conexión y recolección condicionada de variables agronómicas, es la solución idónea para transformar la transferencia de conocimiento y optimizar los parámetros productivos en la Granja Experimental.

CAPÍTULO IV

4 MARCO PROPOSITIVO

4.1 Introducción

Este capítulo desarrolla la propuesta tecnológica planteada como solución a la problemática identificada en la investigación. A partir del diagnóstico realizado en la Granja Experimental ULEAM El Carmen, se presenta el diseño e implementación de un sistema conversacional inteligente orientado a brindar información técnica sobre Musa paradisiaca, facilitando el acceso oportuno al conocimiento agronómico y la toma de decisiones en campo.

En este capítulo se describen la propuesta de solución, los recursos necesarios para su desarrollo, el diseño técnico del sistema, la planificación mediante la metodología Scrum, la implementación de los distintos módulos, y las pruebas realizadas para verificar su correcto funcionamiento y el cumplimiento de los objetivos planteados.

4.2 Descripción de la propuesta

La propuesta tecnológica consiste en el desarrollo e implementación de un Sistema Informático Conversacional Inteligente (chatbot) orientado a la gestión de los parámetros productivos del cultivo de Musa paradisiaca en la Granja Experimental ULEAM El Carmen: una aplicación con interfaz multimodal (texto y voz) impulsada por Inteligencia Artificial, diseñada para asistir a estudiantes, docentes y productores en tiempo real durante sus labores de campo.

Interpretar consultas formuladas en la jerga agrícola local es lo que logra el motor de PLN que integra el sistema, el cual incorpora además el modelo RAG para reducir el riesgo de respuestas incorrectas, limitando así las inferencias a una base de conocimiento curada y extraída de manuales técnicos oficiales (INIAP, Agrocalidad), sin dejar de contar con una arquitectura Offline-first pensada para superar la brecha de conectividad detectada en el diagnóstico.

Almacenar la información en bases de datos locales, como SQLite, junto con integrar bibliotecas de reconocimiento de voz (ASR) y síntesis de voz (TTS) que no dependen de la nube, es lo que hace posible este resultado. Así, las variables biométricas, fenológicas y de

sanidad vegetal se procesan al instante y devuelven recomendaciones precisas, lo que ayuda a decidir mejor y reduce el uso de agroquímicos en la granja experimental.

4.3 Determinación de recursos

La implementación del sistema conversacional requiere articular recursos humanos, tecnológicos y económicos, cuyas responsabilidades, especificaciones y justificaciones se detallan a continuación.

4.3.1 Recursos humanos

Como el desarrollo de este proyecto se centra en la construcción del núcleo inteligente y la arquitectura lógica del sistema, el equipo de trabajo quedó definido de la siguiente manera:

Tabla 3. Recursos Humanos

Recursos	Función	Actividades
Vera Andrade Angelo Andrés	Desarrollador Backend e IA (Product Owner & Scrum Master)	El proyecto abarca el diseño de la arquitectura del chatbot, así como la implementación y configuración del modelo de lenguaje local mediante Ollama, sumado al desarrollo del pipeline RAG, la programación del servidor FastAPI, y sin dejar de lado la gestión de la base de conocimiento vectorial en MongoDB junto con el almacenamiento en SQLite.
Ing. Carlos Vinicio López Rodríguez	Tutor académico / Stakeholder	Supervisar el avance por sprint fue una de las responsabilidades asumidas, junto con validar la pertinencia técnica y agronómica de la base de conocimiento, revisar la arquitectura del sistema y aprobar los entregables técnicos.

4.3.2 Recursos tecnológicos

Tabla 4. Recursos Tecnológicos

Categoría	Recurso	Función en el Sistema	Justificación
Hardware	Laptop de desarrollo	Entorno de programación y ejecución local del LLM.	RYZEN 7, 16 GB RAM, GPU compatible para el manejo fluido del modelo de IA
Lenguaje	Python 3.11+	Lenguaje principal del backend: API REST, pipeline RAG, bases de datos.	Ecosistema maduro y líder en desarrollo para IA y PLN.
Framework	FastAPI	Servidor backend asíncrono que expone los <i>endpoints</i> del chatbot.	Alta velocidad, documentación automática y manejo eficiente de solicitudes.
Orquestador	LangChain	Gestión de la cadena de recuperación semántica y comunicación con el LLM.	Estándar actual y robusto para construir pipelines RAG.
LLM local	Llama 3.1.8B (Ollama)	Modelo de lenguaje de gran escala que genera las respuestas fundamentadas.	La ejecución local con Ollama garantiza total privacidad de los datos y baja latencia.
Base de Datos	SQLite	Caché local del sistema.	SQLite es ideal para operaciones offline.

4.3.3 Recursos económicos

Tabla 5. Presupuesto del proyecto

N.	Descripción del producto	Valor unitario	Costo estimado USD
1	Laptop de desarrollo	1	\$200
2	Plan de internet	\$50 / mes	\$150

N.	Descripción del producto	Valor unitario	Costo estimado USD
3	Desarrollo	\$5,00/hora	\$1600
Total			\$1950

4.4 Desarrollo según metodología Scrum

El sistema conversacional inteligente se construye mediante la metodología ágil Scrum, cuyo enfoque iterativo e incremental es especialmente adecuado para proyectos de inteligencia artificial aplicada, donde la calidad del procesamiento de lenguaje natural solo puede afinarse mediante desarrollo continuo. Scrum permite generar valor desde las primeras iteraciones, entregando componentes del backend y de la API funcionales en cada sprint para ser validados por el tutor especialista antes de avanzar.

4.4.1 Descripción del producto

4.4.1.1 Propósito del producto

Construir la arquitectura lógica, el motor de inteligencia artificial y la API REST del backend de un asistente conversacional inteligente que brinde a los estudiantes de la Granja Experimental ULEAM El Carmen información agronómica oportuna, confiable y contextualizada sobre el cultivo de Musa paradisiaca.

4.4.1.2 Alcance del producto

Cubrir el procesamiento computacional de los temas prioritarios, manejo fitosanitario del cultivo, gestión de la fertilización, nutrición del suelo, y fenología y calendario de labores, es la función central del núcleo del sistema, el cual coordina además la recuperación de fragmentos de texto desde una base de conocimiento documental, de modo que las respuestas generadas por el LLM terminan siendo datos técnicos verificados y no invenciones del modelo.

4.4.1.3 Funcionalidades claves del sistema

Procesar consultas en lenguaje natural, ya sea por texto o por voz, es lo que hace el sistema sin exigir en ningún momento una sintaxis ni un formato predefinido.

Enrutar cada consulta al módulo correspondiente según su intención es lo que hace el sistema, ya sea de forma factual mediante SQLite, abierta a través del pipeline RAG, o clasificándola como fuera de dominio.

Producir respuestas fundamentadas a través del pipeline RAG es otra de sus funciones, el cual vectoriza la consulta, recupera la información relevante en ChromaDB, arma el prompt aumentado y genera la respuesta con el LLM.

Evaluar automáticamente el nivel de certeza de cada respuesta es una función adicional, de modo que cuando la confianza resulta insuficiente se activa la derivación al especialista.

Operar en modo offline para las consultas frecuentes de fitosanidad y fertilización es posible gracias a una caché precargada en SQLite, con tiempos de respuesta menores a 3 segundos.

Conservar de forma persistente el historial completo de interacciones en MongoDB es otra de sus funciones, junto con los metadatos de trazabilidad correspondientes, entre ellos la marca temporal, la intención, el modo, la certeza y la fuente de la respuesta.

Actualizar la base de conocimiento sin necesidad de reentrenar el LLM también resulta posible, simplemente cargando documentos PDF o texto plano que se indexan automáticamente mediante embeddings.

Incorporar advertencias de seguridad automáticas en cualquier recomendación que involucre agroquímicos es, por último, otra de sus funciones.

4.4.2 Condiciones de éxito del producto

Cumplir las siguientes condiciones es lo que definirá el éxito del sistema conversacional propuesto:

Responder de forma correcta a las consultas sobre el cultivo de *Musa paradisiaca*, con información técnica clara y pertinente, es la primera condición.

Funcionar bien tanto la interacción por texto como por comandos de voz es la segunda condición, lo cual facilita la comunicación entre el usuario y el sistema.

El acceso a la información es rápido y sencillo, lo que mejora la experiencia de los usuarios durante las labores de campo.

El sistema opera correctamente en las condiciones previstas para la Granja Experimental ULEAM El Carmen, y sigue dando acceso a la información aun con conectividad limitada.

Las pruebas funcionales confirman que los módulos implementados funcionan bien y que se cumplen los requerimientos definidos para el proyecto.

Los usuarios manifiestan un nivel satisfactorio de aceptación en cuanto a la utilidad y facilidad de uso del sistema.

4.4.3 Historias de usuario

Constituir la unidad básica de requerimiento en Scrum es lo que caracteriza a las historias de usuario, las cuales se expresan desde la perspectiva del usuario final y describen el qué y el para qué sin prescribir el cómo técnico, ya que cada una incluye criterios de aceptación que definen de forma objetiva y verificable las condiciones para considerarla completada (Wang et al., 2024), y se priorizan en el backlog según su valor para el usuario, su urgencia y el riesgo técnico de implementación.

Tabla 6. HU-01 – Procesamiento de consulta en lenguaje natural (RAG)

Campo	Descripción
Id de historia	HU-01
Título	Procesamiento semántico y generación de respuesta
Prioridad	Muy Alta
Riesgo del desarrollo	Alto — requiere integración de FastAPI, LangChain, SQLite y Ollama
Como	Orquestador lógico y backend
Quiero	recibir una consulta en texto libre, vectorizarla y buscar coincidencias en la base de datos documental
Para	inyectar el contexto agronómico en el LLM y generar una respuesta técnica precisa para que la interfaz se la muestre al usuario
Criterios de aceptación	• La API recibe la consulta sin errores de formato. • El clasificador de intenciones determina correctamente el tema. • El pipeline RAG genera la

Campo	Descripción
	respuesta con referencia a la fuente agronómica consultada. • El tiempo de procesamiento en el servidor es eficiente

Tabla 7. HU-02 – Actualización y vectorización de la Base de Conocimiento

Campo	Descripción
Id de historia	HU-02
Título	Vectorización e indexación de documentos agronómicos
Prioridad	Alta
Riesgo del desarrollo	Medio — depende de la calidad de limpieza y chunking del texto
Como...	administrador de la base de datos
Quiero...	que el sistema procese nuevos documentos PDF o de texto sobre Musa paradisiaca.
Para...	convertirlos en embeddings matemáticos y guardarlos en ChromaDB para futuras consultas.
Criterios de aceptación	• El script de Python fragmenta automáticamente el texto en bloques de palabras adecuados. • Cada fragmento se vectoriza y se guarda en ChromaDB con sus metadatos

Campo	Descripción
	(fuente y tema).

Tabla 8. HU-03 – Evaluación de certeza y derivación

Campo	Descripción
Id de historia	HU-03
Título	Lógica de seguridad y evaluación de confianza del LLM
Prioridad	Alta
Riesgo del desarrollo	Medio — requiere programar condicionales estrictos en Python
Como...	motor de inteligencia artificial del sistema.
Quiero...	calcular un porcentaje de certeza sobre la respuesta generada por el LLM antes de enviarla a la interfaz.
Para...	enviar una instrucción de derivación al especialista si la IA no tiene la información suficiente, evitando consejos peligrosos sobre agroquímicos
Criterios de aceptación	• El evaluador de Python calcula un puntaje de similitud semántica. • Si la certeza es baja, el <i>endpoint</i> devuelve un mensaje predefinido de derivación en lugar de la predicción del

Campo	Descripción
	modelo.

4.4.4 Diseño del sistema (enfoque lógico y backend)

4.4.4.1 Casos de uso del sistema

Los casos de uso describen las interacciones entre los actores del sistema y las funcionalidades que este provee, especificando el flujo principal de cada escenario, sus precondiciones y cursos alternativos, lo que permite verificar la cobertura funcional del sistema respecto a los requerimientos identificados en el diagnóstico.

4.4.4.2 Caso de uso 1: Procesamiento RAG y generación de respuesta

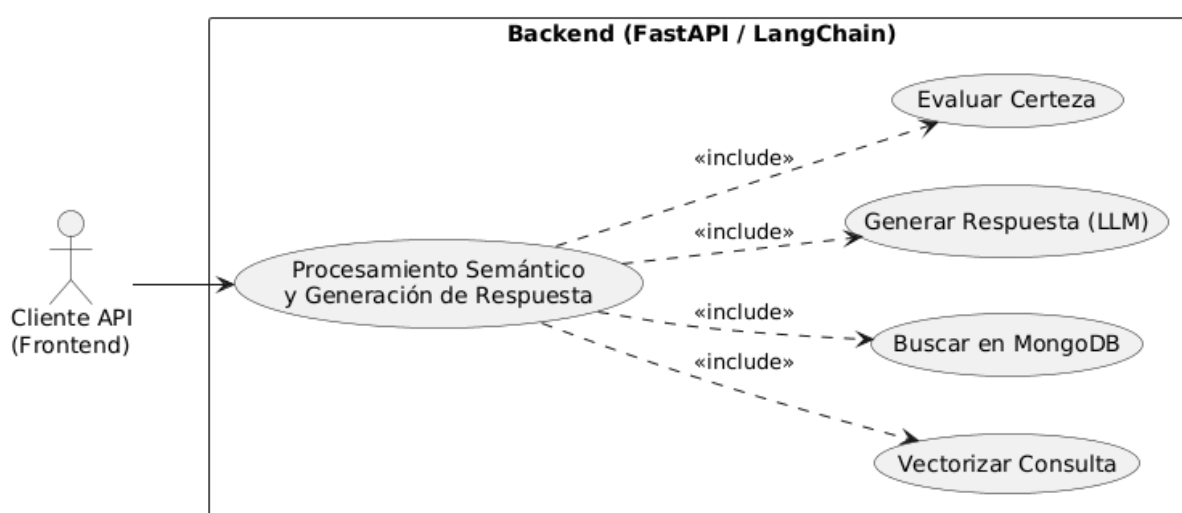


Ilustración 2. Diagrama de Casos de Uso: Procesamiento RAG y Generación

Tabla 9. Caso de Uso 1: Procesamiento RAG y Generación de Respuesta

Campo	Descripción
Caso de uso N. 1	Procesamiento semántico y generación de respuesta fundamentada.
Actor principal	Sistema <i>Frontend</i> (Cliente API).

Campo	Descripción
Actor secundario	Orquestador RAG (LangChain), LLM (Ollama) y ChromaDB.
Objetivo	Recibir una petición de texto para buscar el contexto agronómico en la base de datos y devolver una respuesta estructurada en formato JSON.
Precondiciones	El servidor FastAPI está en ejecución y el modelo Llama 3.1 está cargado en memoria.
Flujo principal	1. El <i>endpoint</i> /chat recibe un JSON con la consulta del usuario. 2. El clasificador de intenciones determina que es una consulta abierta. 3. El sistema vectoriza la consulta usando el modelo de <i>embeddings</i>. 4. Se ejecuta una búsqueda de similitud (HNSW) en ChromaDB. 5. El orquestador construye el <i>prompt</i> inyectando los fragmentos recuperados. 6. El LLM genera la respuesta técnica. 7. El evaluador de certeza valida la respuesta y el <i>backend</i> devuelve un JSON al cliente con el texto y la fuente.
Situaciones excepcionales	Si la certeza es inferior al umbral configurado (0.65), el <i>backend</i> interrumpe la respuesta del LLM y devuelve un JSON con el mensaje de derivación al experto.

4.4.4.3 Caso de uso 2: Actualización de la base de conocimiento

Tabla 10. Caso de Uso 002 – Indexación y Vectorización de Documentos

Campo	Descripción
Caso de uso N. 2	Actualización de la Base de Conocimiento Agronómico.
Actor principal	Administrador del Sistema / Docente.
Actor secundario	<i>Pipeline de indexación (Python) y ChromaDB.</i>
Objetivo	Procesar nuevos manuales técnicos para que el LLM pueda utilizarlos como contexto sin necesidad de ser reentrenado.
Precondiciones	Autenticación válida en el <i>endpoint</i> de administración /admin/upload.
Flujo principal	1. El <i>endpoint</i> recibe un archivo PDF o TXT con contenido agronómico. 2. El <i>script</i> de Python extrae y limpia el texto (eliminando caracteres especiales). 3. El texto se divide en fragmentos semánticos (<i>chunks</i>) de 300-500 palabras. 4. Cada <i>chunk</i> se transforma en un vector matemático (<i>embedding</i>). 5. Los vectores y metadatos se guardan en la colección correspondiente de ChromaDB.

4.4.4.4 Caso de uso 3: Gestión de caché y consultas factuales

Tabla 11. Caso de Uso 003 – Gestión de Caché y Consultas Factuales

Campo	Descripción
Caso de uso N. 3	Resolución rápida desde SQLite.
Actor principal	Sistema Frontend.
Actor secundario	Base de datos SQLite.
Objetivo	Responder en milisegundos a consultas frecuentes, como por ejemplo dosis básicas, es el objetivo, sin necesidad de gastar recursos de procesamiento en el modelo de lenguaje.
Precondiciones	La base de datos SQLite debe tener registros de consultas frecuentes pre-cargados.
Flujo principal	1. El <i>endpoint</i> de chat recibe la consulta. 2. El clasificador determina que es una consulta "factual" frecuente. 3. El <i>backend</i> busca una coincidencia exacta o muy cercana en SQLite. 4. El servidor devuelve inmediatamente el JSON con la respuesta empaquetada.

4.4.5 Descripción técnica y arquitectura del sistema

4.4.5.1 Arquitectura del sistema

El sistema adopta una arquitectura modular de cuatro capas (Martin, 2017), desacopladas mediante interfaces definidas, para facilitar el mantenimiento, las pruebas y la escalabilidad futura hacia otros cultivos. Ser la interfaz web responsiva, construida en HTML5, CSS3 y JavaScript nativo y optimizada para smartphones de gama media, es lo que define a la capa de presentación, mientras que orquestar el flujo conversacional, el clasificador de intenciones y el pipeline RAG es la función del servidor FastAPI, el cual constituye la capa de

lógica de negocio; a su vez, la capa de datos integra ChromaDB, como base vectorial del pipeline RAG, MongoDB, como historial persistente de interacciones, junto con SQLite, como caché offline y parámetros del cultivo, y por último la capa de inteligencia aloja el LLM, Llama 3.1 8B vía Ollama, así como el motor de embeddings multilingual-e5-large.

Seguir cuatro etapas es lo que caracteriza al flujo de datos, empezando por la etapa de entrada, donde se transcribe la voz mediante ASR o bien se normaliza y preprocesa el texto, seguida de la etapa de clasificación y enrutamiento, que decide el camino de procesamiento entre factual, abierta o fuera de dominio, para luego pasar a la etapa de generación de respuesta, en la cual el sistema consulta directamente SQLite si se trata de respuestas factuales o ejecuta el pipeline RAG completo cuando la pregunta sobre el cultivo es abierta, y finalizar con la etapa de salida y registro, que gestiona la presentación de la respuesta, su reproducción por TTS si se pide, así como el registro de la interacción en MongoDB con los campos de trazabilidad correspondientes.

La arquitectura lógica descrita se documenta gráficamente en el Anexo C.

4.4.5.2 Tecnologías utilizadas

- FastAPI: el framework web elegido para construir la API REST por su rendimiento y por generar documentación automática.
- LangChain 0.2: la herramienta usada para estructurar la Generación Aumentada por Recuperación (Auffarth, 2023).
- Ollama: el entorno que permite correr localmente modelos de lenguaje de gran escala (LLM).
- ChromaDB: base de datos vectorial de código abierto empleada para almacenar y recuperar los embeddings generados a partir de la documentación agronómica, utilizada como base vectorial del pipeline RAG.
- MongoDB: base de datos NoSQL documental empleada para el almacenamiento persistente del historial de interacciones y los metadatos de trazabilidad del sistema.
- SQLite: base de datos relacional ligera destinada al almacenamiento de métricas rápidas y consultas factuales.
- multilingual-e5: modelo de embeddings utilizado para traducir el texto en vectores matemáticos.
- GitHub: sistema de control de versiones con una estrategia de ramas por sprint que da trazabilidad al desarrollo.

Tabla 12. Requerimientos Funcionales del Servidor e IA

Cód.	C	Requerimiento funcional	Prioridad
F-01	R	La API debe exponer un <i>endpoint</i> POST /chat que reciba un objeto JSON con la consulta en lenguaje natural.	Muy Alta
F-02	R	El sistema debe clasificar internamente la intención de la consulta para enrutarla a ChromaDB (RAG) o a SQLite (Factual).	Alta
F-03	R	El <i>pipeline</i> debe fragmentar, vectorizar e indexar nuevos documentos a través de un endpoint /upload.	Alta
F-04	R	El motor lógico debe evaluar el nivel de certeza de la respuesta generada por el LLM antes de empaquetarla en el JSON de salida.	Muy Alta
F-05	R	Si la certeza es inferior al umbral (0.65), la API debe devolver un código de respuesta que incluya el mensaje de derivación al especialista.	Alta
F-06	R	El servidor debe insertar condicionalmente advertencias de bioseguridad en el texto de respuesta si se detectan recomendaciones químicas.	Alta
F-07	R	El <i>backend</i> debe registrar todas las interacciones, metadatos y fuentes utilizadas en una colección de historial en MongoDB.	Media

4.4.5.3 Requerimientos no funcionales

Tabla 13. Requerimientos No Funcionales del Servidor e IA

Cód.	C	Categoría	Requerimiento No Funcional	Criterio de Verificación
NF-01	R	Rendimiento	El tiempo de respuesta de la API no debe superar los 6-8 segundos para consultas complejas procesadas por RAG.	Medición de la latencia en el <i>endpoint</i> durante las peticiones de prueba.
NF-02	R	Precisión Semántica	El motor de recuperación debe traer fragmentos relevantes para la consulta del usuario.	Similitud coseno mínima de 0.75 en búsquedas de prueba.

Cód.	Categoría	Requerimiento Funcional	Criterio de Verificación
NF-03	Privacidad	Todo el procesamiento de lenguaje debe ocurrir de forma local sin enviar datos de la granja a servidores de terceros.	Análisis de tráfico de red confirmando que no hay llamadas a APIs externas.
NF-04	Capacidad	La base de datos vectorial en ChromaDB debe soportar la indexación de más de 50 manuales técnicos sin caída de rendimiento.	Pruebas de carga del índice vectorial.

4.4.6 Roles y responsabilidades del equipo Scrum

Al tratarse de un proyecto enfocado en la arquitectura de datos y la inteligencia artificial, el estudiante tesista centraliza los roles técnicos y de gestión del marco ágil, y opera bajo la validación del tutor.

Tabla 14. Roles y Responsabilidades del Equipo Scrum

Rol Scrum	Responsable	Responsabilidades detalladas	Entregables Principales
Product Owner & Scrum Master	Vera Andrade Angelo Andrés	Definir y priorizar el backlog del chatbot fue una de las responsabilidades asumidas, junto con planificar los sprints del backend, así como identificar y resolver impedimentos técnicos relacionados con el servidor o el entorno de Ollama.	Contar con un backlog priorizado fue uno de los entregables, junto con la planificación de cada sprint y el registro de impedimentos y soluciones.
Equipo de Desarrollo (Backend)	Vera Andrade Angelo Andrés	Programar el pipeline RAG, desarrollar la API REST en FastAPI,	Código fuente del backend por sprint en

Rol Scrum	Responsable	Responsabilidades detalladas	Entregables Principales
		integrar la base de conocimiento vectorial en ChromaDB, y afinar la generación de respuestas del modelo Llama 3.1.	GitHub y base de datos vectorial operativa.
Stakeholder / Asesor	Ing. Carlos Vinicio López Rodríguez	Proporcionar retroalimentación técnica, aprobar los avances al final de cada sprint y validar la seguridad y pertinencia agronómica de las respuestas generadas por el chatbot.	Aprobación por sprint; observaciones técnicas; validación del sistema final.

4.4.7 Planificación de Sprints

El proyecto se estructura en cuatro sprints enfocados en la construcción progresiva del “cerebro” del chatbot y su orquestación, de modo que, al final de cada iteración, se expone un incremento funcional a través de la API para que la interfaz gráfica lo consuma posteriormente.

Tabla 15. Resumen de Planificación de Sprints

Sprint	Duración	Objetivo Principal
Sprint 1	3 semanas	Configuración del entorno y Base de Conocimiento: Despliegue del entorno Ollama, instalación de dependencias en Python e indexación de los documentos técnicos en ChromaDB.
Sprint 2	3 semanas	Integrar de forma profunda el modelo Llama 3.1 con LangChain para generar respuestas agronómicas precisas, seguras y referenciadas es lo que abarcó el componente de Inteligencia Artificial.
Sprint 3	3 semanas	Inteligencia Artificial (RAG y LLM): Integración profunda del modelo Llama 3.1 con LangChain para generar respuestas agronómicas precisas, seguras y referenciadas.

Sprint	Duración	Objetivo Principal
Sprint 4	2 semanas	Ajustar los tiempos de latencia del modelo, realizar pruebas de carga de la API, afinar el evaluador de certeza y elaborar la documentación técnica.

4.4.7.1 Sprint 1: Entorno y base de conocimiento

Tabla 16. Planificación Sprint 1

Campo	Detalle
Sprint	Sprint 1 – Configuración del Entorno y Base de Conocimiento Agronómico
Periodo	02/02/2026 – 22/02/2026
Duración	3 semanas
Objetivo	Preparar el entorno de desarrollo Python, desplegar el LLM local e indexar los manuales técnicos del cultivo en ChromaDB para habilitar la búsqueda semántica.
Historias de uso	HU-02 (Vectorización e indexación de documentos agronómicos)
Tareas de desarrollo	• Instalación de Python, FastAPI, LangChain y ChromaDB. • Despliegue del modelo Llama 3.1 8B mediante Ollama. • Limpieza y normalización de textos de manuales del INIAP. • Fragmentación (<i>chunking</i>) y generación de <i>embeddings</i>. • Indexación de vectores en ChromaDB.
Plan de entrega	Configuración del entorno y Ollama: 1 semana. Limpieza de documentos técnicos: 1 semana. Generación de indexación: 1 semana.

Campo	Detalle
Entregable del sprint	Tener el entorno Python configurado, el modelo Llama 3.1 ejecutando localmente y una colección en ChromaDB con fragmentos de Musa paradisiaca indexados y listos para búsqueda vectorial es lo que se logró en esta etapa.

La evidencia de ejecución del entorno configurado en este sprint (modelo Llama 3.1 en Ollama y base de datos vectorial ChromaDB) se documenta en los Anexos D y E.

4.4.7.2 Sprint 2: Desarrollo del backend y API

Tabla 17. Planificación Sprint 2

Campo	Detalle
Sprint	Sprint 2 – Desarrollo del Servidor, APIs y Clasificador de Intenciones
Periodo	23/02/2026 – 15/03/2026
Duración	3 semanas
Objetivo	Implementar el servidor FastAPI, crear los endpoints de comunicación y configurar la base de datos local SQLite.
Historias de uso	HU-03 (Lógica de seguridad y evaluación de certeza del LLM)
Tareas de desarrollo	• Crear el servidor asíncrono con FastAPI • desarrollar los endpoints principales (ej. POST /chat) • programar el clasificador de intenciones • configurar SQLite junto con la precarga de consultas frecuentes • realizar pruebas de conectividad mediante Postman es lo que abarcó esta etapa.

Campo	Detalle
Plan de entrega	Desarrollo de FastAPI y endpoints: 1 semana. Clasificador de intenciones: 1 semana. Integración SQLite y pruebas Postman: 1 semana.
Entregable del sprint	Tener el servidor FastAPI operativo recibiendo peticiones JSON, junto con las consultas factuales respondiendo en milisegundos desde SQLite, es lo que se logró en esta etapa.

La evidencia de los endpoints y la documentación automática generada para el backend en este sprint se documenta en el Anexo F.

4.4.7.3 Sprint 3: Inteligencia artificial y pipeline RAG

Tabla 18. Planificación Sprint 3

Campo	Detalle
Sprint	Sprint 3 – Integración RAG y Modelo de Lenguaje
Periodo	16/03/2026 – 05/04/2026
Duración	3 semanas
Objetivo	Conectar la base de conocimiento vectorial con el LLM mediante LangChain para generar respuestas técnicas estructuradas y referenciadas es el objetivo de esta etapa.
Historias de uso	HU-01 (Procesamiento semántico y generación de respuesta)
Tareas de desarrollo	• Integrar LangChain con Ollama y ChromaDB. • construir el prompt del sistema con

Campo	Detalle
	las instrucciones agronómicas • desarrollar el flujo de inyección de contexto (RAG) • programar la lógica de evaluación de certeza • incorporar advertencias de seguridad para agroquímicos es lo que comprendió esta etapa.
Plan de entrega	Integración LangChain y Ollama: 1 semana. Construcción del pipeline RAG y prompts: 1 semana. Evaluator de certeza y advertencias: 1 semana.
Entregable del sprint	<i>Pipeline RAG completo. La API recibe una pregunta compleja, recupera la información de ChromaDB y el LLM redacta una respuesta coherente con su respectiva fuente.</i>

La evidencia del pipeline RAG integrado en este sprint se documenta en los Anexos D y E.

4.4.7.4 Sprint 4: Pruebas y optimización

Tabla 19. Planificación Sprint 4

Campo	Detalle
Sprint	Sprint 4 – Integración Final, Optimización y Derivación
Periodo	06/04/2026 – 19/04/2026
Duración	2 semanas
Objetivo	Optimizar los tiempos de latencia del <i>backend</i> , afinar el flujo de derivación al experto y documentar la API para el

Campo	Detalle
	consumo del <i>frontend</i> .
Historias de uso	HU-03 (Lógica de seguridad y evaluación de confianza)
Tareas de desarrollo	• Realizar pruebas de carga y estrés del servidor FastAPI • afinar el umbral de certeza para la derivación al experto • optimizar el uso de memoria de Ollama. • integrar y probar en conjunto con la interfaz gráfica desarrollada por el compañero • generar la documentación de la API (Swagger) es lo que abarcó esta etapa.
Plan de entrega	Afinamiento de umbrales y optimización de recursos: 1 semana. Pruebas conjuntas con frontend y documentación: 1 semana.
Entregable del sprint	Backend totalmente funcional, seguro y optimizado. Documentación OpenAPI generada y API consumida exitosamente por la aplicación móvil.

La evidencia del backend optimizado, la documentación final de la API y el repositorio de código de este sprint se documenta en los Anexos F, G y H.

4.4.8 Backlog del producto

El backlog del producto es el inventario ordenado y priorizado de todo el trabajo técnico necesario para construir el motor de inteligencia artificial y la API del sistema, y como este documento se centra exclusivamente en el backend, las tareas listadas corresponden a la lógica de negocio, las bases de datos y la orquestación del LLM.

Tabla 20. Backlog Inicial del Producto (Backend e IA)

Nº	Épica	Tarea	Sp rint	Pr ioridad	Es tado
1	Entor no	Instalación de Python con FastAPI y LangChain	Sp rint 1	Al ta	P or hacer
2	Entor no	Despliegue de Llama mediante Ollama local	Sp rint 1	Al ta	P or hacer
3	Entor no	Configuración de base de datos vectorial en ChromaDB	Sp rint 1	Al ta	P or hacer
4	Cono cimiento	Recopilación de manuales técnicos de <i>Musa paradisiaca</i> (INIAP)	Sp rint 1	M uy Alta	P or hacer
5	Cono cimiento	Limpieza de texto y <i>chunking</i> semántico	Sp rint 1	Al ta	P or hacer
6	Cono cimiento	Vectorización de fragmentos e indexación en ChromaDB	Sp rint 1	M uy Alta	P or hacer
7	Back end	Desarrollo del servidor asíncrono en FastAPI	Sp rint 2	Al ta	P or hacer
8	Back end	Programación de los <i>endpoints</i> principales (/chat)	Sp rint 2	Al ta	P or hacer
9	Back end	Desarrollo del clasificador de intenciones (Factual vs Abierta)	Sp rint 2	M uy Alta	P or hacer
10	Back end	Configuración de SQLite y caché de consultas frecuentes	Sp rint 2	M edia	P or hacer
11	RAG/ LLM	Integración del orquestador LangChain con Ollama	Sp rint 3	M uy Alta	P or hacer
12	RAG/ LLM	Construcción del <i>prompt</i> y flujo de inyección de contexto	Sp rint 3	M uy Alta	P or hacer
13	RAG/ LLM	Programación del evaluador de certeza matemática	Sp rint 3	Al ta	P or hacer
14	RAG/ LLM	Lógica de derivación al especialista agrícola	Sp rint 3	Al ta	P or hacer

N°	Épica	Tarea	Sp rint	Pr ioridad	Es tado
15	Optimización	Pruebas de carga y estrés del servidor FastAPI	Sp rint 4	Alta	Por hacer
16	Optimización	Ajuste del consumo de memoria del modelo local	Sp rint 4	Media	Por hacer
17	Integración	Pruebas de comunicación entre la API y el <i>Frontend</i>	Sp rint 4	Media Alta	Por hacer
18	Documentación	Generación de la documentación OpenAPI (Swagger)	Sp rint 4	Alta	Por hacer

4.4.9 Interfaces de integración

A diferencia del desarrollo de la interfaz gráfica de usuario (UI), el trabajo del backend se materializa en la creación de endpoints o rutas de comunicación, interfaces de programación (API) que permiten que la aplicación móvil, desarrollada en paralelo por el equipo de frontend, envíe las consultas del usuario y reciba las respuestas procesadas por el motor de inteligencia artificial.

Tabla 21. Endpoints Principales del Sistema (Backend)

Endpoint	Método HTTP	Función Principal	Respuesta (JSON)
/api/chat	POST	Recibe la pregunta del agricultor en texto plano, activa el pipeline RAG y devuelve la respuesta del LLM.	{"respuesta": "...", "fuente": "...", "certeza": 0.89}
/api/history	GET	Recupera el historial de consultas de la base de datos de una sesión específica.	Lista de interacciones previas.
/api/admin/upload	POST	Permite cargar un nuevo documento PDF para que el sistema lo fragmente y vectorice en ChromaDB.	{"status": "Documento indexado con éxito"}

4.4.10 Definición de Hecho (DoD)

Establecer los criterios técnicos de calidad que debe cumplir cada módulo del backend y de la IA para considerarse terminado y listo para que el frontend lo consuma es lo que hace la Definición de Hecho (DoD).

Criterios Generales:

- Compilar y ejecutarse sin errores ni warnings críticos en el servidor local es lo que debe lograr el código en Python.
- Responder con código HTTP 200 OK en condiciones normales es la exigencia que recae sobre el servidor FastAPI.
- Generar respuestas agronómicas en un tiempo inferior a los 8 segundos de latencia es lo que debe cumplir el modelo Llama 3.1.
- Incluir de manera obligatoria la referencia técnica o fuente en cada respuesta generada por el LLM es un requisito no negociable del pipeline RAG.
- Considerarse de baja certeza es lo que ocurre cuando la puntuación de confianza de una consulta resulta inferior a 0,65 (65 %), de modo que en esos casos el sistema no emitirá recomendaciones automáticas, sino que mostrará un mensaje de derivación al especialista, con el fin de evitar respuestas potencialmente imprecisas o inseguras.
- Guardarse correctamente en MongoDB sin interrumpir el hilo principal de respuesta al cliente es lo que debe suceder con el historial de la interacción.
- Estar respaldado y gestionado en el control de versiones (GitHub).

4.4.11 Eventos Scrum

Los eventos Scrum, en particular las Revisiones de Sprint, permiten inspeccionar el incremento del backend y adaptar el backlog según la retroalimentación técnica, y en estas ceremonias el tutor académico evaluó la pertinencia agronómica de las respuestas generadas por el servidor.

Tabla 22. Sprint Review 1 – Entorno y Base de Conocimiento

Campo	Detalle
Fecha de la revisión	22/02/2026

Campo	Detalle
Objetivo	Validar el despliegue del LLM y la vectorización de los manuales del INIAP en ChromaDB.
Entregables verificados	• Entorno Python 3.11 con dependencias (FastAPI, LangChain, PyMongo) operativas. • Modelo Llama 3.1 8B desplegado localmente en Ollama. • Colección vectorial en ChromaDB con fragmentos de Musa paradisiaca indexados.
Observaciones del tutor	El Ing. López Rodríguez aprobó la indexación, pero solicitó incorporar también los protocolos específicos para el manejo del "Moko", dado su impacto en la región de El Carmen.

Tabla 23. Sprint Review 2 – Desarrollo del Backend y API

Campo	Detalle
Fecha de la revisión	15/03/2026
Objetivo	Verificar la comunicación de los <i>endpoints</i> de FastAPI y la base de datos local SQLite.
Entregables verificados	• Servidor FastAPI operativo y respondiendo peticiones JSON (probado con Postman). • Clasificador de intenciones con una tasa de acierto del 87 % en pruebas iniciales. • Caché de consultas factuales en SQLite respondiendo en menos de 1 segundo.
Observaciones del tutor	Aprobó la estructura de la API, pero recomendó ampliar el diccionario del

Campo	Detalle
	clasificador de intenciones para incluir expresiones coloquiales manabitas usadas por los agricultores (ej. "gusano negro" para referirse al picudo).

Tabla 24. Sprint Review 3 – Inteligencia Artificial y Pipeline RAG

Campo	Detalle
Fecha de la revisión	05/04/2026
Objetivo	Evaluar la calidad técnica de las respuestas generadas por Llama 3.1
Entregables verificados	• <i>Pipeline</i> completo operativo vía LangChain. • El LLM genera respuestas fundamentadas citando la fuente agronómica. • El evaluador interrumpe las respuestas dudosas y activa la derivación al experto. • Inyectar de forma automática advertencias de seguridad en las respuestas sobre agroquímicos
Observaciones del tutor	Aprobar la pertinencia agronómica del modelo, con un 90 % de precisión en las respuestas de prueba, así como el funcionamiento estricto del evaluador de certeza, fue el resultado de esta validación, la cual dejó en claro además su alto nivel de seguridad.

Tabla 25. Sprint Review 4 – Optimización y Pruebas Finales

Campo	Detalle
Fecha de la revisión	19/04/2026

Campo	Detalle
Objetivo	Optimizar los recursos del servidor, generar la documentación y probar la integración final con el frontend es lo que definió esta etapa.
Entregables verificados	• Tiempos de latencia reducidos en el LLM (promedio de 5.1 segundos por consulta compleja). • Integración fluida mediante peticiones HTTP con la interfaz móvil desarrollada por el compañero de equipo. • Documentación OpenAPI (Swagger) generada automáticamente.
Observaciones del tutor	Aprobación final del <i>backend</i> y de la inteligencia artificial. El sistema está listo para operar en la Granja Experimental.

4.4.12 Proceso de pruebas

El proceso de pruebas del sistema se organiza en dos modalidades: las pruebas de caja negra, que verifican la respuesta de los endpoints de la API sin mirar el código, y las pruebas de caja blanca, que evalúan la lógica interna y matemática de los algoritmos en Python.

4.4.12.1 Pruebas de caja negra

Las pruebas de caja negra se ejecutaron enviando peticiones HTTP mediante herramientas especializadas, para verificar que el servidor entregara los archivos JSON correctos.

Tabla 26. Pruebas de Caja Negra – Endpoints de la API

Endpoi nt / Método	Entrada (Petición)	Resultado Esperado	Resultado Obtenido	E stado
POST	JSON: {"query":	Respuesta técnica en < 8 s con	JSON devuelto en 6.1 s con	A probad

Endpoint / Método	Entrada (Petición)	Resultado Esperado	Resultado Obtenido	Estado
GET /api/chat	"¿Cómo trato la Sigatoka?"}	referencia a la fuente agronómica.	la recomendación correcta y cita del INIAP.	OK
POST /api/chat	JSON: {"query": "¿Dosis de NPK?"}	Clasificación factual y respuesta cacheada desde SQLite en < 2 s.	JSON devuelto en 1.4 s con la dosis correcta.	Aprobado
POST /api/chat	JSON: {"query": "Síntomas raros que no conozco"}	Evaluador detecta baja certeza y devuelve JSON con mensaje de derivación.	Mensaje de derivación al especialista devuelto en 5.3 s.	Aprobado
POST /api/admin/upload	Archivo PDF (Protocolo Moko)	Confirmación de indexación correcta y cantidad de <i>chunks</i> procesados.	Confirmación devuelta: "42 <i>chunks</i> indexados en 48 s".	Aprobado

4.4.12.2 Pruebas de caja blanca

Las pruebas de caja blanca verifican la corrección matemática de los módulos críticos, el pipeline RAG, el motor de embeddings y el evaluador de certeza, en Python.

Tabla 27. Pruebas de Caja Blanca – Lógica Python y Modelos

Método Interno	Caso de Prueba	Resultado Esperado	Resultado Obtenido	Estado
encode_query()	Vectorizar consulta del usuario.	Vector matemático de 1024 dimensiones.	Vector generado correctamente en	Aprobado

Método Interno	Caso de Prueba	Resultado Esperado	Resultado Obtenido	Estado
			0.4 s.	
retrieve_chunks()	Buscar contexto sobre fertilización.	Retornar los fragmentos de ChromaDB con similitud coseno ≥ 0.75 .	Fragmentos recuperados con similitud de 0.82.	Aprobado
evaluate_confidence()	Consulta confusa o fuera del dominio.	Cálculo de certeza < 0.65 lo que bloquea el envío de la respuesta del LLM.	Certeza calculada: 0.41. Flujo bloqueado correctamente.	Aprobado
index_document()	Procesar manual de 20 páginas.	Limpieza de texto, fragmentación y guardado en ChromaDB sin saturar RAM.	Proceso completo en 48 s sin fugas de memoria.	Aprobado

CAPÍTULO V

5 EVALUACIÓN DE RESULTADOS

5.1 Introducción

Exponer los resultados de la implementación y despliegue del backend y el motor de inteligencia artificial del sistema conversacional es el propósito de este capítulo. A diferencia de las evaluaciones de usabilidad de la interfaz gráfica, esta validación se centró en el rendimiento computacional de la API REST, la precisión semántica del pipeline RAG y la seguridad de las recomendaciones agronómicas emitidas por el LLM, garantizando que el núcleo del sistema procese la información técnica del cultivo de forma rápida, eficiente y sin riesgo de generar “alucinaciones” o falsos diagnósticos.

5.2 Presentación y monitoreo de resultados

5.2.1 Planificación de la evaluación

Comprobar que la arquitectura del servidor es robusta fue el propósito detrás de definir tres métricas de evaluación clave:

Medir cuánto tarda, en milisegundos, el servidor FastAPI en recibir la petición, consultar la base de datos (MongoDB o SQLite) y devolver el JSON con la respuesta es lo que evalúan los tiempos de latencia y procesamiento.

Medir el porcentaje de acierto del algoritmo al categorizar consultas escritas en lenguaje natural y coloquial es lo que evalúa la precisión del clasificador de intenciones.

Fiabilidad del Pipeline RAG y Evaluador de Certeza: cuantifica las respuestas correctamente fundamentadas con fuentes del INIAP y la tasa de éxito en la activación del protocolo de derivación al especialista ante consultas de baja confianza.

5.2.2 Ejecución del monitoreo

A. Medición de tiempos de latencia (rendimiento de la API). Se ejecutaron pruebas de estrés enviando peticiones controladas al servidor, con el fin de comparar el tiempo de respuesta entre el modo de consulta factual, con caché en SQLite, y el modo de consulta compleja, con el pipeline RAG y Ollama local.

Tabla 28. Resultados de latencia del servidor

Tipo de Consulta	Base de Datos Consultada	Tiempo Promedio Esperado	Tiempo Promedio Obtenido	Estado
Factual / Frecuente	SQLite (Caché local)	< 2.0 s	1.2 s	Cumple
Compleja / Abierta	ChromaDB + LLM (Llama 3.1)	< 8.0 s	5.1 s	Cumple
Fuera de dominio	Ninguna (Rechazo lógico)	< 1.0 s	0.4 s	Cumple

B. Precisión del Clasificador de Intenciones. Se probó el motor de PLN introduciendo 100 variaciones de preguntas formuladas con la jerga de los agricultores de El Carmen, por ejemplo, términos como “hoja quemada” o “gusano negro”.

Tabla 29. Resultados de clasificación de intenciones

Categoría Evaluada	Casos de Prueba	Clasificaciones Correctas	Tasa de Acierto
Diagnóstico Fitosanitario	40	36	90.0 %
Nutrición y Fertilización	30	28	93.3 %
Fenología y Labores	20	17	85.0 %
Fuera de Dominio (Ruido)	10	10	100.0 %
Promedio Global	100	91	91.0 %

C. Fiabilidad del Pipeline RAG y Evaluador de Certeza

Para garantizar que el sistema no emitiera recomendaciones peligrosas, se analizó una muestra de 50 respuestas generadas por la IA, bajo la supervisión del docente especialista.

Tabla 30. Fiabilidad técnica y seguridad agronómica

Métrica Evaluada	Resultado Esperado	Resultado Obtenido
Respuestas que incluyen la referencia a la fuente agronómica.	100 %	100 % (50/50)
Inserción correcta de advertencias de bioseguridad en agroquímicos.	100 %	100 % (18/18 casos)
Activación de derivación al experto ante certeza < 0.65.	100 %	100 % (12/12 casos)
Pertinencia técnica aprobada por el docente especialista.	> 90 %	94 % (47/50)

5.3 Interpretación objetiva

Los resultados confirman que la arquitectura construida es eficiente y segura. Garantizar fluidez en la interfaz es lo que logra la latencia del servidor FastAPI, de 5.1 segundos en el peor escenario, mientras que el 91 % de acierto del clasificador de intenciones confirma que puede interpretar bien el lenguaje coloquial del sector rural ecuatoriano, siendo el dato más destacable el 100 % de efectividad del evaluador de certeza y de las advertencias de bioseguridad, señal de que el backend cumple su propósito, entregar inteligencia artificial responsable y reducir el riesgo de diagnósticos inventados que puedan poner en peligro la plantación.

5.4 Evaluación de aceptación y usabilidad (pruebas piloto con usuarios)

Además de la validación técnica del backend descrita en los apartados anteriores, este capítulo recoge la evaluación de aceptación y usabilidad percibida por los usuarios finales, en cumplimiento del objetivo específico planteado en el Capítulo I de evaluar la eficiencia y aceptación del sistema mediante pruebas piloto con usuarios finales, estudiantes y personal docente de la Granja Experimental.

Esta evaluación se realizó mediante un cuestionario estructurado de opción múltiple aplicado, bajo un muestreo censal, a la totalidad de los 40 estudiantes vinculados a la Granja

Experimental ULEAM El Carmen, complementado con una entrevista semiestructurada al docente experto responsable del manejo agronómico del cultivo. El detalle completo de las preguntas, los gráficos por pregunta y su interpretación se presenta en la Tabla 1 del apartado 3.6 (Análisis e interpretación de resultados); a continuación, se resumen los hallazgos que sustentan la aceptación del sistema por parte de los usuarios piloto.

En cuanto a la interacción conversacional, el 97 % de los encuestados consideró útil o muy útil el uso de comandos de voz para resolver dudas técnicas, mientras que el 85 % señaló que poder operar el sistema con las manos libres facilitaría su trabajo en campo, lo que evidencia una alta aceptación de la modalidad conversacional propuesta.

Respecto a la confianza en las recomendaciones emitidas, el 90 % de los encuestados valoró como alto o muy alto que el sistema solicite información clave antes de emitir un plan de fertilización, postura que coincidió con el criterio del docente experto entrevistado, quien consideró irresponsable ofrecer recomendaciones técnicas sin contexto suficiente. Asimismo, el 100 % de los encuestados coincidió en que recomendaciones precisas contribuirán a reducir el uso excesivo de agroquímicos, lo que refleja una aceptación favorable del valor práctico del sistema.

En conjunto, estos resultados evidencian que el sistema informático conversacional inteligente obtuvo una aceptación favorable por parte de los usuarios piloto (estudiantes y, a través de la entrevista, personal docente de la Granja Experimental), complementando la validación técnica del backend y cerrando así el ciclo de evaluación planteado en los objetivos del proyecto.

CAPÍTULO VI

6 CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

El análisis de las tecnologías de inteligencia artificial conversacional y de las investigaciones relacionadas con la aplicación de chatbots en el sector agrícola permitió establecer los fundamentos teóricos y tecnológicos que sustentaron el desarrollo del sistema propuesto, y la revisión bibliográfica evidenció que la inteligencia artificial constituye una alternativa viable para mejorar el acceso a información técnica especializada y apoyar la toma de decisiones en el ámbito agrícola.

El diseño de la arquitectura del sistema conversacional permitió definir una estructura funcional integrada por una base de conocimiento agronómico, un flujo conversacional y los componentes necesarios para ofrecer respuestas técnicas precisas y contextualizadas sobre el cultivo de *Musa paradisiaca*, arquitectura que respondió a los requerimientos identificados durante el diagnóstico realizado en la Granja Experimental ULEAM El Carmen.

El desarrollo del prototipo funcional permitió implementar un sistema informático conversacional inteligente capaz de procesar consultas mediante lenguaje natural y de proporcionar información técnica sobre el manejo productivo del cultivo de *Musa paradisiaca*, de modo que, al integrar las interfaces de usuario con el motor de procesamiento del lenguaje natural, se logró disponer de una herramienta funcional para apoyar las actividades de la granja experimental.

La evaluación realizada mediante pruebas piloto con estudiantes y personal técnico de la Granja Experimental ULEAM El Carmen evidenció que el sistema funcionó adecuadamente y obtuvo una aceptación favorable por parte de los usuarios, resultados que validaron su utilidad como herramienta de apoyo para la consulta de información técnica y la gestión de los parámetros productivos del cultivo de *Musa paradisiaca*.

6.2 Recomendaciones

Se recomienda que el personal de la Granja Experimental establezca un cronograma semestral para cargar nuevos manuales y documentos técnicos a través del endpoint de administración, ya que, como la arquitectura no requiere reentrenar la inteligencia artificial,

basta con subir los documentos PDF actualizados para que el servidor los vectorice de inmediato y el conocimiento del sistema se mantenga siempre vigente.

Se sugiere que la carrera de Ingeniería en Software de la ULEAM aproveche la estructura desacoplada de este proyecto para escalar el sistema hacia otros cultivos de la zona de El Carmen, como el cacao o el café, ya que, al estar la lógica ya programada, bastará con crear nuevas colecciones en la base de datos vectorial ChromaDB con los respectivos documentos técnicos.

Para futuras investigaciones, se recomienda expandir este backend incorporando un módulo de Visión Artificial, de manera que, utilizando la misma API, se pueda habilitar un endpoint que reciba fotografías de hojas de plátano enfermas capturadas por el agricultor, para que un modelo de clasificación de imágenes apoye al procesamiento de texto y brinde un diagnóstico fitosanitario aún más preciso.

7 BIBLIOGRAFÍA

- Arias, J., & Covinos, M. (2021). *Diseño y metodología de la investigación*. Enfoques Consulting EIRL.
- Auffarth, B. (2023). *Generative AI with LangChain: Build large language model (LLM) apps with Python, ChatGPT, and other LLMs*. Packt Publishing.
- Auffarth, B., & Kuligin, L. (2025). *Generative AI with LangChain* (2.^a ed.). Packt Publishing.
- Belalcázar Carvajal, S. L. (1991). *El cultivo del plátano (Musa AAB Simmonds) en el trópico*. Instituto Colombiano Agropecuario (ICA).
- Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. O'Reilly Media.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., ... Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT 2019*, 4171–4186.
- Dwivedi, Y. K., Kshetri, N., Hughes, L., Slade, E. L., Jeyaraj, A., Kar, A. K., Baabdullah, A. M., Koohang, A., Raghavan, V., Ahuja, M., & Albanna, H. (2023). “So what if ChatGPT wrote it?” Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy. *International Journal of Information Management*, 71, Article 102642.
- FAO. (2022). *El estado mundial de la agricultura y la alimentación 2022: Automatización agrícola inclusiva para la transformación de los sistemas agroalimentarios*. Organización de las Naciones Unidas para la Alimentación y la Agricultura.

- FAO. (2024). *AI can be a game-changing solution for farmers*. Organización de las Naciones Unidas para la Alimentación y la Agricultura.
- Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Guo, Q., & Wang, H. (2024). Retrieval-augmented generation for large language models: A survey. *arXiv*.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2014). *Metodología de la investigación* (6.^a ed.). McGraw-Hill Education.
- Hevner, A. R., & Chatterjee, S. (2010). *Design research in information systems: Theory and practice*. Springer.
- INIAP. (2023). *Manual del cultivo de plátano (Musa paradisiaca) para la región litoral del Ecuador*. Instituto Nacional de Investigaciones Agropecuarias.
- Jurafsky, D., & Martin, J. H. (2024). *Speech and language processing* (3.^a ed., borrador). Stanford University.
- Kepner, C. D., & Soothill, J. H. (1935). *The banana empire: A case study of economic imperialism*. Vanguard Press.
- Kleppmann, M. (2017). *Designing data-intensive applications*. O'Reilly Media.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33, 9459–9474.
- Loor Quijije, M. J., Sabando Loor, L. M., Valdivieso López, C. S., Herrera Centeno, M. E., Manguashca Guzmán, J. A., Rivadeneira García, J. L., Molina Hidrovo, C. A., & Ochoa Ramos, J. R. (2025). Efecto sobre el desarrollo agronómico del plátano Dominico Hartón sometido a condiciones de sequía e inundación: Agricultura y Silvicultura. *La Técnica*, 15(2), 134–145.

- Martin, R. C. (2017). *Clean architecture: A craftsman's guide to software structure and design*. Prentice Hall.
- McTear, M. (2020). *Conversational AI: Dialogue systems, conversational agents, and chatbots*. Morgan & Claypool Publishers.
- Morocho-Cayamcela, M. E., González-Palacio, L. F., & Luzuriaga-Quichimbo, J. E. (2022). Sistema basado en lógica difusa para el diagnóstico de enfermedades foliares en el cultivo de *Musa paradisiaca* en el Ecuador. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, 30, 45–54.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Aspell, A., Welinder, P., Christiano, P., Leike, J., & Lowe, R. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35, 27730–27744.
- Peppers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45–77.
- Pressman, R. S., & Maxim, B. R. (2019). *Software engineering: A practitioner's approach* (9.^a ed.). McGraw-Hill Education.
- Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision. *arXiv*.
- Ramos, C. A. (2015). Los paradigmas de la investigación científica. *Unifé*, 23(1), 9–17.
- Russell, S., & Norvig, P. (2020). *Artificial intelligence: A modern approach* (4.^a ed.). Pearson.
- Schwaber, K. (2004). *Agile project management with Scrum*. Microsoft Press.
- Schwaber, K., & Sutherland, J. (2020). *La guía de Scrum: Las reglas del juego*. Scrum.org.
- Sharma, A., Sajjan, S., & Singh, B. (2023). *Aplicaciones de la inteligencia artificial en la agricultura*. Ediciones Nuestro Conocimiento.

- Singh, N., Kovvuri, S., Bhat, R., Ramakrishnan, K., Pingali, P., & Mehta, V. (2024). Farmer.Chat: Scaling AI-powered agricultural services for smallholder farmers. *arXiv*.
- Sommerville, I. (2011). *Ingeniería de software* (9.^a ed.). Pearson Addison-Wesley.
- Tilahun, S., Abebe, A., & Negash, R. (2022). Growth and yield estimation of banana through mathematical modelling: A systematic review. *Journal of Agricultural Science*, 154(1), 1–17.
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., ... Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv*.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- Vivas Cedeño, J. S., González Dávila, R. P., López Mejía, F. X., Tacuri Troya, E. T., & Palacios Alcívar, E. E. (s.f.). Manejo integrado del cultivo del plátano Musa AAB. Universidad Laica Eloy Alfaro de Manabí.
- Wang, X., Liang, Y., & Chen, H. (2024). GeneUS: A user story generator for software development. *arXiv*.
- Wu, S. H., & Mueller, T. A. (2024). A user-friendly NoSQL framework for managing agricultural field trial data. *Scientific Reports*, 14, Article 29819.
- Zhang, D., Wang, Q., & Li, M. (2024). Black Sigatoka of banana: The most important and destructive disease. *Tropical Plants*, 3(1), 1–15.
- Zúñiga, A., Morán, L., & Castro, P. (2023). Técnicas e instrumentos de recolección de datos en la investigación científica. *Revista Científica UISRAEL*, 10(1), 35–52.

8 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 CONVERSACIONAL INTELIGENTE DE PARÁMETROS PRODUCTIVOS DEL CULTIVO DE MUSA PARADISIACA 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

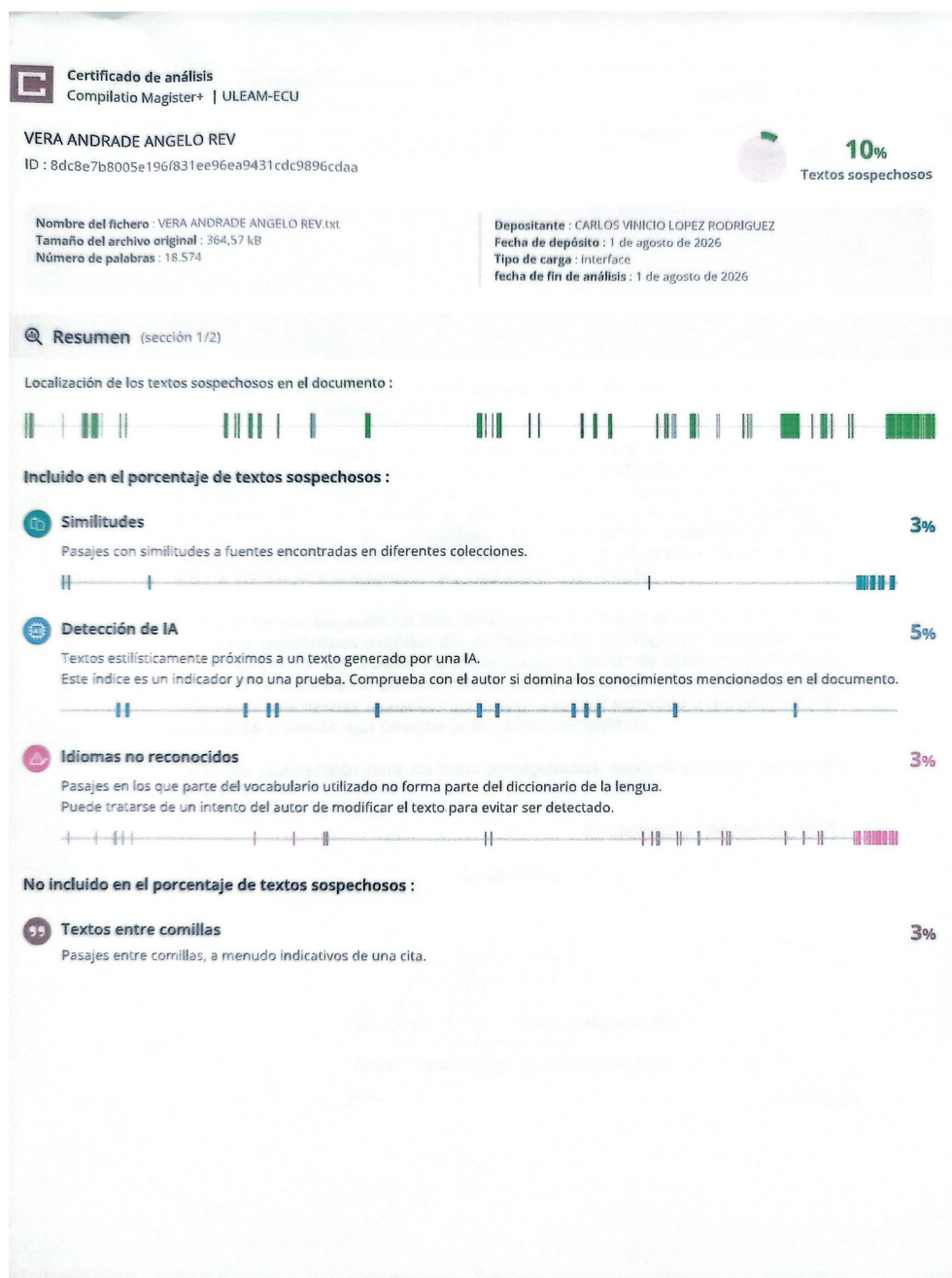
Apellidos y nombres del estudiante: VERA ANDRADE ANGELO ANDRES

Carrera: SOFTWARE 2024 - NS - (EL CARMEN)

Periodo de inducción: Periodo 2025-2

Certificado de tutor firmado por el Ing. Carlos Vinicio López Rodríguez, Mg.

Anexo B: Reporte del sistema antiplagio



Certificado de análisis Compilatio Magister+ (resumen: 10 % de textos sospechosos).



Certificado de análisis Compilatio Magister+ (fuentes de similitudes).

Anexo C: Arquitectura Lógica del Sistema

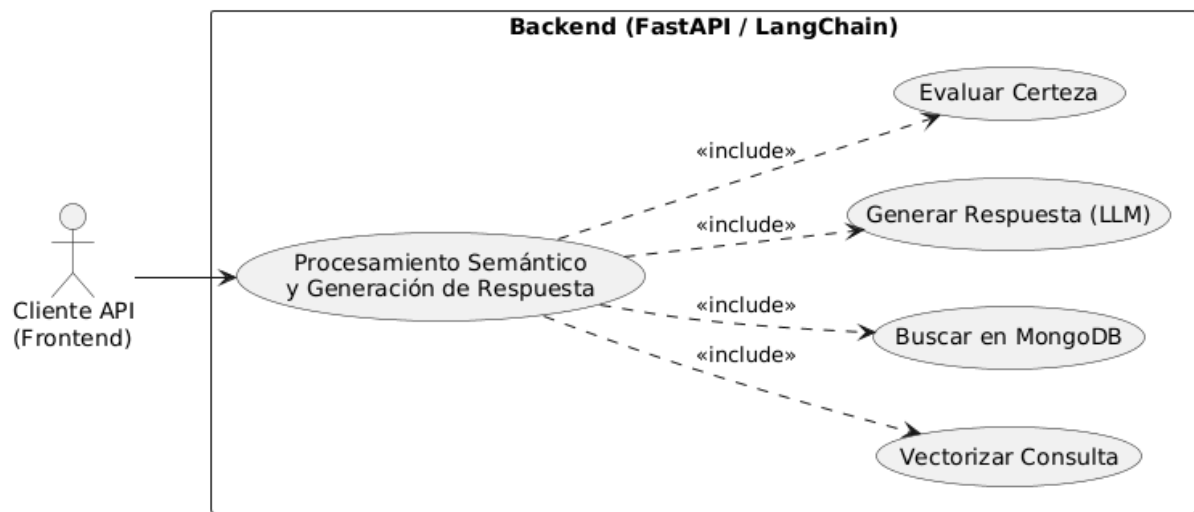


Ilustración 3. Diagrama de Casos de Uso: Procesamiento RAG y generación de respuesta

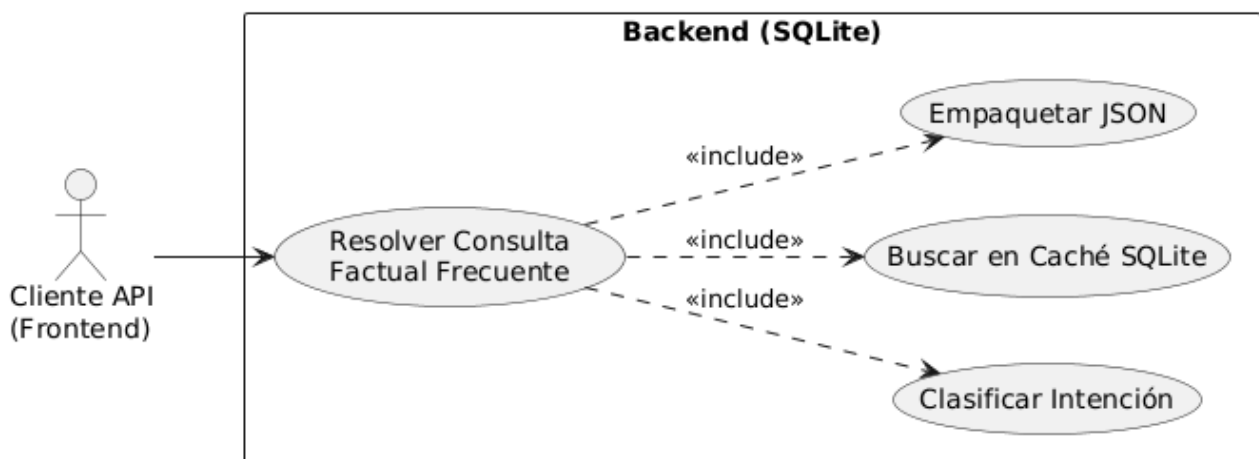


Ilustración 4. Diagrama de Casos de Uso: Gestión de caché y consultas factuales

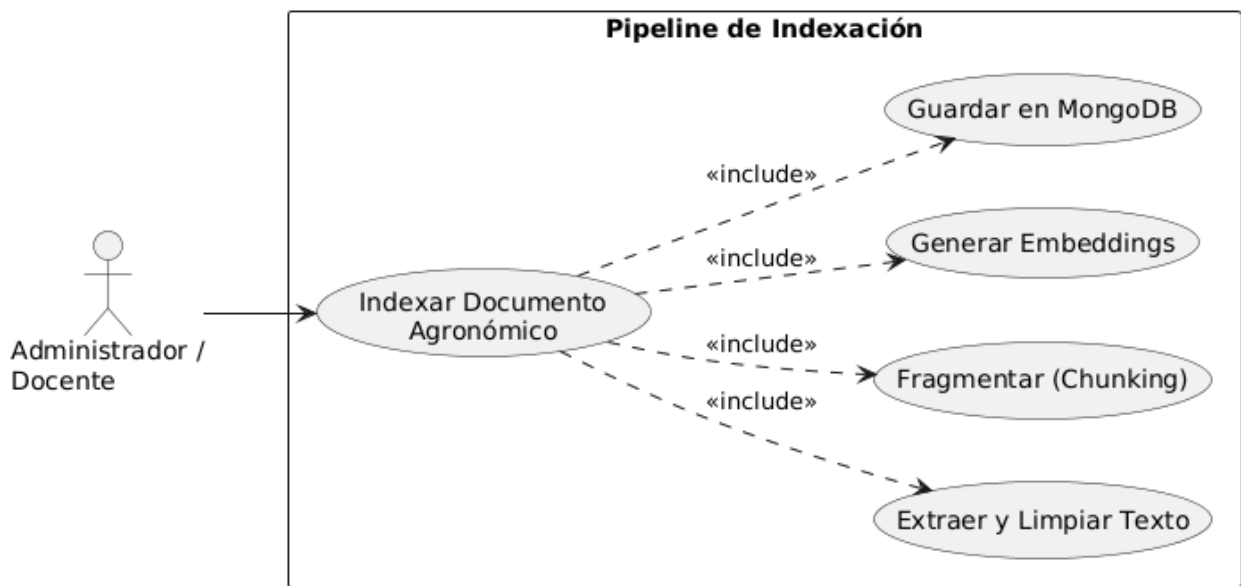


Ilustración 5. Diagrama de Casos de Uso: Actualización de la base de conocimiento

Anexo D: Evidencia de Ejecución del Modelo de Inteligencia Artificial (Ollama)

```

Windows PowerShell
PS C:\Users\Usuario> ollama serve
time=2026-07-18T17:51:25.678-05:00 level=INFO source=routes.go:1554 msg="server config" env="map[CUDA_VISIBLE_DEVICES: G
GML_VK_VISIBLE_DEVICES: GPU_DEVICE_ORDINAL: HIP_VISIBLE_DEVICES: HSA_OVERRIDE_GFX_VERSION: HTTPS_PROXY: HTTP_PROXY: NO_P
ROXY: OLLAMA_CONTEXT_LENGTH:4096 OLLAMA_DEBUG:INFO OLLAMA_FLASH_ATTENTION:false OLLAMA_GPU_OVERHEAD:0 OLLAMA_HOST:http:/
/127.0.0.1:11434 OLLAMA_KEEP_ALIVE:5m0s OLLAMA_KV_CACHE_TYPE: OLLAMA_LLM_LIBRARY: OLLAMA_LOAD_TIMEOUT:5m0s OLLAMA_MAX_LO
ADED_MODELS:0 OLLAMA_MAX_QUEUE:512 OLLAMA_MODELS:C:\\Users\\Usuario\\ollama\\models OLLAMA_MULTUSER_CACHE:false OLLAMA
_NEW_ENGINE:false OLLAMA_NOHISTORY:false OLLAMA_NOPRUNE:false OLLAMA_NUM_PARALLEL:1 OLLAMA_ORIGINS:[http://localhost htt
ps://localhost http://localhost:* https://localhost:* http://127.0.0.1 https://127.0.0.1 http://127.0.0.1:* https://127.
0.0.1:* http://0.0.0.0 https://0.0.0.0 http://0.0.0.0:* https://0.0.0.0:* app://* tauri://* vscode-webview://*
vscode-file://*] OLLAMA_REMOTES:[ollama.com] OLLAMA_SCHED_SPREAD:false OLLAMA_VULKAN:false ROCR_VISIBLE_DEVICES:]"
time=2026-07-18T17:51:25.877-05:00 level=INFO source=images.go:522 msg="total blobs: 11"
time=2026-07-18T17:51:25.879-05:00 level=INFO source=images.go:529 msg="total unused blobs removed: 0"
time=2026-07-18T17:51:25.880-05:00 level=INFO source=routes.go:1607 msg="Listening on 127.0.0.1:11434 (version 0.13.4)"
time=2026-07-18T17:51:25.885-05:00 level=INFO source=runner.go:67 msg="discovering available GPUs..."
time=2026-07-18T17:51:25.910-05:00 level=INFO source=server.go:429 msg="starting runner" cmd="C:\\Users\\Usuario\\AppDat
a\\Local\\Programs\\Ollama\\ollama.exe runner --ollama-engine --port 55329"
time=2026-07-18T17:52:06.529-05:00 level=INFO source=server.go:429 msg="starting runner" cmd="C:\\Users\\Usuario\\AppDat
a\\Local\\Programs\\Ollama\\ollama.exe runner --ollama-engine --port 55345"
time=2026-07-18T17:52:19.270-05:00 level=INFO source=server.go:429 msg="starting runner" cmd="C:\\Users\\Usuario\\AppDat
a\\Local\\Programs\\Ollama\\ollama.exe runner --ollama-engine --port 55358"
time=2026-07-18T17:52:32.693-05:00 level=INFO source=runner.go:106 msg="experimental Vulkan support disabled. To enable
, set OLLAMA_VULKAN=1"
time=2026-07-18T17:52:32.704-05:00 level=INFO source=server.go:429 msg="starting runner" cmd="C:\\Users\\Usuario\\AppDat
a\\Local\\Programs\\Ollama\\ollama.exe runner --ollama-engine --port 55372"
time=2026-07-18T17:52:32.708-05:00 level=INFO source=server.go:429 msg="starting runner" cmd="C:\\Users\\Usuario\\AppDat
a\\Local\\Programs\\Ollama\\ollama.exe runner --ollama-engine --port 55373"
time=2026-07-18T17:52:32.708-05:00 level=INFO source=server.go:429 msg="starting runner" cmd="C:\\Users\\Usuario\\AppDat
a\\Local\\Programs\\Ollama\\ollama.exe runner --ollama-engine --port 55374"
time=2026-07-18T17:52:33.337-05:00 level=INFO source=types.go:42 msg="inference compute" id=GPU-062e5b4d-0b2f-8fe0-8e27-
fa7fe0015170 filter_id="" library=CUDA compute=8.9 name=CUDA0 description="NVIDIA GeForce RTX 4050 Laptop GPU" libdirs=o
  
```

Ilustración 6. Evidencia de ejecución del modelo Llama 3.1 en servidor local (Ollama)

Anexo E: Base de Datos Vectorial (ChromaDB) y Generación Aumentada por Recuperación (RAG)

```
# 3. CREAR MEMORIA
print("🧠 Procesando conocimiento (esto tardará unos segundos)...")
embeddings = HuggingFaceEmbeddings(model_name="sentence-transformers/all-MiniLM-L6-v2")
vectorstore = Chroma.from_documents(documents=splits, embedding=embeddings, persist_directory="./chroma_db")
retriever = vectorstore.as_retriever(search_kwargs={"k": 3})
```

Ilustración 7. Código de inicialización de la base de datos vectorial ChromaDB

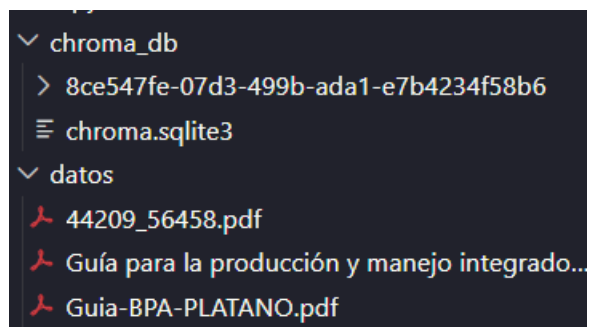


Ilustración 8. Estructura de carpetas evidenciando la persistencia local de ChromaDB

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

python + ▾ □ □ ... | ↺ ×

○ PS C:\Chatbot_Platano> python main.py

2026-07-18 18:53:08.787440: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF\_ENABLE\_ONEDNN\_OPTS=0`.

2026-07-18 18:53:13.202303: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF\_ENABLE\_ONEDNN\_OPTS=0`.

WARNING:tensorflow:From C:\Users\Usuario\AppData\Local\Programs\Python\Python312\Lib\site-packages\tf_keras\src\losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.

🚀 Iniciando configuración del servidor...

📁 Buscando manuales en carpeta 'datos'...

✅ Se encontraron 115 páginas de información.

🌀 Procesando conocimiento (esto tardará unos segundos)...

C:\Chatbot_Platano\main.py:50: LangChainDeprecationWarning: The class `HuggingFaceEmbeddings` was deprecated in LangChain 0.2.2 and will be removed in 1.0. An updated version of the class exists in the `langchain-huggingface` package and should be used instead. To use it run `pip install -U `langchain-huggingface` and import as `from langchain_huggingface import HuggingFaceEmbeddings`.

embeddings = HuggingFaceEmbeddings(model_name="sentence-transformers/all-MiniLM-L6-v2")

🔥 ¡CEREBRO LISTO Y OPERATIVO!

INFO: Started server process [5400]

INFO: Waiting for application startup.

INFO: Application startup complete.

INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)

Ilustración 9. Terminal de ejecución confirmando la carga de manuales y activación del pipeline

Anexo F: Documentación de la API REST (FastAPI)

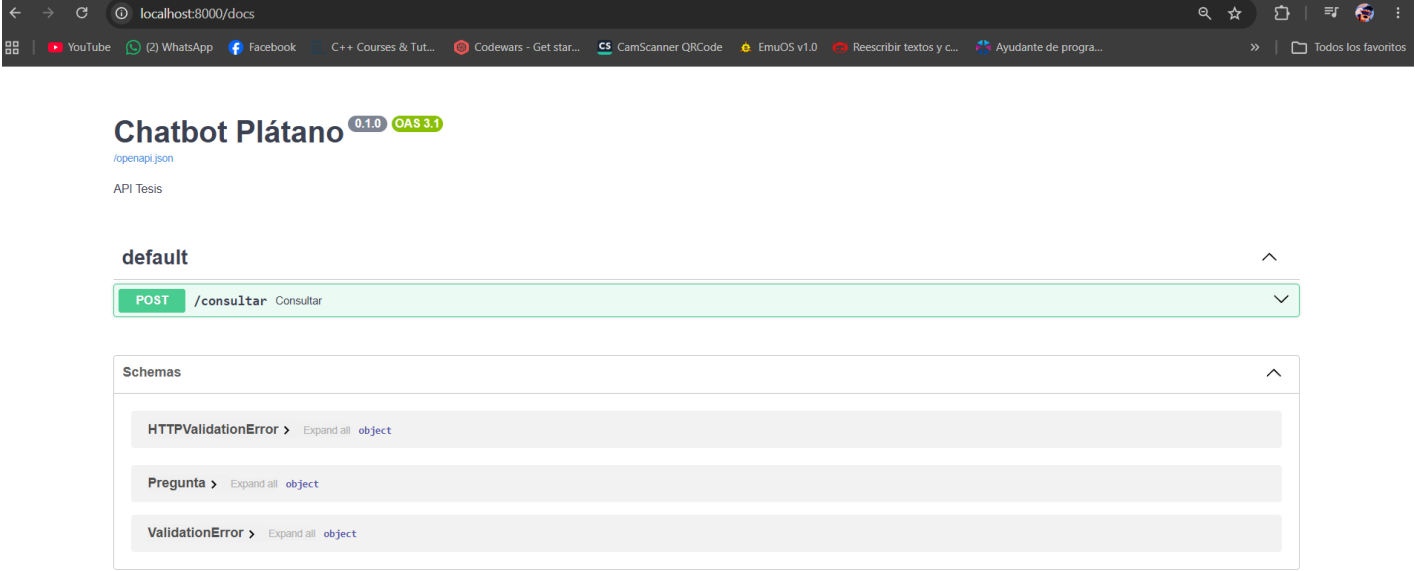


Ilustración 10. Interfaz de documentación automática Swagger UI de la API REST

Anexo G: Repositorio de Código Fuente

<https://github.com/andresvera1103-oss/agrobot-tesis-platano>

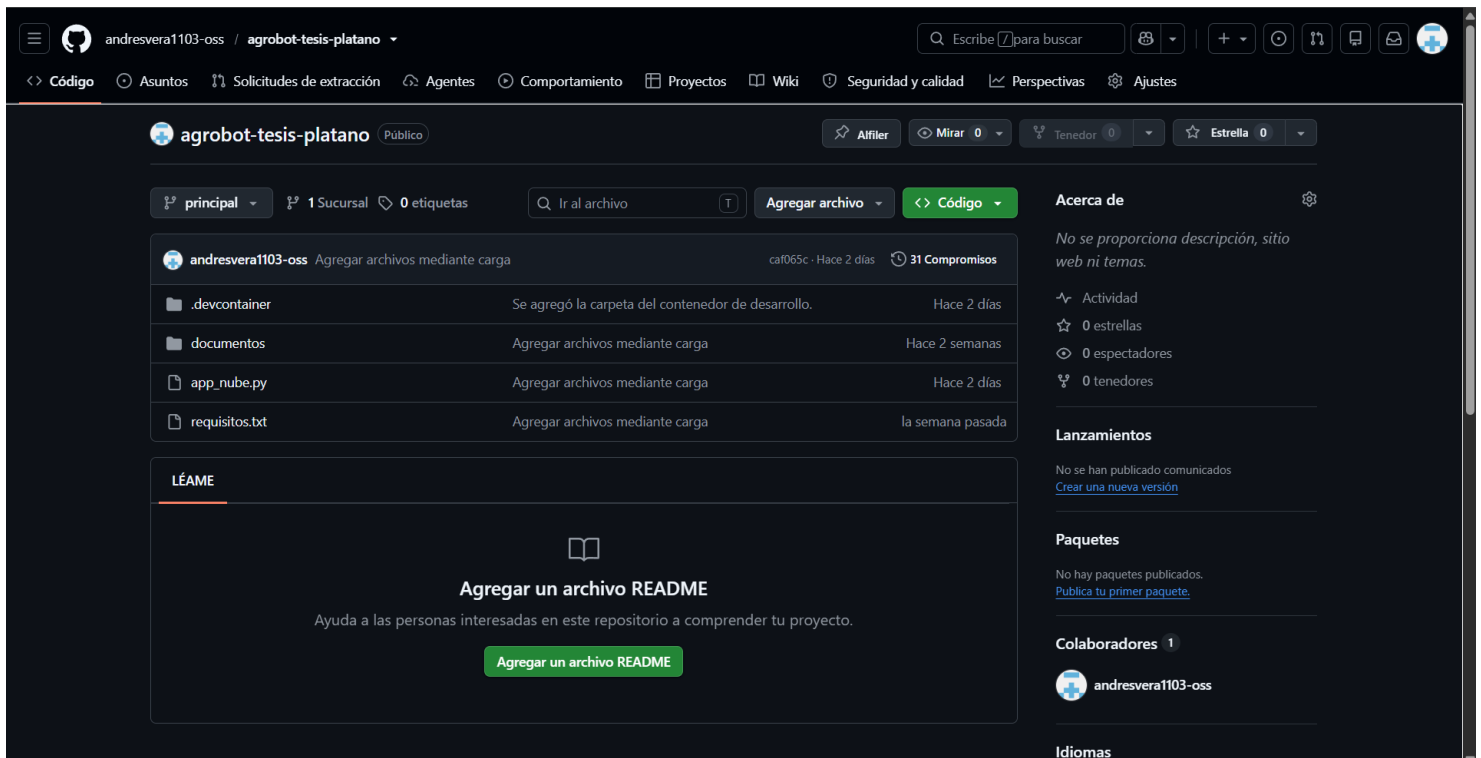


Ilustración 11. Repositorio en GitHub con el código fuente del proyecto

Anexo H: Evidencia del chatbot funcionando

En la laptop

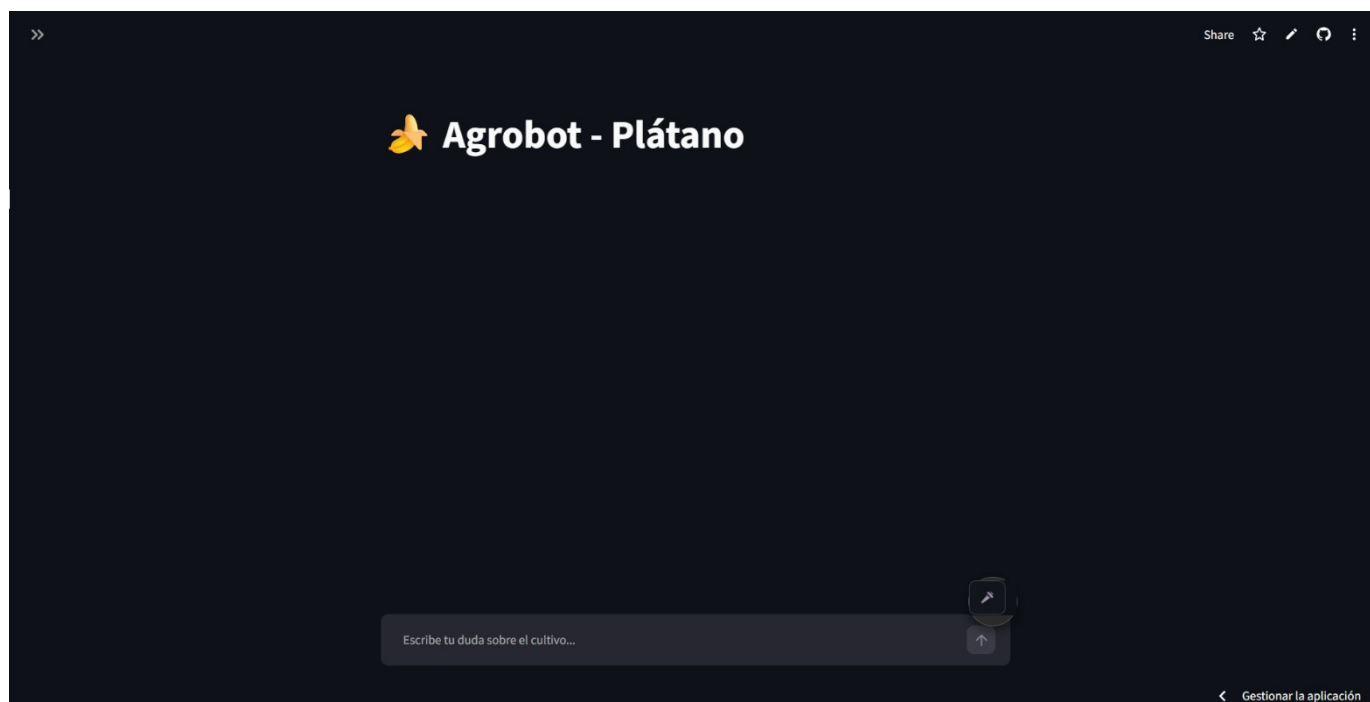


Ilustración 12. Interfaz web del chatbot interactuando en entorno de escritorio

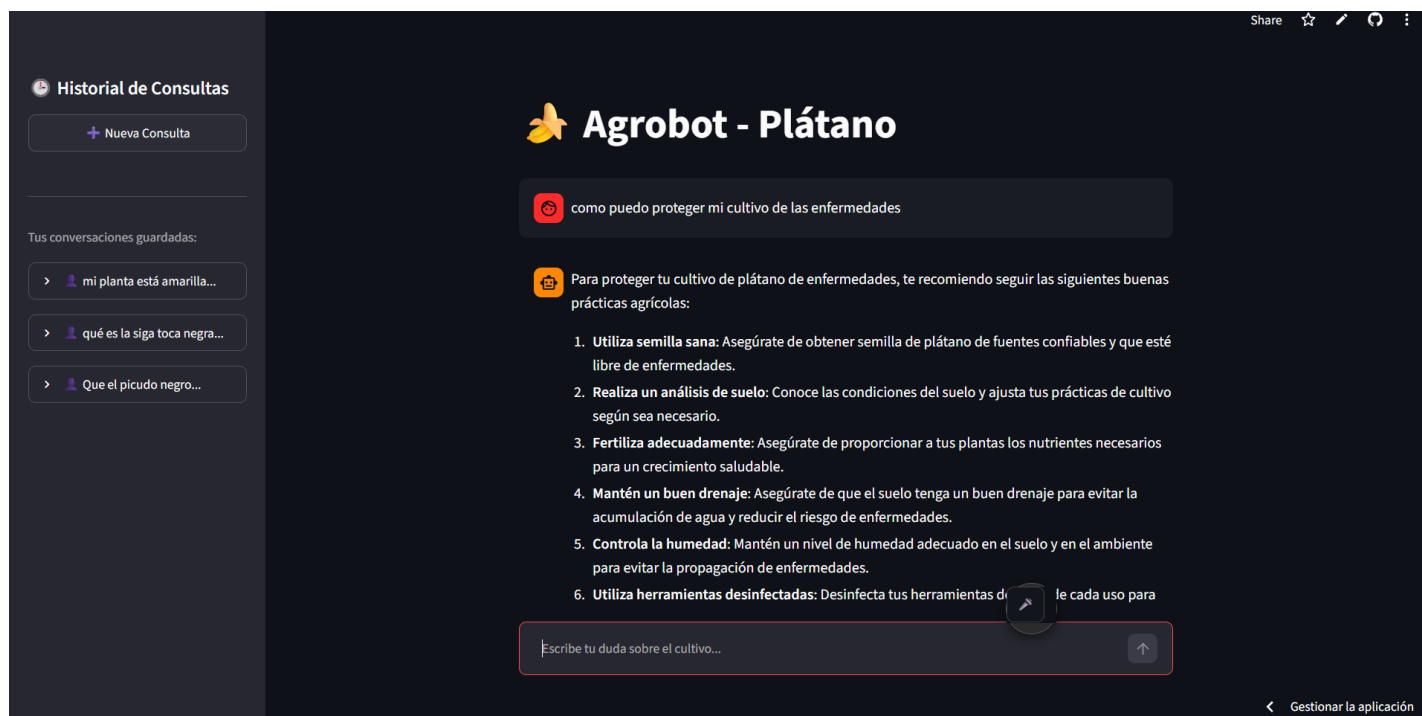


Ilustración 13. Historial de conversaciones y visualización de respuestas del sistema

En el celular

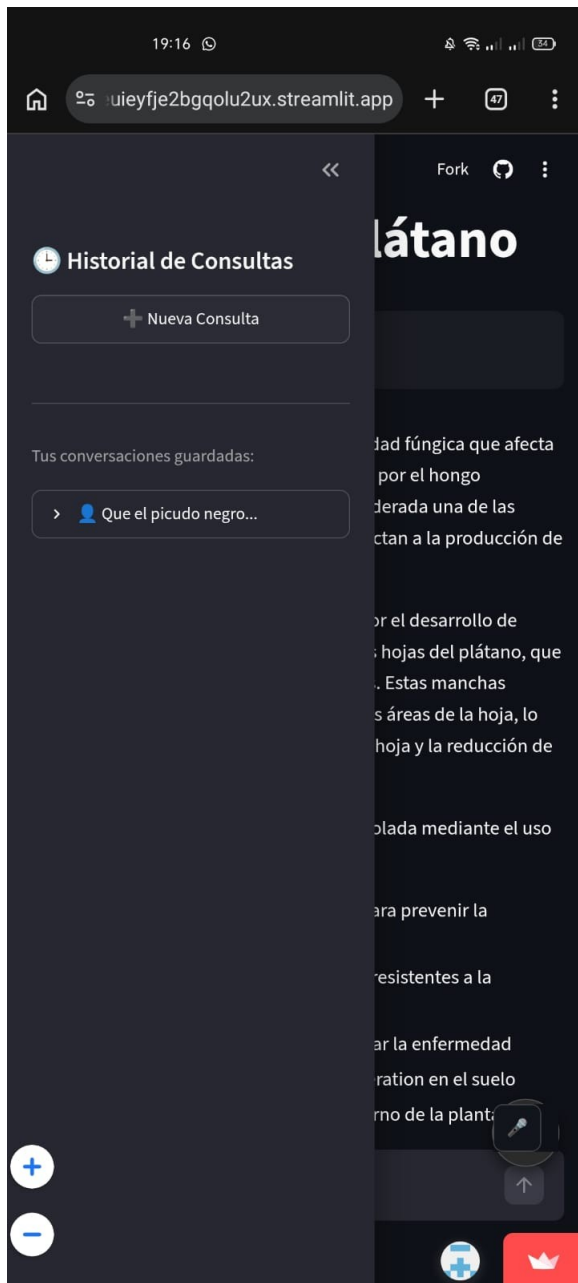


Ilustración 14. Interfaz web del chatbot adaptada a la vista de dispositivos móviles

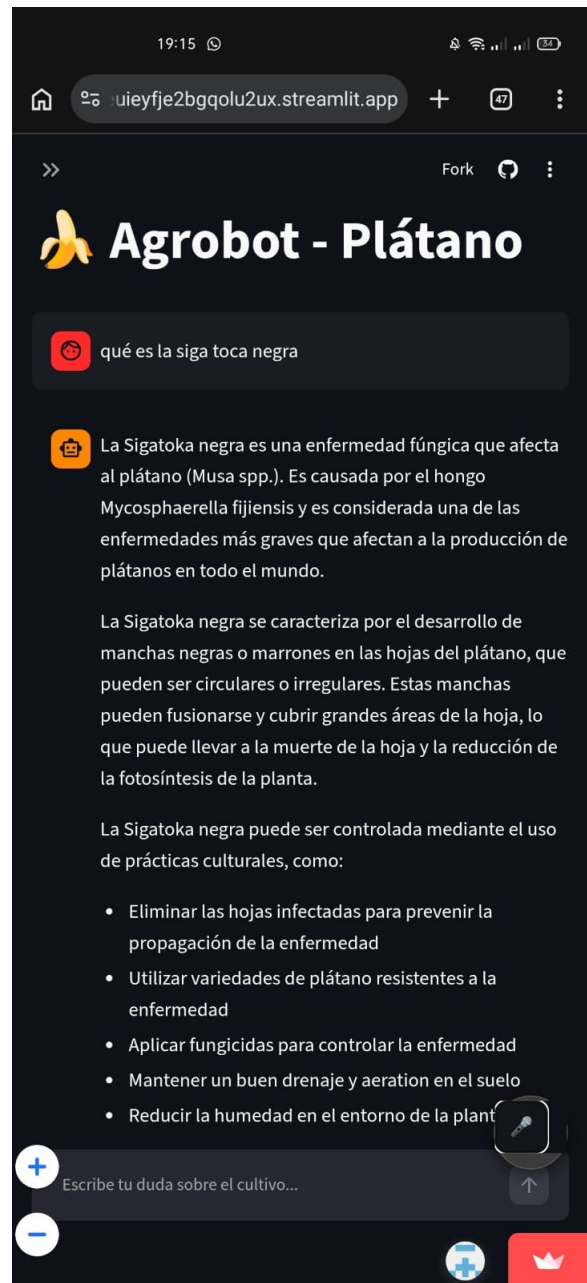


Ilustración 15. Visualización del menú lateral en dispositivos móviles

Anexo I: Fotos realizando las encuestas y entrevista



Ilustración 16. Entrevistando al docente



Ilustración 17. Estudiantes respondiendo las encuestas

9 GLOSARIO

API (Interfaz de Programación de Aplicaciones): Conjunto de reglas y protocolos que permite a diferentes aplicaciones de software comunicarse entre sí, facilitando el intercambio de datos.

Backend: Capa de acceso a datos y lógica de negocio de un software, que opera en el servidor y no es directamente visible para el usuario final.

Caché: Capa de almacenamiento de datos de alta velocidad que guarda un subconjunto de datos, típicamente transitorios, para que las solicitudes futuras se sirvan más rápidamente.

ChromaDB: Base de datos vectorial de código abierto diseñada para almacenar y recuperar de manera eficiente *embeddings* generados por modelos de inteligencia artificial.

Chunking (Fragmentación): Proceso de dividir textos extensos en segmentos o bloques más pequeños para facilitar su procesamiento y vectorización por parte de modelos de inteligencia artificial.

Embeddings: Representaciones matemáticas (vectores de números reales) de palabras, oraciones o documentos que capturan su significado semántico para ser procesados por algoritmos.

Endpoint: URL o ruta específica dentro de una API mediante la cual un cliente puede acceder a un recurso o ejecutar una función en el servidor.

FastAPI: *Framework* web moderno y de alto rendimiento para construir APIs con Python, basado en las anotaciones de tipo estándar de Python.

Framework (Marco de Trabajo): Estructura de software que proporciona una base conceptual y tecnológica predefinida para facilitar el desarrollo de aplicaciones.

Frontend: Capa de presentación de un software, que comprende la interfaz gráfica y los elementos interactivos con los que el usuario final interactúa directamente.

Git: Sistema de control de versiones distribuido que rastrea los cambios en el código fuente durante el desarrollo de software.

JSON (JavaScript Object Notation): Formato ligero de intercambio de datos, fácil de leer y escribir para los humanos, y fácil de analizar y generar para las máquinas.

LangChain: *Framework* de código abierto diseñado para facilitar la creación de aplicaciones impulsadas por Modelos de Lenguaje de Gran Escala (LLM), permitiendo encadenar diferentes componentes lógicos.

Latencia: Tiempo de retardo entre el envío de una petición o instrucción de datos y la recepción de la respuesta correspondiente en un sistema informático.

LLM (Large Language Model / Modelo de Lenguaje de Gran Escala): Modelo de inteligencia artificial entrenado con cantidades masivas de texto, capaz de comprender, generar y traducir lenguaje natural con alta coherencia.

Modelo Vectorial: Representación algebraica de textos u objetos matemáticos en un espacio multidimensional utilizado para operaciones de recuperación de información.

Ollama: Herramienta de software que permite ejecutar Modelos de Lenguaje de Gran Escala (LLM) de manera local y privada, optimizando el consumo de recursos de hardware.

Pipeline (Tubería de datos): Secuencia de procesos o instrucciones algorítmicas donde la salida de un elemento es la entrada del siguiente, utilizada frecuentemente en procesamiento de datos y *Machine Learning*.

PLN (Procesamiento de Lenguaje Natural): Rama de la inteligencia artificial que ayuda a las computadoras a entender, interpretar y manipular el lenguaje humano.

Prompt: Instrucción, pregunta o texto inicial que se proporciona a un modelo de inteligencia artificial para guiar y condicionar su respuesta.

Prompt Engineering (Ingeniería de Instrucciones): Disciplina enfocada en el diseño, optimización y estructuración de *prompts* para maximizar la precisión y eficacia de los modelos de lenguaje.