



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

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

CHAT BOT CON APRENDIZAJE AUTOMÁTICO SUPERVISADO
PARA LA EFICIENCIA DEL RIEGO DE THEOBROMA CACAO EN
EL VIVERO LASTENIA ULEAM EL CARMEN

RODRÍGUEZ ORDOÑEZ KEVIN DAVID
AUTOR

ARÉVALO HERMIDA RÓMULO DANILO
TUTOR

EL CARMEN, AGOSTO 2026



CERTIFICACIÓN DEL TUTOR

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

CERTIFICACIÓN

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

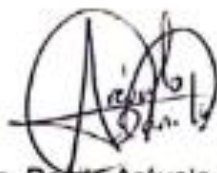
Haber dirigido, revisado y aprobado preliminarmente el Trabajo de Integración Curricular bajo la autoría del estudiante **RODRÍGUEZ ORDOÑEZ KEVIN DAVID**, legalmente matriculado en la carrera de Software, período académico 2026-2027, cumpliendo el total de 384 horas, cuyo tema del proyecto es **"CHAT BOT CON APRENDIZAJE AUTOMÁTICO SUPERVISADO PARA LA EFICIENCIA DEL RIEGO DE THEOBROMA CACAO EN EL VIVERO LASTENIA 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, 27 de julio de 2026.

Lo certifico,



Ing. Danilo Arévalo Hermida
Docente Tutor
Área: Software

TRIBUNAL DE SUSTENTACIÓN



Universidad Laica Eloy Alfaro de Manabí
Extensión El Carmen
Carrera de Ingeniería de Software

TRIBUNAL DE SUSTENTACIÓN

Título de Trabajo de Titulación: Chat Bot con Aprendizaje Automático Supervisado para la Eficiencia del riego de Theobroma Cacao en el Vivero Lastenia Uleam El Carmen.

Modalidad: Proyecto integrador

Autor: Rodríguez Ordoñez Kevin David

Tutor: Ing. Arévalo Hermida Rómulo Danilo.

Tribunal de sustentación:

- Presidente: Ing. Minaya Macias Renelmo Wladimir, Mg.

- Miembro: Ing. Reascos Pinchao Raúl Saed, Mg.

- Miembro: Ing. Carlos López Rodríguez, Mg.

Fecha de sustentación: 26 de agosto de 2026

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



DECLARACIÓN DE AUTORÍA

La responsabilidad del contenido de este Trabajo de titulación, cuyo tema es: Chatbot con aprendizaje automático supervisado para la eficiencia del riego de theobroma cacao en el vivero Lastenia Uleam El Carmen corresponde exclusivamente a: Rodríguez Ordoñez Kevin David con CI. 2300778780, y los derechos patrimoniales de la misma corresponden a la Universidad Laica “Eloy Alfaro” de Manabí.

A handwritten signature in blue ink, which appears to read "Kevin Rodríguez". The signature is written over a horizontal line.

Rodríguez Ordoñez Kevin David

C.I. 2300778780

DEDICATORIA

Dedico este proyecto de titulación, fruto de incontables horas de esfuerzo y sacrificio, a mis padres, quienes con su trabajo incondicional y ejemplo diario me enseñaron el verdadero valor de la perseverancia, convirtiéndose en el pilar fundamental que me mantuvo firme durante todos estos años. A mi familia entera, por ser mi refugio seguro y brindarme su apoyo moral y palabras de aliento en cada desafío de mi etapa universitaria. De manera muy especial, dedico este gran logro a mi pareja, por su inquebrantable compañía, su infinita paciencia y por ser esa fuente de motivación indispensable para no rendirme, especialmente en aquellas largas jornadas donde el agotamiento de combinar las exigentes responsabilidades laborales diarias con el desarrollo de este sistema de software parecía superarme. A mis docentes, por guiarme con genuina vocación, y a mis amigos, por hacer de este camino una experiencia invaluable. Este sueño profesional es tanto mío como de cada uno de ustedes.

Kevin David Rodríguez Ordoñez

AGRADECIMIENTO

Expreso mi más sincero agradecimiento a la Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen, por haberme brindado una formación académica de excelencia. A mis docentes de la carrera de Ingeniería de Software, quienes compartieron sus conocimientos y me dotaron de las herramientas técnicas necesarias para convertirme en un profesional capacitado. Un agradecimiento especial a mi tutor, el Ing. Rómulo Danilo Arévalo Hermida, por su guía constante, exigencia y valiosas observaciones, las cuales fueron fundamentales para estructurar y culminar con éxito este trabajo de titulación. Asimismo, extendiendo mi gratitud al Vivero Lastenia por abrirme sus puertas y permitirme implementar este proyecto tecnológico en sus instalaciones. Finalmente, agradezco a todos aquellos colegas y compañeros que, de manera directa o indirecta, me apoyaron y motivaron para sacar adelante este sistema, demostrando que con disciplina es posible equilibrar el arduo trabajo diario con la excelencia académica.

Kevin David Rodríguez Ordoñez

ÍNDICE GENERAL

PORTADA.....	I
CERTIFICACIÓN DEL TUTOR.....	III
TRIBUNAL DE SUSTENTACIÓN.....	¡Error! Marcador no definido.
DECLARACIÓN DE AUTORÍA.....	V
DEDICATORIA	VI
AGRADECIMIENTO	VII
ÍNDICE GENERAL	VIII
ÍNDICE DE TABLAS	XVIII
ÍNDICE DE ILUSTRACIONES	XX
ÍNDICE DE ANEXOS	XXII
RESUMEN	XXIII
ABSTRACT.....	XXIV
CAPÍTULO I	1
1 INTRODUCCIÓN	1
1.1 Introducción	1
1.2 Presentación del tema.....	2
1.3 Ubicación y contextualización de la problemática.....	2
1.4 Planteamiento del problema	3
1.4.1 Problematización.....	3
1.4.2 Génesis del problema	4

1.4.3	Estado actual del problema	5
1.5	Diagrama causa – efecto del problema	6
1.6	Objetivos	7
1.6.1	Objetivo general.....	7
1.6.2	Objetivos específicos	7
1.7	Justificación.....	7
1.8	Impactos esperados	8
1.8.1	Impacto tecnológico.....	8
1.8.2	Impacto social	9
1.8.3	Impacto ecológico	9
CAPÍTULO II		11
2. MARCO TEÓRICO.....		11
2.1	Antecedentes históricos.....	11
2.1.1	Historia de los Chatbots y el Aprendizaje Automático Supervisado	11
2.1.2	Historia de la Gestión del Riego	12
2.1.3	Historia de Chatbots para Riego	13
2.2	Antecedentes de investigaciones relacionadas al tema presentado	13
2.2.1	Análisis de investigaciones similares.....	14
2.3	Definiciones conceptuales.....	15
2.3.1	Chatbot con Aprendizaje Automático Supervisado	15
2.3.1.1	Definición y Evolución Histórica de Chatbots	15

2.3.1.2	Arquitectura de Sistemas Conversacionales (Componentes)	16
2.3.1.3	Procesamiento del Lenguaje Natural (NLP).....	17
2.3.1.4	Aprendizaje Automático Supervisado (Enfoque Teórico)	18
2.3.1.5	Clasificación de Chatbots por Nivel de Inteligencia (Reglas vs. ML)	19
2.3.1.6	Plataformas y Canales de Comunicación (Interfaces).....	20
2.3.1.7	Arquitectura IoT y Flujo de Datos del Sistema	21
2.3.1.8	Métricas de Evaluación de Rendimiento de Chatbots (Accuracy, F1-Score) 21	
2.3.2	Eficiencia del riego de Theobroma Cacao.	22
2.3.2.1	Definición y Fundamentos del Riego Inteligente	23
2.3.2.2	Sensores y Adquisición de Datos Agronómicos (Humedad, Temperatura) 24	
2.3.2.3	Actuadores en Sistemas de Riego (Válvulas Solenoides, Bombas)	25
2.3.2.4	Plataformas de Servicios en la Nube para IoT (AWS, Azure, Google Cloud) 26	
2.3.2.5	Técnicas de Programación de Riego (Umbral, Basado en Evapotranspiración)	27
2.3.2.6	Eficiencia y Conservación del Recurso Hídrico	28
2.3.2.7	Retos de la Implementación de IoT en la Agricultura.....	29
2.3.2.8	Cultivo de Theobroma Cacao y sus Requerimientos Hídricos.....	30
2.3.3	Metodología de desarrollo: Scrum.....	30
2.4	Conclusiones del marco teórico	32
CAPÍTULO III.....		33

3. MARCO INVESTIGATIVO	33
3.1 Introducción	33
3.2 Tipos de investigación.....	34
3.2.1 Investigación de campo.....	34
3.2.2 Investigación descriptiva	34
3.2.3 Investigación aplicada.....	35
3.2.4 Métodos de investigación	35
3.2.4.1 Método inductivo-deductivo.....	35
3.2.4.2 Método analítico-sintético	36
3.3 Fuentes de información de datos	37
3.3.1 Fuentes primarias	37
3.3.1.1 Encuestas	37
3.3.1.2 Entrevista	37
3.4 Estrategia operacional para la recolección de datos.....	38
3.4.1 Población.....	38
3.4.2 Muestra	38
3.4.3 Análisis de las herramientas de recolección de datos a utilizar	39
3.4.3.1 Encuesta.....	39
3.4.3.2 Entrevista	39
3.4.3.3 Estructura de los instrumentos de recolección de datos aplicados	40
3.4.3.3.1 Estructura de encuesta.....	40
3.4.3.3.2 Estructura de entrevista	40

3.4.4	Plan de recolección de datos	41
3.5	Análisis y presentación de resultados.....	42
3.5.1	Presentación y descripción de los resultados obtenidos	42
3.5.1.1	Análisis de encuesta dirigida a estudiantes Agropecuaria de la Uleam extensión El Carmen	42
3.5.1.2	Análisis de entrevista dirigida a Coordinador de Vivero Lastenia Vera	45
3.5.2	Informe final del análisis de los datos.....	48
CAPÍTULO IV.....		50
4.	MARCO PROPOSITIVO	50
4.1	Introducción	50
4.2	Descripción de la propuesta	51
4.3	Determinación de recursos	51
4.3.1	Humanos	51
4.3.2	Tecnológicos	52
4.3.3	Económicos.....	53
4.4	Desarrollo mediante metodología Scrum.....	53
4.4.1	Descripción del Producto	54
4.4.1.1	Propósito del Producto.....	54
4.4.1.2	Funcionalidades Clave.....	54
4.4.1.3	Usuarios Objetivo	54
4.4.1.4	Condiciones de Éxito del Producto	55
4.4.2	Requerimientos del sistema	55

4.4.2.1	Historias de Usuario	55
4.4.2.1.1	Historia de Usuario 1: Consulta del estado del cultivo.....	55
4.4.2.1.2	Historia de Usuario 2: Alerta predictiva automática.....	56
4.4.2.1.3	Historia de Usuario 3: Alerta predictiva automática.....	56
4.4.2.1.4	Historia de Usuario 4: Configuración de umbrales.....	57
4.4.2.1.5	Historia de Usuario 5: Historial de datos.....	57
4.4.2.2	Requerimientos funcionales	58
4.4.2.3	Requerimientos no funcionales	58
4.4.3	Diseño del Sistema.....	59
4.4.3.1	Casos de Uso	59
4.4.3.1.1	Caso de uso: captura y envío de telemetría	59
4.4.3.1.2	Caso de uso: Predicción y generación de alertas	60
4.4.3.1.3	Caso de uso: Salida de información.....	60
4.4.3.2	Diagramas de Secuencia.....	61
4.4.3.2.1	Diagrama de Secuencia: Captura y almacenamiento de telemetría	61
4.4.3.2.2	Diagrama de Secuencia: Predicción de estrés hídrico y evaluación de riesgos	62
4.4.3.2.3	Diagrama de Secuencia: Generación y descarga de reportes	63
4.4.3.3	Diagramas de Estado	64
4.4.3.3.1	Diagrama de Estado: Ciclo de vida de la pantalla principal (Dashboard).....	64
4.4.3.3.2	Diagrama de Estado: Evaluación de riesgo ambiental	65
4.4.3.3.3	Diagrama de Estado: Petición HTTP y envío de mensajes.....	66
4.4.3.4	Diagrama de Base de Datos.....	67
4.4.4	Arquitectura del Sistema.....	67
4.4.4.1	Arquitectura general	67
4.4.4.2	Tecnologías utilizadas	69
4.4.4.3	Roles Scrum.....	70
4.4.5	Definición de Hecho (DoD).....	71

4.4.5.1	Criterios Generales	71
4.4.5.2	Criterios Específicos del Proyecto.....	71
4.4.6	Product Backlog.....	72
4.4.7	Sprint 1: Configuración del Backend y Motor Conversacional Básico.....	73
4.4.7.1	Planificación del Sprint	73
4.4.7.2	Sprint Backlog	74
4.4.7.3	Desarrollo del Sprint.....	75
4.4.7.3.1	Funcionalidad de Consulta Climática Mediante Lenguaje Natural	75
4.4.7.4	Sprint Review	76
4.4.7.4.1	Descripción de la funcionalidad	76
4.4.7.5	Pruebas del Sprint.....	78
4.4.7.5.1	Pruebas de Caja Negra (Validación de Usabilidad)	78
4.4.7.5.2	Pruebas de Caja Blanca (Lógica de Estados UI)	79
4.4.7.6	Incremento y Entregables del Sprint	79
4.4.7.7	Estado actual del Product Backlog	79
4.4.8	Sprint 2: Entrenamiento del Motor NLP y Desarrollo del Backend.....	80
4.4.8.1	Planificación del Sprint	81
4.4.8.2	Sprint Backlog	82
4.4.8.3	Desarrollo del Sprint.....	82
4.4.8.3.1	Funcionalidad de Validación de Rangos Físicos y Control de Errores	82
4.4.8.3.2	Funcionalidad de Muestreo Múltiple y Filtrado por Media Recortada.....	83
4.4.8.4	Sprint Review	84
4.4.8.4.1	Retroalimentación recibida:	85
4.4.8.4.2	Mejoras Realizadas:.....	85
4.4.8.5	Pruebas del Sprint.....	86

4.4.8.5.1	Pruebas de Caja Negra (Validación de Filtrado y Rangos Físicos)	86
4.4.8.5.2	Pruebas de Caja Blanca (Lógica interna de Procesamiento)	86
4.4.8.6	Incremento y Entregables del Sprint	87
4.4.8.7	Estado actual del Product Backlog	87
4.4.9	Sprint 3: Integración del Panel Lateral (Drawer) y Gestión de Historial	88
4.4.9.1	Planificación del Sprint	88
4.4.9.2	Sprint Backlog	89
4.4.9.3	Desarrollo del Sprint.....	89
4.4.9.4	Sprint Review	90
4.4.9.4.1	Retroalimentación recibida	91
4.4.9.4.2	Solución aplicada:.....	91
4.4.9.5	Pruebas del Sprint.....	92
4.4.9.5.1	Pruebas de Caja Negra	92
4.4.9.5.2	Pruebas de Caja Blanca.....	92
4.4.9.6	Incremento y Entregables del Sprint	93
4.4.9.7	Estado actual del Product Backlog	93
4.4.10	Sprint 4: Despliegue en Campo, Pruebas de Usabilidad y Empaquetado Final 94	
4.4.10.1.1	Planificación del Sprint	94
4.4.10.2	Sprint Backlog	95
4.4.10.3	Desarrollo del Sprint.....	96
4.4.10.3.1	Integración del Flujo Conversacional (Pipeline)	96
4.4.10.3.2	Pruebas de Funcionamiento y Empaquetado Final (Release).....	98
4.4.10.4	Sprint Review	99
4.4.10.4.1	Retroalimentacion recibida	99
4.4.10.4.2	Mejoras realizadas	100

4.4.10.5	Pruebas del Sprint	100
4.4.10.5.1	Pruebas de Caja Negra (Validación de Usabilidad e Inferencia)	100
4.4.10.5.2	Pruebas de Caja Blanca (Optimización de recursos)	101
4.4.10.6	Incremento y Entregables del Sprint	102
4.4.10.7	Estado actual del Product Backlog	103
4.4.11	Resultado Final del Producto	104
4.4.11.1	Funcionalidades implementadas.....	104
4.4.11.2	Evidencia final del sistema	105
4.4.11.3	Conclusiones del desarrollo.....	107
CAPÍTULO V		108
5. EVALUACIÓN DE RESULTADOS		108
5.1	Introducción	108
5.2	Presentación y monitoreo de resultados	109
5.2.1	Planificación de la evaluación	109
5.2.2	Ejecución del monitoreo	110
5.2.2.1	Proceso 1 Evaluación del Aprendizaje Automático Supervisado (Motor NLP del Chatbot).....	110
5.2.2.1.1	Descripción	110
5.2.2.1.2	Toma de datos	111
5.2.2.2	Proceso 2 Predicción de Estrés Hídrico (Modelo Machine Learning)	113
5.2.2.3	Descripción	113
5.2.2.3.1	Toma de datos	114
5.2.2.4	Proceso 3 Gestión y Recuperación del Historial de Conversaciones	115
5.2.2.5	Descripción	115

5.2.2.5.1 Toma de datos	115
5.3 Interpretación objetiva.....	116
CAPÍTULO VI.....	118
6. CONCLUSIONES Y RECOMENDACIONES	118
6.1 Conclusiones	118
6.2 Recomendaciones.....	119
BIBLIOGRAFÍA	120
ANEXOS	127
GLOSARIO	141

ÍNDICE DE TABLAS

Tabla 1 Plan de recolección de datos	41
Tabla 2 Resultados obtenidos en la encuesta a estudiantes de Agropecuaria.....	42
Tabla 3 Análisis de entrevista a coordinador	45
Tabla 4 Recursos humanos	51
Tabla 5 Recursos tecnológicos.....	52
Tabla 6 Recursos económicos.....	53
Tabla 7 historia de usuario 1	55
Tabla 8 Historia de usuario 2	56
Tabla 9 Historia de usuario 3.....	56
Tabla 10 Historia de usuario 4	57
Tabla 11 Historia de usuario 5.....	57
Tabla 12 Tecnologías utilizadas en el proyecto	69
Tabla 13 Product Backlog del sistema.....	72
Tabla 14 Planificación del sprint 1	73
Tabla 15 Pruebas de Caja Negra Sprint 1	78
Tabla 16 Pruebas de Caja Blanca Sprint 1	79
Tabla 17 Estado actual del Product Backlog - Sprint 1	79
Tabla 18 Planificacion del Sprint 2.....	81
Tabla 19 Sprint Backlog 2	82
Tabla 20 Pruebas de Caja Negra Sprint 2	86

Tabla 21 Prueba de Caja Blanca Sprint 2	86
Tabla 22 Estado actual del Product Backlog - Sprint 2	87
Tabla 23 Planificacion del Sprint 3.....	88
Tabla 24 Sprint Backlog 3	89
Tabla 25 Pruebas Caja Negra Sprint 3.....	92
Tabla 26 Pruebas de Caja Blanca - Sprint 3	92
Tabla 27 Estado actual del Product Backlog - Sprint 3	93
Tabla 28 Planificacion del Sprint 4.....	94
Tabla 29 Backlog Sprint 4	95
Tabla 30 Pruebas de Caja Negra Sprint 4	100
Tabla 31 Pruebas de Caja Blanca Sprint 4.....	101
Tabla 32 Product Backlog Sprint 4.....	103
Tabla 33 Planificación de la evaluación del sistema	109
Tabla 34 Registro de pruebas de predicción de estrés hídrico (ML)	111
Tabla 35 Registro de pruebas de actualización del dashboard visual (Flutter).....	114

ÍNDICE DE ILUSTRACIONES

Ilustración 1 Diagrama causa - efecto del problema de la eficiencia del riego en vivero Lastenia ULEAM El Carmen	6
Ilustración 2 Diagrama de Caso de Uso: Consulta mediante lenguaje natural	59
Ilustración 3 Diagrama de Caso de Uso: Evaluación y actualización de riesgo ambiental	60
Ilustración 4 Diagrama de Caso de Uso: Consulta de historial de conversaciones	60
Ilustración 5 Diagrama de Secuencia: Envío de consulta y actualización de riesgo.	61
Ilustración 6 Diagrama de Secuencia: Predicción de estrés hídrico y evaluación de riesgos..	62
Ilustración 7 Diagrama de Secuencia: Carga del historial en el panel lateral.....	63
Ilustración 8 Diagrama de Estado: Ciclo de vida de la pantalla principal (Dashboard).	64
Ilustración 9 Diagrama de Estado Evaluación de riesgo ambiental.....	65
Ilustración 10 Diagrama de Estado Petición HTTP y envío de mensajes.....	66
Ilustración 11 Diagrama de Base de Datos	67
Ilustración 12 Arquitectura general del sistema.....	68
Ilustración 13 Código fuente: Componente de diseño para las burbujas de chat en Dart	76
Ilustración 14 Alerta ante indisponibilidad del motor LLM local	77
Ilustración 15 Código fuente: Estructura JSON del payload para la petición HTTP.....	83
Ilustración 16 Código fuente: Método asíncrono _enviarConsulta	84
Ilustración 17 Código fuente: Clase MLModelClassifier para predicción local	90
Ilustración 18 Código fuente: Lógica condicional de actualización dinámica de riesgo.....	96
Ilustración 19 Demostración de la actualización dinámica del riesgo en el dashboard.....	97
Ilustración 20 Generación del paquete de distribución (Release APK).....	99

Ilustración 21 Funcionamiento del Chatbot en dispositivo móvil (Estados BAJO y ALTO)	106
Ilustración 22 Evidencia de reconocimiento de intenciones en la interfaz móvil.....	111
Ilustración 23 Evidencia de actualización dinámica de alertas en la interfaz móvil del chatbot.	113
Ilustración 24 Evidencia de peticiones exitosas para la recuperación del historial en el servidor.	115

ÍNDICE DE ANEXOS

Anexo A Asignación de tutor	127
Anexo B: Reporte del sistema antiplagio.....	128
Anexo C Fotografías	¡Error! Marcador no definido.
Anexo D Estructura de la entrevista aplicada	¡Error! Marcador no definido.
Anexo E Estructura de encuesta aplicada	¡Error! Marcador no definido.

RESUMEN

El estrés hídrico en el vivero representa un gran riesgo para el cultivo de cacao. Esta problemática suele perdurar y agravarse cuando no se cuenta con los equipos automáticos adecuados para controlar este proceso, forzando a los agricultores a depender únicamente de su propia experiencia, desencadenando posibles pérdidas en la producción. Dentro de este margen de ideas, se optó por el diseño y ejecución de un asistente virtual basado en el aprendizaje automático supervisado. El propósito principal de este sistema busca mejorar el manejo de los procesos de riego dentro de las instalaciones del Vivero Lastenia, perteneciente a la ULEAM, extensión El Carmen.

El presente estudio manejó una metodología que integra el análisis cuantitativo con el desarrollo tecnológico. La información obtenida se recolectó con base en un grupo de 72 personas seleccionadas, a partir de una población compuesta por 89 personas en total, utilizando técnicas como la encuesta, entrevistas y la observación directa en el terreno de estudio. En el campo de la ingeniería de software, el proyecto se construyó con arquitectura Cliente-Servidor, enlazando una aplicación para dispositivos móviles programada en Flutter con un servidor backend de tipo asíncrono, el cual funciona mediante FastAPI. Para entrenar la inteligencia artificial de la plataforma, se optó por integrar un módulo de Procesamiento de Lenguaje Natural, junto a un modelo de predicción de algoritmo Random Forest.

Durante las pruebas técnicas, el modelo de Procesamiento de Lenguaje Natural demostró una alta confiabilidad al procesar las consultas del usuario, alcanzando una exactitud (Accuracy) del 96.67%, una sensibilidad (Recall) del 93.33% en la detección de alertas reales de estrés hídrico, y un F1-Score general del 96.55%. Estos resultados reflejan que la herramienta automatiza exitosamente las decisiones operativas tomadas por el personal y respalda de manera efectiva el manejo agrícola del cultivo.

ABSTRACT

Water stress in the nursery poses a significant risk to cocoa cultivation. This issue tends to persist and worsen in the absence of appropriate automated equipment to manage the process, forcing farmers to rely solely on their own experience and potentially leading to production losses. Against this backdrop, the decision was made to design and implement a virtual assistant based on supervised machine learning. The system's primary purpose is to improve irrigation management at the Lastenia Nursery, part of the ULEAM El Carmen extension.

This study employed a methodology integrating quantitative analysis with technological development. Data was gathered from a sample of 72 individuals—selected from a total population of 88—using techniques such as surveys, interviews, and direct field observation. From a software engineering perspective, the project utilized a client-server architecture, linking a mobile application programmed in Flutter with an asynchronous backend server running on FastAPI. To train the platform's artificial intelligence, a Natural Language Processing module was integrated alongside a Random Forest prediction model.

Technical testing demonstrated that the application operated swiftly and stably; the language classifier successfully identified query intent in just 0.81 seconds. Additionally, the interface updated visual alerts in 0.22 seconds, while the database retrieved message history in 0.45 seconds without interruption. The results indicate that the tool successfully automates operational decisions made by staff and effectively supports agricultural crop management.

CAPÍTULO I

1 INTRODUCCIÓN

1.1 Introducción

El presente proyecto aborda la importancia de la tecnología inteligente para el sector agrícola, enfocándose en el desarrollo y la aplicación de un chatbot con aprendizaje automático supervisado, el cual su objetivo es la eficiencia del riego en los cultivos de la planta de cacao, en el vivero Lastenia Vera de la Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen. Dicho entorno suele enfrentar diferentes retos como la administración del agua, el rendimiento agrario, la falta de personal calificado y, sobre todo, la escasa conectividad a la red. Evidentemente, es esencial mantener un monitoreo continuo sobre el clima y el suelo, debido a su alta influencia en el crecimiento de las plántulas. Para lograr un suministro de agua efectivo, se necesita medir con exactitud ciertas variables físicas en el terreno, teniendo en cuenta el contenido volumétrico de humedad y la temperatura del ambiente. El análisis detallado de toda esta información permite establecer modelos hídricos precisos. Implementar esta metodología se vuelve determinante para asegurar la viabilidad de las plantas, impulsar el rendimiento agrícola y detener el derroche constante de recursos naturales e insumos en las instalaciones del vivero.

Dentro del aprendizaje supervisado, el sistema analiza la información del terreno y del clima para identificar tendencias y sugerir las mejores decisiones al usuario, valiéndose de la minería de datos. Este proceso ayuda a predecir cuánto riego necesitan las plantas en cada momento, permitiendo que el suministro de agua se adapte automáticamente según los cambios que se realicen.

Este chatbot inteligente une todas las variables descritas mediante una interfaz móvil que toma y envía datos importantes al instante. La herramienta ofrece guía personalizada, facilita el trabajo diario en el vivero y optimiza la hidratación del cultivo. Al juntar estas tecnologías, se entrega a los agricultores una solución moderna que ayuda a reducir las pérdidas, subir la producción y motivar el uso de nuevos sistemas digitales en el campo de nuestra zona.

1.2 Presentación del tema

El tema establecido es relevante por qué el uso de un chatbot con aprendizaje automático supervisado permitirá el uso eficiente del riego para las plantas de cacao, el cual se integrará tecnología inteligente que ayude a optimizar el recurso del agua y mejore la productividad en el vivero agrícola Lastenia Uleam El Carmen.

1.3 Ubicación y contextualización de la problemática

La Universidad Laica Eloy Alfaro de Manabí, o ULEAM, se creó en 1985. Esta universidad es una institución pública que ofrece educación superior. Tiene varias sedes en diferentes partes del país, como la Extensión en El Carmen. La Universidad Laica Eloy Alfaro de Manabí quiere ayudar a la gente que trabaja en la agricultura y la ganadería de la zona. Para lograr esto, la universidad siempre está empezando nuevos proyectos para mejorar la forma en que se trabaja en el campo. Lo que busca la Universidad Laica Eloy Alfaro de Manabí con estos proyectos es que los estudiantes de la región puedan aprender mejor y tener más habilidades prácticas cuando se gradúen.

El vivero donde se realizara la implementación del proyecto se encuentra ubicado en un terreno que antes era un colegio llamado Lastenia Vera, el cual ahora la Uleam es el propietario de ese espacio y lo usa con fines educativos para la apertura de nuevas carreras, actualmente este espacio está situado en una zona rural y apartado del centro del cantón, por lo que permite el desarrollo de cultivos bajo condiciones climáticas privilegiadas gracias a la humedad del cantón, se están dedicando en el cultivo especialmente en la planta del cacao.

Se decidió implementar el proyecto en esta zona debido que, a pesar de contar con un entorno favorable para la agricultura del cacao, el vivero Lastenia enfrenta varias deficiencias en el área operativa de control de riego, originadas por la aplicación empírica y basada en la observación visual del personal encargado del mantenimiento y gestión de las plántulas. Esta práctica genera una distribución inconsistente del recurso causando el estrés hídrico. Además, la disponibilidad limitada de redes de internet y mano de obra especializada evidencia que la necesidad de introducir soluciones inteligentes automatizadas, por esa razón el proyecto se planteará en la aplicación de un chatbot de alta usabilidad con aprendizaje automático

supervisado, con el objetivo de modernizar la gestión hídrica del vivero y fomentar la eficiencia como la productividad.

1.4 Planteamiento del problema

1.4.1 Problematización

El proceso de siembra en el vivero presenta dificultades que obstaculizan la uniformidad y el éxito del cultivo de cacao. La selección inadecuada de semillas y la manipulación manual inconsistente resultan en tasas variables de germinación y crecimiento. Además, la falta de un registro técnico sobre fechas y condiciones de siembra limita la capacidad de seguimiento y mejora del proceso, haciendo difícil identificar y solucionar las causas de bajo rendimiento en las plántulas.

Cuando se realizó la preparación y el llenado de las fundas con tierra abonada, fue muy común encontrar problemas debido a la mezcla ya que no era precisa o era al ojo, esto causaba sobrecompactación del suelo. Esta situación se genera por la falta de herramientas esenciales, como las palas y carretillas, lo que retrasa el ritmo del trabajo y complica la distribución de la tierra, como consecuencia, el crecimiento de las plantas se deteriora, a la vez que el agotamiento físico de los trabajadores aumenta y frena los labores.

En el vivero se enfrenta dificultades sobre el tema de abono ya que actualmente la falta de materiales de trabajo y sacos de abono para fertilizar la tierra limita el trabajo y obliga a los trabajadores compartir recursos, lo que termina en una apuesta para medir la cantidad exacta que se debe insertar, por lo que sin una guía técnica o por mano de obra con poca experiencia puede llegar a perjudicar a la planta, por lo que la aplicación de fertilizante se vuelve empírica. Esto genera dudas sobre las dosis correctas y puede causar que las plantas de cacao no crezcan sanas por falta o exceso de nutrientes.

Actualmente el riego depende bastante del trabajo manual y no sigue una planificación técnica, a razón de que no existe un agente inteligente que ayude a tomar decisiones. Esa falta de control causa que a veces se desperdicie el agua u otras veces las plantas sufren de sequía o exceso de agua por lo que causa el estrés hídrico, como resultado muchas plantas no logran sobrevivir y su crecimiento se retrasa. El acceso limitado a información técnica y la ausencia de asesoría dificultan programar el riego según las necesidades específicas de cada etapa.

La falta de las herramientas y recursos adecuados retrasa el trabajo del agricultor y también reduce la productividad del vivero Lastenia Vera. El riego en particular, se realiza de manera manual y sin datos técnicos precisos, lo que genera desperdicio de agua, estrés hídrico en las plantas y baja tasa de supervivencia. Ante esta problemática, surgen preguntas clave sobre cómo podría ayudar a solucionar estos problemas el chatbot:

- ¿De qué manera afecta al desarrollo del cultivo la falta de conocimiento técnico sobre la necesidad hídrica exacta que requiere la planta de cacao?
- ¿Cómo afecta la carencia de un sistema automatizado en la planificación, organización y ejecución de los horarios de riego en el vivero?
- ¿Qué impacto tendría la sistematización de la información agronómica en la optimización de los volúmenes de agua destinados al mantenimiento de las plántulas?
- ¿Cómo influye la implementación de esta tecnología para el ahorro de agua y en el crecimiento saludable de la planta de cacao?

Frente a este escenario, surge la necesidad de investigar mecanismos tecnológicos que logren resolver las dudas técnicas del personal en tiempo real, permitiendo así optimizar el proceso de riego y aportar soluciones operativas y accesibles para los actuales desafíos del vivero.

1.4.2 Génesis del problema

Hace aproximadamente un año, se implementó el proyecto de siembra de cacao en el vivero Lastenia de la Universidad Laica Eloy Alfaro de Manabí. Este proyecto consistió en la siembra de un millón de plantas con la finalidad de fortalecer la producción de cacao y generar recursos para la extensión universitaria. Sin embargo, a medida que avanzaban las actividades, comenzaron a evidenciarse diversas dificultades operativas asociadas al manejo y mantenimiento de las plantas.

Se identificó que una de las mayores dificultades es la pérdida constante de las plantas, a causa de que muchos estudiantes no tienen la experiencia previa en tareas agrícolas y usan las herramientas de manera inadecuada ya que aún no son expertos. Esta situación no solo genera la pérdida de agua e insumos, sino que también nos confirma que hace falta una guía técnica

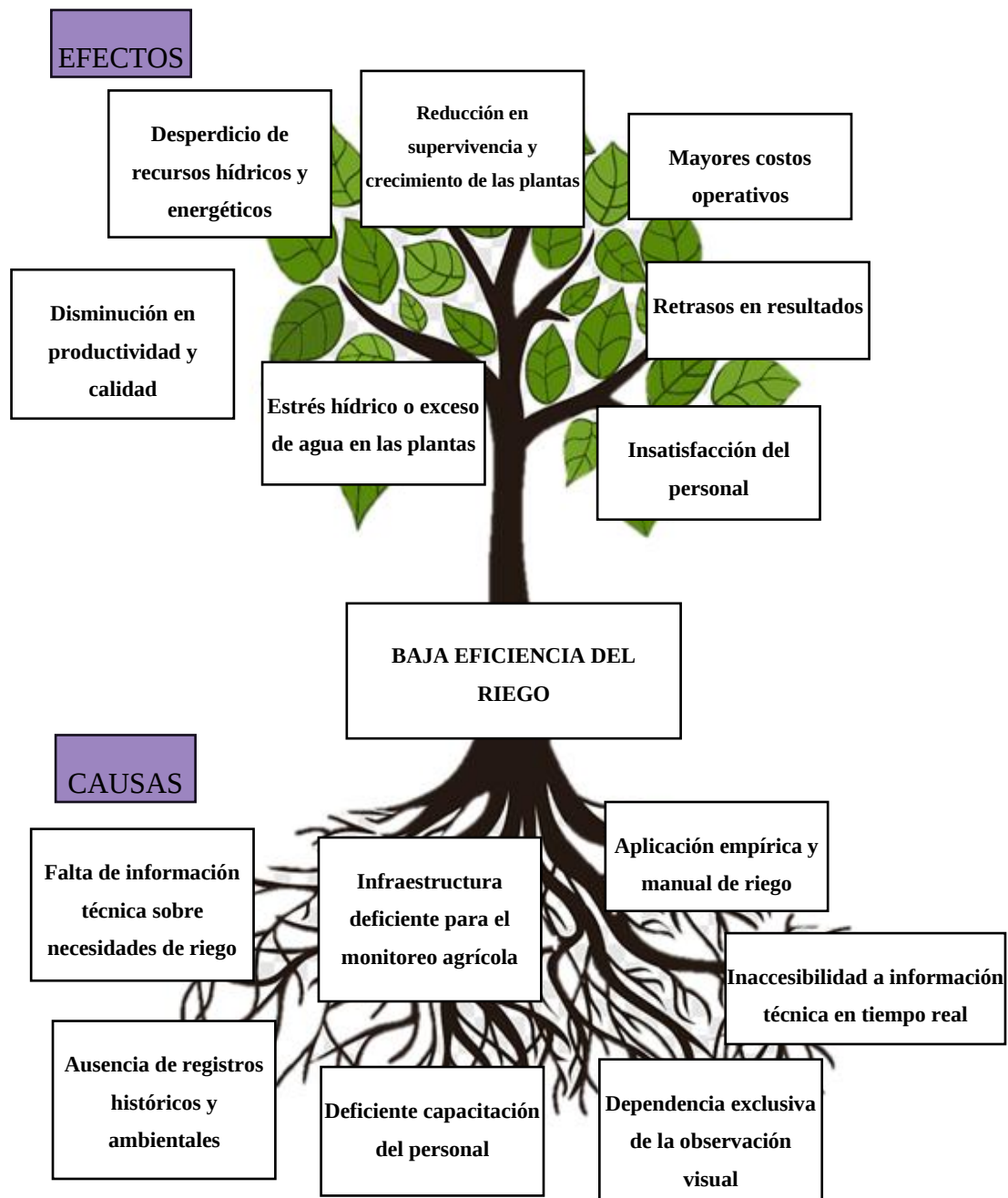
inteligente, de esta manera se logrará optimizar el trabajo diario y la enseñanza de los estudiantes dentro del vivero.

1.4.3 Estado actual del problema

Actualmente, la gestión del riego en el vivero Lastenia presenta dificultades operativas. La falta de experiencia técnica en el manejo del recurso hídrico limita la garantía del suministro eficiente del agua. En la práctica diaria, gran parte de los estudiantes desconocen la cantidad exacta de agua que se debe aplicar. Esta brecha de conocimiento provoca el deterioro de los insumos, el desperdicio sistemático del agua y el retraso del desarrollo de las plantas, afectando el desarrollo en la producción. Este escenario sumando la ausencia de la supervisión técnica necesaria y la carencia de los instrumentos de monitoreo o guías, lo que perpetúa la ejecución de prácticas agrícolas empíricas. Esta acumulación de deficiencias compromete la sostenibilidad operativa de las instalaciones y afecta de manera directa a la calidad del producto final.

1.5 Diagrama causa – efecto del problema

Ilustración 1 Diagrama causa - efecto del problema de la eficiencia del riego en vivero Lastenia ULEAM El Carmen



1.6 Objetivos

1.6.1 Objetivo general

Desarrollar un chatbot con aprendizaje automático supervisado para que gestione la información agronómica del cultivo del cacao, con el fin de optimizar la eficiencia del riego en el vivero Lastenia Uleam El Carmen.

1.6.2 Objetivos específicos

Identificar el problema relacionado con la eficiencia del riego en el vivero Lastenia Vera de la Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen, analizando las causas y efectos que afectan el desarrollo del cultivo de la planta del cacao.

Analizar la información teórica recopilada acerca de los sistemas inteligentes, el aprendizaje automático y las técnicas de riego para garantizar que la solución sea apropiada para el cultivo del cacao.

Realizar una investigación acerca del campo actual en el vivero Lastenia Vera, Uleam El Carmen y así poder recopilar la información más relevante acerca de las condiciones actuales, los recursos, procesos y problemas asociadas al riego de las plantas de cacao.

Desarrollar un chatbot que ayude a no desperdiciar el agua en el vivero, usando varios tipos de modelos de aprendizaje que permita manejar mejor el riego de las plantas.

Probar el funcionamiento del sistema y ver cómo afecta directamente en el vivero, para confirmar si realmente ayuda a manejar el agua y si sirve para que las plantas crezcan mejor.

1.7 Justificación

La implementación de un chatbot para la eficiencia del riego responde a la necesidad de modernizar los procesos agrícolas en el vivero Lastenia Vera ULEAM El Carmen. Este sistema representará una solución innovadora que contribuirá a resolver las limitaciones identificadas en la gestión hídrica, como la falta de información técnica, la aplicación manual y empírica del

riego, y lo difícil que es aprovechar mejor lo que se tiene en el campo cuando no se cuenta con las herramientas necesarias y a la tecnología.

El mayor beneficio de este sistema es que ofrece recomendaciones de riego al instante y de forma personalizada adaptándose a lo que la planta de cacao necesita según el clima actual. Gracias a esto se podrá controlar el agua de manera inteligente y así evitando desperdicio logrando que muchas plantas sobrevivan. El chatbot ayudará en a controlar y guiar de manera constante al personal que trabaje en el vivero, por lo que si el trabajador tiene una duda el asistente le guiará de manera sencilla, ayudando a decidir cuándo y cuanto regar sin la necesidad de ser un experto usando la tecnología.

Los principales beneficiados serán los encargados del vivero y productores locales, ya que podrán cuidar mejor sus recursos, bajar los costos de operación y sacar el mayor provecho a sus cosechas. De manera indirecta el proyecto ayudará a los estudiantes y docentes de la Extensión, gracias de una herramienta para aprender e investigar el campo. Con esto la universidad se acerca más a los problemas del sector agrícola, compartiendo tecnología que impulsa el crecimiento del cantón.

En resumen, usar este sistema ayudará a que la producción del cacao sea mejor y que se mantenga bien con el tiempo, haciendo que el vivero funcione de manera profesional. Además esto servirá como un ejemplo de cómo se puede usar la tecnología para mejorar el trabajo en el campo.

1.8 Impactos esperados

1.8.1 Impacto tecnológico

La implementación del chatbot con aprendizaje automático supervisado representará un avance tecnológico clave al introducir una solución de Agricultura de Precisión accesible en un entorno con recursos limitados como el vivero Lastenia. Este sistema facilita el uso de la tecnología de dos formas clave: primero, permitirá controlar el riego mediante un chat sencillo, lo cual es mucho más fácil que usar programas complicados. Segundo, pone al alcance de cualquier usuario datos importantes como la humedad y la temperatura del suelo sin la necesidad de comprar algún tipo de equipo costoso o software profesional, haciendo que la información técnica sea fácil de entender y usar en el día a día.

Gracias al aprendizaje supervisado, el sistema hace más que solo guardar datos, los analiza para dar recomendaciones sobre qué pasará con el suelo y el clima. Esto permite que el encargado del vivero deje de regar solo por intuición o costumbre y empiece a tomar decisiones basadas en lo que el sistema ha aprendido de los sensores, logrando un uso más inteligente del agua mediante el aprendizaje automático. Esto no solo capacitará al personal agrícola en el manejo de sistemas automatizados de forma práctica, sino que establece una base para la futura integración de tecnologías IoT y sensores avanzados en la ULEAM El Carmen.

1.8.2 Impacto social

El objetivo principal en la parte social de este proyecto es que los estudiantes que trabajan en el vivero u otros operadores tengan mejores condiciones en su día a día. Al usar este sistema se evitará cansancio físico y se ahorrarán el montón de horas que se pierden regando a mano, esto permite que los estudiantes de la Uleam y el personal ya no pierdan el tiempo en tareas repetitivas y puedan dedicarse a cosas más útiles, como estudiar el crecimiento de las plantas o aprender nuevas técnicas, así el vivero funcionará de mejor manera y todos estarán más preparados para usar herramientas modernas en el campo.

Además del vivero, este proyecto ayudará en que la tecnología llegue a zonas rurales del cantón, al demostrar que se puede reducir la pérdida de las plantas, los agricultores verán que el sistema es de fiar y querrán aprender a usarlo e implementar en sus campos de trabajo, motivando a los productores a innovar sus cosechas de cacao. Esto mejorará la economía del agricultor y de la zona agrícola ya que asegura una producción estable moderna.

1.8.3 Impacto ecológico

El proyecto ofrece un aporte ambiental clave al fomentar el cuidado responsable del agua en el sector agrícola. Mediante el uso de la inteligencia artificial se logra reducir el desperdicio innecesario y así se asegura que el vivero funcione bajo un modelo que respete a la naturaleza y optimizando cada gota para el crecimiento de la planta. Al eliminar la aplicación de riego manual y empírica, el sistema garantiza que el Theobroma Cacao reciba la cantidad precisa de agua requerida en el momento óptimo, lo que resulta en una drástica reducción del desperdicio de agua por excesos o escorrentías. Es muy importante proteger este líquido para la región. Esto ayuda a evitar que los nutrientes y los agroquímicos se filtren hacia el suelo y el agua

subterránea. Cuando las plantas están sanas, es más probable que sobrevivan cada día. De esta manera, podemos mantener la biomasa vegetal del vivero en buen estado. Al seguir este método, fomentamos el uso de prácticas agrícolas responsables, lo que realmente beneficia al medio ambiente.

CAPÍTULO II

2. MARCO TEÓRICO

2.1 Antecedentes históricos

2.1.1 Historia de los Chatbots y el Aprendizaje Automático Supervisado

Para entender bien cómo funcionará el chatbot que se va a implementar en el vivero Lastenia Vera, es fundamental revisar como se ha evolucionado la comunicación entre las personas y sistemas computacionales. Esto no solo tiene que ver con el tiempo, sino que también muestra cómo hemos pasado de respuestas automáticas simples a una forma de comunicación más inteligente, que es lo que se utiliza ahora en la agricultura. Como bien planteó (Turing, 1950) en su estudio sobre la inteligencia de las maquinas, el desafío siempre fue el poder emular los pensamientos de los seres humanos, el cual es un objetivo que hoy se vuelve práctico al querer inventar que un sistema tome decisiones por si sola en el tema del riego.

Cuando se realizó el estudio sobre la evolución de la tecnología en este caso la historia de los chatbots, se tocó un tema bastante interesante el cual habla sobre ELIZA, este fue el punto de partida en la evolución de esta tecnología. (Weizenbaum, 1966) lo diseñó bajo un esquema rígido en reglas de sustitución, demostró mucha relevancia para este trabajo de titulación ya que abarca temas como la posibilidad de la interacción y comunicación del hombre con la máquina. A diferencia del chatbot, ELIZA solo simula la comprensión, en cambio el chatbot que se propone realizar para el vivero Lastenia Vera busca ir más allá, utilizando herramientas de aprendizaje supervisado para procesar datos reales como la humedad y la temperatura, lo cual es más que solo respuestas configuradas.

Con el paso de los años, el enfoque de estos sistemas evolucionó hacia una comprensión más profunda del lenguaje natural. En pleno siglo XXI, la revolución del Machine Learning permitió que los modelos avanzados no solo identifiquen intenciones, sino que también extraigan datos específicos con mayor precisión. Como señala (Alpaydin, 2016) , el uso del aprendizaje automático supervisado ha sido un pilar fundamental para este cambio, ya que permite predecir resultados previamente etiquetas. En el marco de este proyecto de titulación, esta sofisticación tecnológica es la que permite que el chatbot trascienda el ámbito del entretenimiento y se adapte con éxito en la agricultura, convirtiéndose una herramienta clave para la toma de decisiones en el vivero Lastenia Vera.

2.1.2 Historia de la Gestión del Riego

El riego es una parte fundamental para la agricultura, siendo una de las herramientas más viejas que existen. Su avance siempre ha tenido el mismo objetivo el cual es lograr que los cultivos crezcan bien aunque no llueva, asegurando que siempre haya comida disponible a pesar del cambio climático.

Al estudiar cómo empezó la ingeniería agrícola, vemos que hace mucho tiempo, en la Antigüedad, que va desde 5000 a.C. hasta 500 d.C., las civilizaciones de Mesopotamia y Egipto crearon formas de manejar el agua. Usaban canales, diques y herramientas simples como el shaduf o la noria. Aunque hoy en día esto parece básico, fue el primer paso que dio la humanidad para controlar el medio ambiente y asegurar que las cosechas crecieran bien. Para este proyecto en el vivero Lastenia, rescatamos de esta etapa histórica el concepto fundamental de la distribución del recurso; sin embargo, el desafío actual es transformar esa gestión manual y empírica de los orígenes en un sistema de precisión que responda a las necesidades hídricas específicas de la planta de cacao.

A lo largo del periodo medieval y la época moderna, el avance técnico en la gestión del agua sufrió un evidente estancamiento. El esfuerzo de las comunidades se redujo, básicamente, a perfeccionar las redes de canales que ya tenían construidas. Al estudiar los antecedentes de aquellos años, queda claro que las labores de cultivo recaían por completo en la experiencia práctica y el instinto del agricultor. Puesto que no existían aparatos precisos para programar los turnos de hidratación, la entrega del recurso se ejecutaba guiándose únicamente por la inspección visual directa, omitiendo por completo el uso de procedimientos de medición cuantitativa. No había forma de medir con exactitud. Todo se hacía sin datos concretos. Esta misma dependencia de la observación directa es la que todavía se presencia en los procesos actuales del vivero Lastenia Vera, y es precisamente lo que se busca optimizar mediante el uso de tecnología de software y aprendizaje automático.

Cuando llegó el siglo XX, la gestión del agua experimentó un cambio radical de masificación de los sistemas por aspersión y, especialmente el riego por goteo el cual ganó mucha relevancia mundialmente en la década de 1960. Como bien analizaba (Gebbers, 2010) , estos métodos introdujeron el concepto de la eficiencia al llevar el recurso directamente a la raíz, minimizando la pérdida de la producción. Sin embargo, aunque el goteo mejoró la entrega física, en el vivero

Lastenia Vera se busca dar el siguiente paso, usar la inteligencia artificial para decidir con precisión el momento exacto de cuando regar y así de ese modo la herencia técnica que destaca Gebbers se complementa con el aprendizaje automático logrando sostenibilidad en el cultivo del cacao.

2.1.3 Historia de Chatbots para Riego

Usar chatbot para el trabajo del campo es algo nuevo que ha surgido gracias al avance de la tecnología, aunque ya existen sistemas para regar de manera automática desde hace mucho tiempo, manejarlos suele ser difícil o se necesita alguna capacitación. Esto hace que los dueños de las fincas no lo usen porque lo ven muy complejo.

Hoy en día se busca que la tecnología avanzada para el campo esté al alcance de todos, por eso han aparecido chatbots para el riego, que juntan la facilidad de usar aplicaciones de mensajes comunes. Estos primeros modelos y pruebas empezaron a verse desde hace pocos años, a finales de 2010.

La importancia de juntar estas tecnologías es para que permitan tomar decisiones al momento según lo que esté pasando en el cultivo. En este proyecto el chatbot no funciona como una simple alarma, sino que se vuelve un asistente para el campo que pueda entender los datos del suelo y ambiente. Al procesar esta información con minería de datos el sistema le dará al usuario una respuesta clara logrando que el manejo del riego sea mucho más fácil para las plantas de cacao en el vivero Lastenia Vera.

2.2 Antecedentes de investigaciones relacionadas al tema presentado

Para poder establecer el marco de referencia del presente trabajo, se debió analizar diferentes tesis de ingeniería recientes que hablen sobre la automatización en el campo mediante el uso de sistemas conversacionales conocidos como chatbots y sistemas informáticos enfocados en el control del riego. Los proyectos seleccionados demostraron la viabilidad técnica y la relevancia del enfoque propuesto.

2.2.1 Análisis de investigaciones similares

Trabajo 1: Desarrollo de un prototipo de un sistema de riego automatizado para el procesamiento, monitoreo y análisis de datos utilizando lógica difusa en tiempo real e IoT para optimizar el uso de agua aplicada en el cultivo

Este proyecto lo llevó a cabo la Universidad Politécnica Salesiana en Ecuador. El objetivo era crear un sistema para aprovechar mejor el agua en los cultivos. Para lograrlo, utilizaron una red de sensores y algoritmos en tiempo real. Los resultados mostraron que el sistema optimizó el uso del agua sin afectar la salud de las plantas. Aunque su enfoque es diferente al de un chatbot, el sistema usa lógica predictiva para medir la humedad del suelo. Esto coincide con el control de riego en tiempo real de esta investigación. El éxito del proyecto se basa en combinar hardware y software para tomar decisiones de forma autónoma (Ortiz Cáceres, 2021).

Trabajo 2: Sistema web y móvil para la detección de la monilia en las mazorcas de cacao (CCN-51) mediante visión por computadora

En la Universidad Agraria del Ecuador, se hizo un estudio. La investigación se enfocó en crear una plataforma digital que se pueda usar en navegadores web y dispositivos móviles. El objetivo principal era utilizar modelos de aprendizaje automático en las plantaciones de cacao. Al probar el sistema, se vio que la inteligencia artificial es muy útil, ya que permite a los productores tomar medidas preventivas de manera rápida. Lo que más se relaciona con este proyecto es precisamente ese enfoque: usar algoritmos para hacer que el manejo agrícola sea digital, adaptando la tecnología a las condiciones del lugar donde se trabaja (Rendón Cuero & Rodríguez Perea, 2023).

Trabajo 3: Sistema de riego automatizado para *Solanum lycopersicum* con IoT y modelos predictivos para el ahorro de agua en entornos urbanos

Esta tesis se hizo en la Universidad de Lima en Perú. El objetivo era crear un sistema automatizado que no cueste mucho dinero y que permita controlar el riego con ayuda de sensores de humedad y modelos que pueden predecir lo que va a pasar. Lo más importante que se logró fue comprobar que el sistema es fuerte, económico y fácil de usar para evitar desperdiciar agua. Esto coincide completamente con la idea de automatizar el riego y usar

formas de comunicación para que los usuarios puedan interactuar con el sistema. Sin embargo, en la Universidad Laica Eloy Alfaro de Manabí Extensión El Carmen (ULEAM) se va a hacer algo un poco diferente, porque allí se va a integrar el control del riego mediante una plataforma de chatbot (Anci Yep & Hermosa Caceres, 2025).

Al revisar estos trabajos, se confirma que la tecnología es muy importante para mejorar el riego. Mi proyecto en el vivero Lastenia Vera es un poco diferente. Ancí Yep y Hermosa Caceres, en el año 2025, se enfocaron en el hardware predictivo. Rendón Cuero y Rodríguez Perea, en el año 2023, comprobaron la eficacia del aprendizaje automático en el cacao. Pero el trabajo de Ortiz Cáceres, en el año 2021, valida la automatización hídrica. La gran diferencia es que aquí, en la ULEAM, se integrará todo en un Chatbot. Esto permite que el sistema aprenda de los datos y, al mismo tiempo, nos dé una interfaz amigable para controlar el riego del cacao desde el celular.

2.3 Definiciones conceptuales

2.3.1 Chatbot con Aprendizaje Automático Supervisado

Un chatbot con aprendizaje automático supervisado es una herramienta diseñada para conversar mediante el uso de la inteligencia artificial. Su funcionamiento se basa en el entrenamiento con datos etiquetados. Durante ese proceso, se le enseña al modelo el cómo debe relacionarse y como debe responder los mensajes que recibe de parte del usuario. A diferencia de los programas tradicionales rígidos, este enfoque analiza ejemplos históricos para identificar patrones y generalizar el aprendizaje, lo que le faculta para interpretar nuevas consultas y proponer recomendaciones técnicas precisas basándose en la experiencia acumulada durante su fase de entrenamiento (Alpaydin, 2016).

2.3.1.1 Definición y Evolución Histórica de Chatbots

Un chatbot se define técnicamente como un programa informático diseñado para simular conversaciones inteligentes con usuarios humanos, empleando el texto o la voz como canales principales de comunicación. Estos sistemas actúan como una interfaz operativa entre los servicios digitales y el usuario final; mediante el Procesamiento del Lenguaje Natural (PLN), interpretan consultas y ejecutan tareas proporcionando recomendaciones de manera autónoma,

garantizando una disponibilidad de 24 horas al día sin requerir intervención humana directa (Adamopoulou & Moussiades, 2020).

En 1966 apareció el primer chatbot de la historia el cual fue llamado ELIZA, creado en el instituto tecnológico de Massachusetts, este sistema funcionaba buscando palabras clave para responder como si fuera un psicólogo así demostrando que se podía imitar una conversación humana solo usando palabras ya definidas, aunque en sí el sistema no comprendía el significado de las frases (Weizenbaum, 1966).

Pero antes de que existieran esos programas, Alan Turing ya había puesto las bases en 1950. El inventó el juego de la imitación para saber si una máquina era inteligente, esta idea luego se llamó el test de Turing, decía que si una computadora habla de forma que una persona no puede darse cuenta de que es una máquina, entonces se puede decir que piensa. Esta meta es la que ha hecho que los asistentes pasen de ser simples textos a la inteligencia artificial que conocemos en la actualidad (Turing, 1950).

2.3.1.2 Arquitectura de Sistemas Conversacionales (Componentes)

La arquitectura técnica de un sistema conversacional moderno se estructura generalmente bajo un modelo modular cliente-servidor, diseñado para gestionar el flujo de información desde la entrada del usuario hasta la ejecución de una respuesta. La interacción se basa en la pantalla o el lugar donde se envían mensajes. Aquí es donde se escriben o se dicen los comandos. Esta parte se conecta directamente con el servidor principal mediante interfaces especiales llamadas API. El trabajo de esta parte es enviar lo que se escribe o se dice al cerebro del sistema y mostrar la respuesta. De esta manera, se puede navegar de forma sencilla sin tener que entender cómo funcionan el backend (Rendón Cuero & Rodríguez Perea, 2023).

Internamente, el núcleo informático alberga los motores de inteligencia: el Gestor de Diálogo y el módulo de Comprensión del Lenguaje Natural (NLU). La misión del NLU consiste en fragmentar las oraciones escritas. Con esto logra clasificar la "intención" principal del usuario e identificar las "entidades" clave del mensaje, como fechas o ubicaciones exactas. Tras interpretar exitosamente el texto, el Gestor de Diálogo toma el relevo para dictaminar qué acción ejecutará la máquina. Este componente mantiene la situación de la conversación y decide si es necesario pedir más información para aclarar la petición o si ya tiene los datos

necesarios para continuar, basándose en las reglas establecidas y en el contexto (McTear, Callejas, & Griol, 2016).

La capa de integración o cumplimiento (fulfillment), que vincula al chatbot con los sistemas externos y las bases de datos, es la que termina de completar la arquitectura. En este proyecto agrícola, este módulo es el encargado de consultar los sensores de monitoreo en tiempo real o de despachar las alertas predictivas al usuario cuando recibe una orden del Gestor de Diálogo. Esta capa convierte la intención abstracta del usuario en órdenes técnicas que se pueden llevar a cabo en el hardware o consultas directas a la base de datos, lo cual posibilita que el asistente virtual no solo dialogue, sino que también funcione de manera efectiva en el ambiente físico del cultivo (Ortiz Cáceres, 2021).

2.3.1.3 Procesamiento del Lenguaje Natural (NLP)

La disciplina de la inteligencia artificial que se encarga de la interacción entre las computadoras y el lenguaje humano es el Procesamiento del Lenguaje Natural (PLN o NLP, por sus siglas en inglés). Su finalidad es proveer a los sistemas informáticos de la capacidad para leer, interpretar y comprender los idiomas de manera productiva. A diferencia del procesamiento tradicional de datos, el PLN exige que el software no solo reconozca palabras individuales, sino también la estructura gramatical y el significado implícito en grandes cantidades de datos textuales no estructurados, lo cual hace posible convertir la comunicación verbal en información que puede ser procesada para tomar decisiones (Chowdhary, 2020).

El PLN, desde un punto de vista técnico, opera segmentando el lenguaje en diferentes fases con el objetivo de que la máquina entienda cómo se expresa la gente. En primer lugar está la parte morfológica, que analiza la estructura interna de las palabras, en segundo lugar, se encuentra la sintáctica, que se encarga del orden y el sentido gramatical de las oraciones. Luego se lleva a cabo el análisis semántico, que es fundamental para comprender qué quiere decir cada frase según del contexto. Los sistemas actuales incluyen, además, el nivel pragmático, este permite al sistema entender lo que realmente quiere expresar el usuario y no solo las palabras sueltas (Jurafsky & Martin, 2024).

El PLN es importante para transformar el texto que escribe el usuario en datos numéricos que los modelos de aprendizaje automático logran procesar, sobre todo en los asistentes virtuales.

Este proceso usa técnicas de tokens y reconocimiento de entidades que se reconocen como NER, permitiendo que el chatbot identifique datos claves dentro de un mensaje. De esta forma el sistema puede reconocer fácilmente si el usuario se refiere a la ubicación de un sector específico de la planta o a una variable del clima para determinar de manera adecuada el riego. La adecuada puesta en práctica de estos procesos lingüísticos distingue a un asistente inteligente, que puede comprender peticiones diversas, de un sistema de comandos estricto que no logra adaptarse a cambios sintácticos mínimos (Eisenstein, 2019).

2.3.1.4 Aprendizaje Automático Supervisado (Enfoque Teórico)

Desde un punto de vista teórico, el aprendizaje automático supervisado es el modelo más común en la creación de sistemas inteligentes. Este se basa en inducir una función de mapeo a partir de pares de datos de entrada y salida que han sido marcados con anterioridad. Este enfoque tiene como finalidad principal que el algoritmo, al examinar un conjunto de datos de práctica o conocidos como training set, logre entender cómo se conectan los datos de las entradas con los resultados que esperamos y la variable objetivo; así podrá predecir con gran precisión el resultado correspondiente para datos nuevos que no se han analizado previamente. Este procedimiento es diferente de los métodos no supervisados, en los que el sistema tiene que deducir patrones sin tener una dirección explícita sobre cuáles son los resultados adecuados (Russell & Norvig, 2021).

La esencia operativa de este modelo se encuentra en el proceso de optimización durante la etapa de entrenamiento. En esta etapa un desafío clave es lograr que el sistema aprenda a balancear la teoría con la práctica de los datos registrados. Un modelo bien entrenado no es el que se limita a memorizar los casos pasados, sino aquel que logra entender los patrones generales para no caer en el sobreajuste. Esto garantiza que el software sea confiable y funcione de manera correcta cuando le toque enfrentarse a situaciones nuevas o cambios inesperados en el clima (Géron, 2019).

Dependiendo a que tipo de respuesta se necesite que obtengamos, el aprendizaje supervisado se separa en tareas de clasificación. La meta de la clasificación es prever una etiqueta de categoría discreta, lo cual es esencial para que un chatbot reconozca el propósito de las preguntas del usuario. Por otro lado, en la regresión, se prevé un valor continuo; esto resulta útil, por ejemplo, al estimar cuántos litros de agua se necesitan para el riego tomando como

referencia factores climáticos. La efectividad del sistema para solucionar el problema concreto del dominio de aplicación está determinada por la elección adecuada del algoritmo dentro de estas categorías (Müller & Guido, 2016).

2.3.1.5 Clasificación de Chatbots por Nivel de Inteligencia (Reglas vs. ML)

El sistema de clasificación más simple de los asistentes virtuales se determina según su organización de las ideas, separando primero los chatbots que funcionan con lenguaje o reglas fijas.

Estos sistemas trabajan siguiendo una lista de pasos y reglas ya establecidas, donde la persona que programa el software decide que responder según lo que escriba el usuario, el sistema no es para tareas fáciles, como responder preguntas comunes, porque necesita encontrar palabras exactas para funcionar. Por eso si el usuario escribe algo que no estaba planeado o tiene errores de ortografía, el chatbot se confunde y no puede dar una respuesta clara (Nimavat & Champaneria, 2017).

Por el contrario, los chatbots que funcionan con inteligencia artificial o con el aprendizaje automático (ML) constituyen un avance cognitivo que tiene la capacidad de manejar interacciones complejas y no lineales. Estos agentes no se basan en respuestas fijas, sino que emplean algoritmos dinámicos para "aprender" a partir de interacciones anteriores y entender el contexto semántico de la oración, más allá de las palabras sueltas. Estos bots, utilizando métodos de entrenamiento supervisado, son capaces de deducir la intención del usuario, aun cuando se trate de frases ambiguas o desconocidas. En vez de recitar un texto grabado previamente, producen respuestas que se ajustan al tono y a la necesidad concreta del instante (Singh & Thakur, 2020).

La habilidad de generalizar y escalar es la diferencia crucial entre las dos aproximaciones. Un chatbot basado en reglas necesita una reprogramación manual incesante para añadir nuevas funcionalidades o reparar fallos de entendimiento, a diferencia de un chatbot propulsado por Machine Learning, que va perfeccionando su exactitud de manera autónoma a medida que maneja más datos. En ambientes agrícolas, donde las consultas pueden ir desde una simple pregunta sobre horarios hasta un diagnóstico complejo de riego basado en variables climáticas que están cambiando, es esencial contar con la adaptabilidad del enfoque de aprendizaje

automático para brindar una ayuda técnica verdaderamente útil y resistente (Rendón Cuero & Rodríguez Perea, 2023)

2.3.1.6 Plataformas y Canales de Comunicación (Interfaces)

La elección del canal de comunicación es una decisión crucial en la implementación de un asistente virtual, pues esta determina el acceso y lo que el usuario experimenta al final. Las interfaces o puntos de contacto mediante los cuales el ser humano interactúa con el bot se conocen como canales. Estos pueden incluir desde redes sociales y aplicaciones de mensajería instantánea (como WhatsApp o Telegram) hasta widgets web o interfaces de voz. Hoy en día, la tendencia tecnológica es propensa a emplear plataformas de mensajería móvil ya existentes, ya que estas eliminan el obstáculo que representa para el usuario descargar e instalar nuevas aplicaciones específicas. Esto posibilita que la interacción tenga lugar en un ambiente digital al cual ya está habituado y le resulta familiar (Smutny & Schreiberova, 2020).

En el ámbito específico de la agricultura y las áreas rurales, donde la conexión a internet puede ser limitada o intermitente, se vuelven más importantes desde un punto de vista técnico las herramientas sencillas como las aplicaciones de chat que todos usamos o los mensajes de texto comunes. Estas vías presentan beneficios significativos en comparación con las aplicaciones web convencionales; por ejemplo: el manejo asíncrono de mensajes, que permite al agricultor enviar una consulta y recibir luego la recomendación técnica sin tener que mantener una conexión constante. Esta propiedad es fundamental para asegurar que se reciban a tiempo alertas sobre el riego o alertas para dar mantenimiento en lugares donde la señal falla o cambia mucho (Anci Yep & Hermosa Caceres, 2025).

En cuanto a cómo está construido el sistema, la unión entre la inteligencia del chatbot y el usuario se hace con herramientas de comunicación comunes como APIs y Webhooks para conectar todo. Estos componentes actúan como el enlace que envía el mensaje del usuario desde la app móvil hasta el servidor donde se procesa los datos, para luego devolver la respuesta generada. Esta separación modular posibilita que un solo agente inteligente se extienda al mismo tiempo en varias plataformas, concentrando la administración del conocimiento agronómico en un único núcleo y ampliando el espectro de la herramienta hasta los dispositivos de los empleados (Ortiz Cáceres, 2021).

2.3.1.7 Arquitectura IoT y Flujo de Datos del Sistema

Se estructura la arquitectura técnica del Internet de las Cosas (IoT) en un modelo jerárquico de capas con el fin de asegurar que haya interoperabilidad entre los mundos físico y digital. La capa de percepción, que está formada por los dispositivos de hardware (como actuadores y sensores) que se comunican directamente con el ambiente físico, es la base de esta estructura. Encima de esta se ubica la capa de red, cuya misión es transmitir paquetes de datos de manera segura a través de protocolos inalámbricos como LoRaWAN o Wi-Fi. Esta organización modular posibilita que la información transcurra de forma ascendente, desde el campo hasta los servidores remotos, sin perder su integridad (Lea, 2018).

Dentro de la agricultura de precisión, obtener los datos del entorno es el primer paso en el proceso de obtener la información, los sensores colocados de forma estratégica miden constantemente los valores reales como la humedad del suelo y la temperatura del ambiente. Un microcontrolador conocido como nodo IoT recibe estas señales eléctricas, convierte la información a formato digital y lo manda de forma rápida al servidor en la nube. De esta manera, se genera un flujo permanente de telemetría que proporciona al sistema central información actualizada y veraz sobre el estado del cultivo (Ray, 2017).

Para concluir, la gestión integral de la información termina en la capa de aplicación y procesamiento. En este punto, los datos crudos se almacenan en bases de datos históricas y son procesados por los algoritmos del servidor. La integración con el chatbot tiene lugar en este momento: si el sistema inteligente observa que los niveles de humedad bajan del límite óptimo establecido previamente, o si el usuario pide un informe, la parte trasera consulta estos registros y procede de manera automática a despachar una notificación de alerta predictiva al dispositivo móvil del agricultor, completando de esta manera el ciclo de monitoreo preventivo (Madisetti & Bahga, 2014).

2.3.1.8 Métricas de Evaluación de Rendimiento de Chatbots (Accuracy, Recall, F1-Score)

Antes de que un chatbot se implemente en producción, es esencial realizar una evaluación cuantitativa de su desempeño cuando está basado en aprendizaje supervisado para validar su

eficacia. Como la identificación de intenciones (como diferenciar si el usuario desea "consultar humedad" o "activar riego") es, en esencia, un problema de clasificación de texto, se recurren

a métricas estadísticas que provienen de la matriz de confusión. Esta matriz facilita la visualización del número de aciertos del sistema (Verdaderos Positivos y Negativos) en comparación con sus errores (Falsos Positivos y Falsos Negativos), lo que brinda una base objetiva para el cálculo de indicadores de calidad, los cuales trascienden la mera percepción subjetiva del usuario (Grandini et al., 2020).

La métrica más frecuente y fácil de entender es la exactitud (accuracy), que se define como el cociente total entre las predicciones correctas y la cantidad total de casos analizados. Aunque es un indicador útil para obtener una visión general del sistema, tiene importantes limitaciones en conjuntos de datos desbalanceados. Esto es común en los chatbots, donde algunas intenciones (como el saludo) suceden con mayor frecuencia que otras (como una alerta técnica específica). Un modelo podría parecer que tiene una precisión muy elevada simplemente prediciendo siempre la clase más común, desatendiendo las intenciones críticas menos frecuentes, lo cual lo hace insuficiente como única medida del desempeño (Hossin & Sulaiman, 2015).

Para corregir estos fallos se emplea el F1-Score, que combina la precisión y así poder saber que tan exacto es el sistema al identificar el riego, con la sensibilidad sabremos cuantas ordenes reales se capturaron. Esta métrica es vital en los asistentes para el campo, pues ayuda a evitar los avisos falsos como el olvido de tareas importantes. Al usar el F1-Score aseguramos un equilibrio real, evitando que el chatbot de ordenes de riego equivocadas o ignore las necesidades urgentes de la planta. Un valor elevado en esta métrica asegura que el chatbot no solo es certero al responder, sino también que es constante en no ignorar las peticiones del agricultor (Sokolova & Lapalme, 2009).

2.3.2 Eficiencia del riego de Theobroma Cacao.

Se entiende como eficiencia de riego, en términos técnicos, la relación porcentual entre el volumen de agua que se almacena realmente en la zona radicular y es utilizada por el cultivo para sus funciones fisiológicas, en comparación con el total del agua que se extrae o aplica desde su fuente de abastecimiento. Este indicador agronómico es esencial para analizar la

ejecución operativa de un sistema hidráulico, porque hace posible medir y reducir las pérdidas de recursos hídricos que se producen por motivos como la evaporación durante el transporte y distribución, la escorrentía superficial o la percolación profunda. De este modo, se puede establecer cuán capaz es el sistema de cubrir la demanda hídrica de una planta sin producir derroches (Ali, 2010).

2.3.2.1 Definición y Fundamentos del Riego Inteligente

El riego inteligente se define como la incorporación de tecnologías sofisticadas para el control y la detección automatizados, los cuales son utilizados para administrar con precisión y en función del contexto el uso de agua en las cosechas. Este método, a diferencia de los tradicionales que se basan en la revisión visual subjetiva del agricultor o en calendarios fijos, toma sus determinaciones basándose en datos cuantitativos que se recogen en tiempo real acerca de las condiciones climáticas y edafológicas (suelo y clima). El objetivo principal es proporcionar el recurso hídrico exacto en el momento adecuado, de modo que se optimice la utilización de los insumos y se incremente al máximo la productividad dentro del marco de la agricultura de precisión (Gebbers, 2010).

La arquitectura de bucle cerrado (closed-loop system) es la base operativa de estos sistemas. Esta estructura une una red de sensores que se encuentran en el campo, con la responsabilidad de vigilar variables físicas esenciales como la temperatura del ambiente o la humedad del suelo, con un núcleo central de procesamiento que lleva a cabo algoritmos de decisión lógica. El sistema activa automáticamente los actuadores (bombas y electroválvulas) cuando los valores recogidos por los sensores bajan de los límites de seguridad establecidos con anterioridad para el cultivo, sin que sea necesaria la intervención directa del ser humano, lo cual asegura una reacción instantánea frente al estrés hídrico (Navulur et al., 2017).

La eficacia basada en la demanda biológica real, además de la automatización mecánica, es el soporte teórico del riego inteligente. El sistema tiene como objetivo armonizar la oferta de agua con el índice de evapotranspiración del cultivo, evitando así tanto el déficit que obstaculiza el desarrollo fisiológico como la sobreabundancia que genera asfixia radicular, lixiviación de nutrientes y derroche energético. El uso de esta tecnología hace posible que la gestión del agua pase de ser un proceso empírico y reactivo a uno previsor y efectivo, lo cual es fundamental

para lograr la sostenibilidad en contextos productivos intensivos como los viveros (Anci Yep & Hermosa Caceres, 2025).

2.3.2.2 Sensores y Adquisición de Datos Agronómicos (Humedad, Temperatura)

La variable esencial para calcular la programación del riego de manera eficaz es la medición exacta de la humedad del suelo, que se lleva a cabo utilizando sensores edáficos que funcionan como transductores con el fin de estimar el contenido volumétrico de agua. Los sensores capacitivos, que determinan la constante dieléctrica del entorno, son una de las tecnologías más utilizadas. Como el agua presenta una constante dieléctrica mucho más alta que la del aire o el suelo seco, el sensor es capaz de identificar alteraciones leves en la humedad al observar los cambios en la capacitancia eléctrica. Esta tecnología proporciona beneficios operativos más altos en comparación con los sensores resistivos convencionales, pues son menos propensos a la salinidad y corrosión del sustrato, lo que asegura lecturas estables durante un largo periodo (Vellidis et al., 2008).

Al mismo tiempo, para saber cuánta agua pierde la planta, es importante controlar el clima, especialmente el calor y la humedad del aire. Para conseguirlo se utilizan sensores digitales (como los modelos DHT o SHT), que contienen en un único componente protegido la tecnología necesaria para medir con precisión ambos valores. Cuando el sistema recibe estos datos, no sólo entiende cómo está el cultivo en ese momento, sino que también es capaz de anticiparse a posibles problemas de estrés térmico que podrían afectar al crecimiento del cacao. Esto se logra a través del ajuste de las dosis de riego con el fin de contrarrestar la pérdida natural de agua por transpiración durante las horas con más radiación solar (Ruiz-Garcia et al., 2009).

La fase de conversión y transmisión es la última del proceso de adquisición de datos (DAQ), en la que un microcontrolador central intercepta las señales eléctricas analógicas o digitales producidas por los sensores. Este aparato transforma la señal cruda, utiliza factores de corrección y la convierte en un valor numérico estandarizado (como el porcentaje de humedad, que puede ir del 0 al 100%). La información es empaquetada y enviada al servidor o a la nube de manera secuencial después de ser digitalizada. Esto garantiza que el algoritmo de aprendizaje automático tenga una base de datos histórica ordenada y sin errores para su entrenamiento y toma de decisiones (Foughali et al., 2018).

2.3.2.3 Actuadores en Sistemas de Riego (Válvulas Solenoides, Bombas)

En la estructura de un sistema de riego automatizado, los actuadores electromecánicos son el componente que ejecuta las acciones finales. Su función consiste en convertir las señales digitales de voltaje bajo que llegan del controlador en movimientos físicos que regulan el flujo del agua. Las electroválvulas o válvulas solenoides son los dispositivos más importantes para la zonificación del riego; operan a través de una bobina magnética que, cuando se energiza, eleva un émbolo interno y deja pasar el fluido. La aplicación de esto permite dividir la siembra en zonas hidráulicas autónomas, garantizando que solo se provea agua a las regiones que el algoritmo ha detectado que les falta agua, así se logra aprovechar mejor tanto la cantidad como la presión que se reparte (Dursun & Ozden, 2011).

Las bombas hidráulicas son las responsables de transportar el fluido desde la fuente de almacenamiento hasta los emisores. Estas cumplen el rol del corazón de la red de distribución, pues producen la carga de presión requerida. El control de estas bombas en los modernos sistemas de riego inteligente, que antes era binario (encendido/apagado), se ha vuelto más sofisticado y frecuentemente se combina con variadores de frecuencia que modifican la velocidad del motor dependiendo de la demanda real del caudal. Esta modulación es crucial para conservar una presión estable en las líneas de goteo, previniendo la ruptura de tuberías por sobrepresión y la distribución desigual debida a una potencia inadecuada (Kumar et al., 2013).

Los canales de comunicación, normalmente módulos de relés de estado sólido o contactores sirven para conectar de forma segura equipos de alta potencia a componentes electrónicos más sensibles (como sensores y microcontroladores). Dichos elementos aíslan el circuito de control (operado a 3,3 V o 5 V) frente a las líneas de alto voltaje (110 V o 220 V). Esta barrera resguarda al procesador de sobretensiones destructivas. Usualmente, las infraestructuras completamente automatizadas usan estos módulos para que el software encienda equipos de hardware pesado. Sin embargo, el enfoque local es distinto. El sistema inteligente (chatbot) únicamente evalúa la información capturada para despachar alertas predictivas y formular recomendaciones exactas. La ejecución final del riego y la manipulación de estos actuadores quedan bajo el control manual del operador, basándose en la información inteligente recibida (Gutiérrez, Villa-Medina, Nieto-Garibay, & Porta-Gándara, 2014).

2.3.2.4 Plataformas de Servicios en la Nube para IoT (AWS, Azure, Google Cloud)

La puesta en marcha de plataformas de computación en la nube (Cloud Computing) representa el eje central para que los sistemas de riego inteligente puedan escalar, ya que estas ofrecen la infraestructura requerida para almacenar, procesar e ingerir datos telemétricos producidos por los sensores. Servicios líderes como Google Cloud IoT Core, Servicios como Microsoft Azure IoT Hub o Amazon Web Services (AWS) IoT proporcionan una base de "Plataforma como servicio" (PaaS), que permite que toda la lógica del chatbot y los controles de sensores estén estrechamente acoplados a la nube sin tener que administrar físicamente estos servidores complejos, lo que hace innecesario el manejo de servidores físicos locales. Esto posibilita que el sistema aumente automáticamente sus recursos computacionales dependiendo del volumen de datos entrantes desde el vivero, lo cual asegura una disponibilidad elevada y una tolerancia a fallos (Botta et al., 2016).

Una función clave en estas plataformas es la gestión del estado de equipos mediante conceptos usando lo que se conoce como el estado virtual del equipo o Device Shadow, también gemelos digitales o digital twins. Esta característica técnica permite que el sistema sepa siempre como están los sensores en el vivero, incluso si la conexión falla, ya que el sistema guarda una versión virtual que siempre está al día para cada dispositivo. Esta herramienta realiza una copia física de cada sensor y actuador en Internet, mediante un archivo JSON que almacena los últimos datos enviados, como el nivel de humedad y la temperatura actual. Gracias a ello, el chatbot puede consultar estas variables instantáneamente desde la versión en la nube, sin depender de una conexión física con el equipo cada segundo. Si Internet falla por un momento, el sistema realiza ajustes automáticamente tan pronto como regresa la señal, asegurando que el trabajo en la guardería nunca se detenga (Stergiou et al., 2018).

Por último, la gran ventaja de estas plataformas es que permiten usar herramientas para analizar grandes cantidades de datos y sistemas que aprenden solos, al guardar toda la información del ambiente en la base de datos como BigQuery o DynamoDB es mucho más sencillo preparar los modelos de predicción que hacen que el chatbot mejore sus respuestas. Así, la nube no solo funciona como un repositorio pasivo, sino también como el motor analítico que ejecuta los algoritmos complejos para determinar la eficacia del riego y enviar al usuario final únicamente la inferencia procesada o la alerta simplificada (Hashem et al., 2015).

2.3.2.5 Técnicas de Programación de Riego (Umbral, Basado en Evapotranspiración)

La técnica reactiva de programar el riego en función de los límites de humedad del suelo toma como base la medición directa del estado del agua en la zona raíz. Este método define dos límites esenciales: el Punto de Marchitez Permanente (el límite inferior que indica la muerte de la planta) y la Capacidad de Campo (el límite superior que señala cuánta agua puede almacenar el suelo). El sistema automatizado funciona estableciendo un "Umbral de Gestión Permitido" (MAD, por sus siglas en inglés). Cuando los sensores advierten que la humedad baja por debajo del umbral fijado previamente, se inicia inmediatamente el riego para devolver el agua hasta su capacidad de campo. De este modo, se garantiza que las plantas nunca padezcan estrés hídrico fisiológico (Shock & Wang, 2011).

Por otra parte, la técnica basada en la evapotranspiración usa una forma de medir el equilibrio del agua según el clima. Con este proceso se calcula cuanta agua suelta el cultivo, sumando lo que sale por las hojas y lo que se evapora directamente desde la tierra. Este procedimiento utiliza, para calcular la Evapotranspiración del Cultivo (ET_c), la ecuación estándar de Penman Monteith (FAO-56) y se sirve de variables meteorológicas como son la temperatura, la velocidad del viento y la radiación solar. En este esquema, el sistema no mide directamente el suelo, sino que repone con exactitud el volumen de líquido que se "perdió" en la jornada anterior; es como una cuenta bancaria de agua donde las entradas y salidas están equilibradas (Allen et al., 1998).

En la actualidad, la agricultura de precisión tiene como objetivo fusionar estos métodos para ser más eficiente, en particular el riego controlado (CDR). El objetivo central es emplear el agua más eficazmente y, en ciertos casos, obtener una mayor cantidad de resultados en la planta. Para lograrlo, se disminuye intencionadamente la cantidad de agua suministrada, teniendo cuidado de no exceder lo que la planta respira en las fases de crecimiento, cuando es menos vulnerable a la falta de riego. La combinación del cálculo climático (ET) con la revisión del estado real del sensor permite que los sistemas de aprendizaje cambien de manera dinámica las dosis, previniendo el drenaje profundo no necesario y el estrés severo (Fereres & Soriano, 2007).

2.3.2.6 Eficiencia y Conservación del Recurso Hídrico

La eficiencia del uso de agua, abreviada como WUE en inglés, va más allá de la mera disminución del volumen utilizado; se establece como el aumento máximo de biomasa generada por cada unidad de agua transpirada dentro del contexto de la ingeniería agrícola contemporánea. Al modernizar el riego y cambiar los métodos antiguos de gravedad por sistemas a presión controlados con software, se consigue que la operación sea mucho más eficiente. Este avance tecnológico permite el manejo preciso del agua en el vivero, logrando que cada planta reciba el riego de manera justa y mejorando el rendimiento general de la producción. Cuando el agua se proporciona de forma directa en la zona de absorción de raíces y en las cantidades precisas que son necesarias, se reducen al mínimo las pérdidas por evaporación superficial y conducción. Esto hace posible que un porcentaje mucho mayor del recurso obtenido sea utilizado efectivamente por la planta (Playán & Mateos, 2006).

Ante la escasez de agua y la inestabilidad climática que se observan en todo el mundo, la conservación de los recursos hídricos es una prioridad inevitable. La introducción de tecnologías de riego inteligentes sirve como herramienta de mitigación ambiental, ya que previene la sobreexplotación de acuíferos y fuentes superficiales. Fuertes controles automáticos previenen la profundidad de la filtración, que ocurre cuando el exceso de agua (que a menudo contiene fertilizantes químicos) se filtra sobre las raíces y contamina los acuíferos. De este modo, la tecnología no solo garantiza que el agua esté disponible en el futuro para el vivero, sino también que la integridad ecológica del ecosistema circundante se mantenga (Pereira et al., 2002).

La meta última, desde el punto de vista de la productividad hídrica, es disminuir la diferencia entre la demanda neta y bruta del cultivo. Los sistemas de monitoreo constante posibilitan la detección y eliminación del uso de agua "no beneficioso", que se define como el que no favorece el crecimiento de la planta. La incorporación de algoritmos de decisión y sensores de humedad permite una regulación dinámica que suprime los riesgos innecesarios comunes en la gestión manual (por ejemplo, regar antes de llover o cuando el terreno todavía posee suficiente humedad), lo cual convierte cada litro de agua ahorrado en un descenso directo de los costos energéticos y operativos (Molden et al., 2010).

2.3.2.7 Retos de la Implementación de IoT en la Agricultura

La infraestructura de conectividad insuficiente en las áreas rurales es uno de los retos más importantes en la implementación de redes IoT en contextos agrícolas. A diferencia de las ciudades, las zonas de cultivo suelen carecer de internet de banda ancha o de una señal celular estable, dificultando que los datos suban a la nube de manera continua. Aunque existen tecnologías de gran alcance y poco consumo (LPWAN), como LoRaWAN o Sigfox, que sirven para mandar datos pequeños a muchos kilómetros, estas todavía tienen fallas importantes si se necesita rapidez o enviar mucha información al mismo tiempo. Esto puede limitar la posibilidad de implementar sistemas que necesiten video en tiempo real o control de actuadores con respuesta milimétrica (Elijah et al., 2018).

Un segundo desafío técnico importante es la durabilidad del hardware expuesto y la gestión energética. Los nodos de sensores que se instalan en un vivero o plantación al aire libre tienen que funcionar bajo condiciones ambientales adversas, resistiendo radiación UV fuerte, humedad elevada, polvo y productos agroquímicos corrosivos que deterioran las piezas electrónicas. Asimismo, la mayor parte de estos equipos se basa en baterías que no es posible reemplazar manualmente en centenares de lugares por razones logísticas. Esto lleva a los especialistas a buscar formas de aprovechar la energía del entorno usando por ejemplo paneles solares pequeños, lo cual aumenta el coste inicial y la dificultad del mantenimiento del sistema (Tzounis et al., 2017).

Por último, la privacidad de los datos y la seguridad de la información son obstáculos cada vez mayores a medida que se digitaliza el sector agrícola. La arquitectura del Internet de las cosas (IoT), que se distribuye a través de varios puntos de acceso físicos (sensores) y lógicos (API en la nube), incrementa el área susceptible a ataques cibernéticos. La toma de información agronómica delicada o, incluso peor, el apoderamiento del manejo de sistemas de riego automatizados (hackeo de actuadores) tiene la capacidad de arruinar una producción completa. Asegurar protocolos de encriptación sólidos y autenticación segura en dispositivos con escasa potencia computacional continúa siendo un reto abierto para la investigación, el cual busca preservar la integridad del cultivo inteligente (Ferrag et al., 2017).

2.3.2.8 Cultivo de Theobroma Cacao y sus Requerimientos Hídricos

El *Theobroma cacao* L. es una especie perenne que se originó en la cuenca amazónica y tiene adaptaciones fisiológicas para los entornos de bosque húmedo tropical, donde las temperaturas son cálidas y la humedad relativa es elevada. Para su óptimo desarrollo, esta planta necesita precipitaciones anuales repartidas de manera homogénea entre 1,500 y 2,500 mm y una temperatura media que varíe entre los 22 °C y los 30 °C. Posee un bajo rendimiento intrínseco en el uso del agua a nivel foliar debido a una débil regulación estomática frente a cambios drásticos en el déficit de presión de vapor, lo que la vuelve muy propensa al estrés hídrico si la demanda atmosférica excede la disponibilidad inmediata de agua en el terreno (Carr & Lockwood, 2011).

En lo que respecta a la morfología de la raíz, el cacao tiene un sistema radicular superficial, con casi todos sus pelos absorbentes ubicados en los primeros 30 a 40 centímetros del perfil del suelo. Esta configuración radicular limita el acceso de la planta a la humedad de las capas profundas, obligándonos a vigilar que la superficie nunca se seque por completo. Si el riego falla, el cacao entra en crisis rápidamente: detiene su alimentación, bloquea la transpiración en sus hojas y deja morir a los frutos jóvenes o 'cherelles'. Al final, este descuido en el riego se traduce en granos más pequeños y una producción mucho más pobre, lo que justifica la necesidad de un sistema de control automático que actúe a tiempo (Lahive et al., 2019).

En cuanto al manejo técnico del riego, los requerimientos de agua del cacao no son fijos, sino que cambian dependiendo de la fase fenológica. Se presentan en su punto más alto durante el llenado de los frutos y la floración. Se debe conservar la humedad en el cultivo dentro de un rango próximo a la capacidad de campo, según lo indica la literatura técnica, evitando que el agua disponible en el suelo se acabe más allá del 30-40%. Para prevenir el encharcamiento, es necesario manejar el riego de manera precisa, porque el cacao no tolera la hipoxia radicular (escasez de oxígeno) y que la humedad en exceso favorece que se multipliquen patógenos fúngicos destructivos como la *Phytophthora* (Wood & Lass, 2008).

2.3.3 Metodología de desarrollo: Scrum

Para la realización del proyecto actual, he elegido Scrum como metodología. Esta es un enfoque ágil que se ha diseñado para resolver problemas complejos y adaptativos, a la vez que

se producen productos de gran valor de manera creativa y productiva. A diferencia de los métodos tradicionales en cascada, que tienen una planificación predecible y rígida, Scrum está basado en el "empirismo" o teoría del control de procesos empíricos. Bajo esta metodología, el aprendizaje se construye mediante la práctica y las decisiones se toman analizando los resultados obtenidos en el campo. De esta forma la metodología Scrum aplica un proceso de avances pequeños y seguidos para ver los fallos antes de que ocurran y evitar problemas a futuro, organizando todas las tareas en ciclos de desarrollo cortos y bien definidos (Schwaber, 2020).

Para que el control empírico funcione en scrum, es necesario apoyarse en tres bases importantes, mantener la transparencia en el proceso, realizar inspecciones constantes y tener la capacidad de adaptación frente a los cambios. La transparencia asegura que los elementos importantes del proceso sean perceptibles para quienes están a cargo del resultado; la inspección requiere que los usuarios revisen regularmente el progreso y los artefactos para identificar cambios no deseados; y la adaptación establece que, si se descubre que un elemento del proceso excede los límites permitidos, es necesario modificar el proceso o el material en cuestión con prontitud para reducir las desviaciones en el futuro. Estos pilares se concretan mediante eventos formales y artefactos particulares que organizan el flujo de trabajo del equipo (Rubin, 2012).

La metodología establece tres roles o responsabilidades a nivel estructural: el Product Owner (Dueño del Producto), cuya misión es maximizar el valor del producto y gestionar la Lista de Producto (Backlog); el Scrum Master, que se desempeña como un líder servicial para garantizar la comprensión y aplicación de la metodología, eliminando obstáculos; y el equipo de desarrolladores, conformado por un grupo multifuncional y autoorganizado que tiene la responsabilidad de generar al final de cada ciclo un incremento del producto "Terminado" y utilizable. Esta distribución de papeles suprime la jerarquía de mando y control tradicional, promoviendo que se comparta la responsabilidad y que se tomen decisiones técnicas con rapidez (Rubin, 2012).

En Scrum, el ciclo de vida de desarrollo se lleva a cabo en bloques temporales fijos llamados Sprints, que generalmente duran entre una y cuatro semanas. Cada Sprint comienza con la planificación del Sprint, en la cual se escoge el trabajo a efectuar; sigue con el Daily Scrum, una reunión de 15 minutos para ponerse de acuerdo sobre las tareas del día y es supervisado

diariamente; y termina con dos eventos finales: la revisión del Sprint, donde se examina el incremento de software funcional junto a los interesados (en este caso, lo que hace el chatbot), y la retrospectiva del Sprint, centrada en mejorar constantemente el proceso del equipo. Este ciclo reiterativo posibilita que el proyecto se ajuste pronto a modificaciones en los requisitos o a nuevas exigencias localizadas en el vivero, garantizando que el software final satisfaga efectivamente lo que espera el usuario (Schwaber, 2020).

2.4 Conclusiones del marco teórico

Este capítulo ha posibilitado que se definan los requisitos necesarios para el proyecto y que se establezcan los fundamentos esenciales de la ingeniería del software. Al explicar estos conceptos de manera clara, se consigue un entendimiento mutuo que es esencial para prevenir malentendidos en el momento de realizar la tarea. Esta información no únicamente explica los términos técnicos del chatbot, además establece de forma ordenada los pasos para crear el programa. De este modo, se consigue que la tecnología satisfaga las verdaderas necesidades del vivero y se garantice que el sistema sea capaz de desarrollarse y emplearse con éxito.

La metodología de Cascada se caracteriza por ser muy rígida y seguir un orden estricto, lo que dificulta la realización de cambios. Este rasgo supone un inconveniente para esta investigación, puesto que, durante el desarrollo tecnológico, suelen surgir requisitos nuevos o imprevistos que requieren atención. Por lo tanto, es razonable dejar de lado este modelo lineal en favor de métodos de trabajo más flexibles, ya que el entorno del proyecto exige la habilidad de adaptarse a circunstancias imprevistas para no comprometer el progreso alcanzado.

La evaluación de otros modelos, como el de Prototipos y el Incremental, evidenció que son alternativas más convenientes cuando se requiere obtener resultados veloces y recibir retroalimentación continua. Dado que posibilitan realizar pruebas de secciones del sistema con el personal del vivero conforme avanza el desarrollo, estos métodos son muy útiles para este proyecto. Su uso contribuye a disminuir de manera importante el peligro de incurrir en errores serios al final, puesto que se tiene la posibilidad de corregir las fallas gradualmente y garantizar que el producto final cumpla verdaderamente con el propósito de mejorar el riego.

En última instancia, el análisis de las metodologías ágiles, en particular Scrum y la Programación Extrema, verifica que son las más apropiadas para tareas que necesitan ajustarse

a modificaciones y cooperar estrechamente. Estos marcos de trabajo permiten la flexibilidad que se requiere para crear software en un ambiente dinámico y variable. Todo este análisis teórico avala la elección de emplear una metodología mixta en la propuesta, fusionando una planificación inicial nítida con la habilidad de perfeccionar el sistema constantemente en función de los resultados.

CAPÍTULO III

3. MARCO INVESTIGATIVO

3.1 Introducción

En este capítulo se describe con detalle la metodología que orienta el desarrollo de la investigación científica propuesta. Se presentan las bases de los procedimientos elegidos para analizar el problema de la ineficiencia del riego en el Vivero Lastenia Vera, respaldando cada elección con la literatura académica más reciente. Esta sección facilita la comprensión de la secuencia lógica del estudio, desde que se identificó la problemática hasta la recolección

empírica de información, garantizando que los resultados sean lo suficientemente rigurosos para su validación profesional e institucional. La estructura incluye la determinación del tipo de estudio, los enfoques lógicos utilizados para examinar la situación y las técnicas de campo empleadas para obtener los datos que fundamentan la investigación.

3.2 Tipos de investigación

3.2.1 Investigación de campo

De acuerdo con Arias Gonzáles (2020), la investigación de campo es aquella que se lleva a cabo en el mismo lugar donde suceden los hechos, lo que posibilita al investigador acceder a datos primarios de la realidad sin alterar las variables del entorno. Esta metodología facilita el trato directo con el entorno que se investiga, garantiza que los datos obtenidos reflejen la realidad actual del campo sin basarse solo en teoría o información de terceros, así logrando que el análisis inicial sea mucho más preciso a la situación real.

Para el desarrollo de este proyecto, se aplicó la investigación de campo mediante visitas técnicas a las áreas del vivero Lastenia de la ULEAM. Este trabajo presencial permitió observar directamente las condiciones del lugar y recolectar los datos necesarios para el diseño del sistema. Es esencial este tipo de investigación para observar de manera directa la forma en que los empleados llevan a cabo el riego manual en la actualidad, detectar las áreas con más derroche de agua y recopilar información concreta acerca del estado físico del cultivo de cacao; todo esto será fundamental para establecer los sensores y la lógica del chatbot.

3.2.2 Investigación descriptiva

De acuerdo con Medina Romero et al. (2023), la investigación descriptiva busca detallar cómo son y cómo se comportan las características más importantes de los que se está estudiando. En lugar de solo juntar datos, este método trata de medir distintos puntos del problema para dar una imagen clara de la realidad, esto permite tener una base para explicar los resultados más adelante o proponer soluciones.

En este proyecto, se utilizó este tipo de investigación para explicar exactamente cómo se está manejando el riego en el cultivo de cacao actualmente, con esto se detallaron puntos clave como cuánta humedad tiene la tierra, cada cuanto se riega y que problemas tiene el personal

para realizar el trabajo. Así se obtuvo un punto de partida claro para demostrar por qué es necesario automatizar y cuáles son los niveles exactos que el sistema inteligente tendrá que controlar.

3.2.3 Investigación aplicada

Tal como explican Ñaupas Paitán et al. (2023), el interés de la investigación aplicada en el uso de saberes científicos y teóricos para solucionar problemas concretos que impactan a la sociedad o al sector productivo es lo que la caracteriza. Este enfoque está centrado en la utilidad y en el desarrollo de tecnologías, tácticas o herramientas que convierten una realidad problemática en una solución concreta y funcional, en contraste con la investigación básica, cuyo objetivo es expandir el conocimiento puro.

Para este trabajo de titulación, lo más importante es la investigación aplicada, ya que tomó los conceptos de inteligencia artificial y agronomía para construir algo real, un chatbot que aprende solo. No se limitó a estudiar el problema, sino que se usó la ingeniería para diseñar y activar una herramienta que resuelva de verdad el mal uso del agua. El objetivo es que el vivero aproveche mejor sus recursos y que la producción de cacao mejore gracias a este sistema.

3.2.4 Métodos de investigación

3.2.4.1 Método inductivo-deductivo

El método inductivo-deductivo se fundamenta en la interacción lógica entre dos formas de razonamiento opuestas pero complementarias que permiten la construcción del conocimiento científico. Según Ñaupas Paitán et al. (2023), la inducción comienza con la minuciosa observación de casos particulares para llegar a conclusiones o leyes generales, y la deducción utiliza principios teóricos universales para aplicarlos o explicarlos en circunstancias concretas.

El método inductivo-deductivo se fundamenta en la interacción lógica entre dos formas de razonamiento opuestas pero complementarias que permiten la construcción rigurosa del conocimiento científico. Según Ñaupas Paitán et al. (2023), la inducción comienza con la minuciosa observación de casos particulares para llegar a conclusiones o leyes generales, mientras que la deducción utiliza principios teóricos universales para aplicarlos o explicarlos

en circunstancias concretas. Esta perspectiva metodológica posibilita que el investigador transite de manera dinámica desde los sucesos empíricos observados en el entorno de campo hasta la formalización conceptual, permitiendo posteriormente aplicar dichos constructos teóricos para contrastar y validar la operatividad práctica de la solución tecnológica implementada.

Para este proyecto, se empleó la fase inductiva al comienzo de la investigación. Se examinaron los problemas particulares que sucedían todos los días con el riego de cada planta de cacao en el vivero, lo que permitió determinar que había una deficiencia generalizada en la gestión del agua. Más adelante, se aplicó la fase deductiva para utilizar específicamente los principios universales de la teoría de Aprendizaje Automático Supervisado y de la Agricultura de Precisión en la resolución del problema local. Esto implicó ajustar estas tecnologías globales a las exigencias específicas del vivero Lastenia Vera.

3.2.4.2 Método analítico-sintético

El método analítico-sintético funciona dividiendo el problema en partes para estudiar cada detalle por separado y luego unirlos para entender el sistema completo. Como señala Arias Gonzáles (2020), analizar es básicamente desarmar los elementos de un todo para revisarlos a fondo. Después viene la síntesis que consiste en juntar esas piezas de forma lógica para encontrar conexiones y detalles que no se veían cuando las partes estaban sueltas.

Al momento de desarrollar el chatbot, se utilizó este método como base para diseñar el algoritmo de control. Con el análisis, se revisaron individualmente los datos clave del cultivo, como la humedad en la tierra, el calor del ambiente y la cantidad de agua que necesitaba el cacao según su etapa de crecimiento. Posteriormente, mediante la síntesis, se integró toda esa información dentro del código para que el sistema aprendiera a interpretar los datos en conjunto. Esto permitió que el software tomara decisiones de riego lógicas que aprovecharan mejor los recursos.

3.3 Fuentes de información de datos

3.3.1 Fuentes primarias

3.3.1.1 Encuestas

La encuesta se define, según Medina Romero et al. (2023), como un método cuantitativo de investigación de campo que hace posible la recopilación de datos estandarizados mediante una muestra representativa de individuos. Su objetivo principal es adquirir información acerca de los puntos de vista, actitudes, comportamientos o conocimientos de un grupo particular a través de la realización de preguntas estructuradas. Esto permite que los resultados se puedan tratar posteriormente desde una perspectiva estadística para determinar patrones o tendencias objetivas sobre el fenómeno estudiado.

La encuesta se aplicó a los alumnos de la carrera de Ingeniería Agropecuaria de la ULEAM, dentro del marco de este estudio. Este grupo fue elegido porque se encarga directamente del trabajo manual de riego y del cuidado de las plantas de cacao durante sus labores en el vivero Lastenia. El cuestionario trató temas esenciales, tales como la regularidad con que llevaban a cabo el riego, los criterios empíricos que empleaban para establecer cuántos litros de agua eran requeridos, las complicaciones operativas con las que lidiaban regularmente y su grado de disposición para incorporar nuevas herramientas tecnológicas.

3.3.1.2 Entrevista

De acuerdo con Arias Gonzáles (2020), la entrevista es un método cualitativo de recolección de datos fundamentado en una conversación directa y estructurada entre el investigador y el sujeto de estudio. A diferencia de los formularios cerrados, este enfoque permite conocer a fondo las experiencias y conocimientos de las personas que participan. De esta manera, se obtienen datos exactos y en constante movimiento, lo cual es muy importante para descubrir las causas reales del problema que se estudia.

Para realizar esta investigación, fue necesario elegir a un solo informante clave con mucha formación técnica: el ingeniero agrónomo encargado del vivero Lastenia de la ULEAM. Este especialista fue elegido porque conoce muy bien lo que el cacao necesita para crecer y tiene mucha experiencia trabajando en el campo. El diálogo que se llevó a cabo se centró en temas

importantes como el agua y el medio ambiente. Se analizaron las mejores formas de regar las plantas, los errores comunes que se cometen al hacerlo a mano y cómo el clima puede afectar el crecimiento y la supervivencia de las plántulas de cacao.

3.4 Estrategia operacional para la recolección de datos

3.4.1 Población

Según Arias Gonzáles (2020), la población se define como el conjunto total finito o infinito de elementos, seres o eventos que comparten características comunes y observables, sobre los cuales se pretende extender las conclusiones de la investigación. Este macro-grupo constituye la totalidad del fenómeno a estudiar y sirve como marco de referencia para seleccionar, en caso de ser necesario, una muestra representativa que permita obtener datos generalizables sin necesidad de abordar a cada individuo.

La población que participó en este estudio estuvo conformada por un total de 89 personas, integrando a los estudiantes de la carrera de Ingeniería Agropecuaria que realizaban sus labores en el vivero Lastenia Vera de la ULEAM extensión El Carmen, y a los encargados operativos que cuidaban los cultivos de cacao todos los días. Se consideró a este conjunto de individuos como la fuente idónea de información, dado que interactuaban cotidianamente con la problemática del riego manual que se buscaba solucionar.

3.4.2 Muestra

Según Arias Gonzáles (2020), la muestra se caracteriza por ser un subconjunto representativo de la población total. Sin embargo, en investigaciones donde la población es reducida y de fácil acceso, es factible y metodológicamente más riguroso prescindir del cálculo muestral para trabajar con la totalidad de los elementos (censo).

En este estudio, dado que la población total bajo análisis es pequeña (constituida por 89 personas), no se aplicó una fórmula de muestreo probabilístico. En su lugar, se adoptó un enfoque censal, abarcando el 100% de la población objetivo. De esta manera, la recolección de datos mediante la encuesta se aplicó a la totalidad de los estudiantes de la carrera de Ingeniería Agropecuaria y al equipo técnico que realizaban sus actividades en el Vivero Lastenia,

logrando cubrir el universo completo de estudio sin necesidad de estimaciones por subconjuntos.

3.4.3 Análisis de las herramientas de recolección de datos a utilizar

3.4.3.1 Encuesta

Enfocar el estudio en estos perfiles hizo posible recopilar la experiencia real de las personas que interactúan diariamente con el cultivo de cacao. Este acercamiento directo resultó clave para asegurar que las características del software se ajusten con exactitud a las verdaderas necesidades del campo y al nivel de habilidad tecnológica de los usuarios. Para ordenar la recolección de datos, el cuestionario se estructuró en diferentes áreas. La primera se enfocó en las responsabilidades diarias de los trabajadores y como aplican el riego. A continuación, se aborda la forma en la que manejan el agua y que tan dispuestos están en usar nuevas soluciones digitales. En la parte final, la encuesta evalúa los conocimientos del equipo de control del clima en el vivero. También se aprovechó estas mismas preguntas para calcular el tiempo exacto que invierten revisando los cultivos a mano y para hacer un inventario de los equipos que tienen a su alcance durante los días de trabajo. Dicha disposición metodológica facilita la evaluación de viabilidad para integrar un modelo de aprendizaje automático supervisado, dotando de respaldo técnico la selección de funciones que el asistente virtual ejecutará para potenciar la eficiencia hídrica.

3.4.3.2 Entrevista

Para complementar la información cuantitativa, se determinó el uso de una entrevista como herramienta cualitativa de recolección de datos. El análisis de esta técnica demuestra que es el instrumento idóneo para profundizar en fenómenos complejos que no pueden medirse de manera adecuada con encuestas cerradas, tales como las variables biológicas del cultivo y las fallas de la logística en el vivero. Su aplicación permite extraer conocimiento empírico y técnico directamente del personal experto encargado del área, obteniendo un panorama detallado sobre los requerimientos hídricos reales. Esta recolección cualitativa es vital para la investigación ya que permite fundamentar los parámetros agronómicos exactos que el asistente virtual necesitará para emitir diagnósticos predictivos y confiables.

3.4.3.3 Estructura de los instrumentos de recolección de datos aplicados

3.4.3.3.1 Estructura de encuesta

La encuesta está estructurada como un instrumento cuantitativo compuesto por un total de 30 reactivos, diseñados para recopilar datos estadísticos precisos sobre la realidad operativa del vivero. El instrumento de evaluación se administró de forma presencial a los estudiantes que cursaban el segundo y quinto semestre de la carrera de Ingeniería Agropecuaria. De igual manera, la consulta abarcó al equipo técnico y a los trabajadores operativos del vivero Lastenia. El formato detallado y la estructura completa de este cuestionario utilizado para la recolección de datos cuantitativos se pueden observar en el **Anexo E**. Asimismo, las evidencias de la aplicación presencial de este instrumento a los estudiantes se encuentran adjuntas en el **Anexo C**.

3.4.3.3.2 Estructura de entrevista

La entrevista se diseñó como un instrumento de tipo cualitativo y semiestructurado, dirigido específicamente al Ingeniero Agrónomo responsable técnico del vivero Lastenia. Elegir a esta persona especialista hizo posible reunir los detalles técnicos de manera muy precisa, que servirán como base directa para configurar los parámetros del software. Durante la conversación se definieron las necesidades del cacao y se documentaron los fallos en las rutinas diarias de riego. En conjunto, estos datos forman la lógica para programar las funciones del chatbot y aportan el material necesario para entrenar correctamente el modelo de aprendizaje automático supervisado.

El formulario estructura sus 13 preguntas clave rigiéndose por una guía técnica validada. Estas interrogantes se distribuyen estratégicamente en tres áreas temáticas: diagnóstico del problema, parámetros técnicos ambientales y gestión de la información. Al principio, el cuestionario explora lo que sucede cuando hay estrés hídrico y evidencia los problemas del equipo que se utiliza normalmente. Luego, la segunda parte establece los niveles exactos de temperatura y humedad que se necesitan para cuidar el cultivo. Finalmente, la última parte revisa los registros manuales del personal, la forma en que se emiten avisos y las consecuencias de no contar con respaldos digitales. La estructura detallada de este instrumento cualitativo se encuentra disponible para su revisión integral en el **Anexo D**.

3.4.4 Plan de recolección de datos

Tabla 1 Plan de recolección de datos

ACTIVIDAD	RESPONSABLE	FECHA DE EJECUCIÓN
Entrevista al ingeniero agrónomo responsable del vivero (Ing. Pedro Nivelá).	Kevin David Rodríguez Ordoñez Dereck Elian Castro Segovia David Ricardo Gerrero Salvatierra Anderson Steven Plua Chamorro Anthony Javier Zambrano Alcivar Bryan David López Lóor	15/04/2026 – 12:00
Encuesta a los estudiantes de la Uleam extensión El Carmen, pertenecientes a la carrera de agropecuaria de los niveles 1ero y 5to	Kevin David Rodríguez Ordoñez Anderson Steven Plua Chamorro Dereck Elian Castro Segovia David Ricardo Gerrero Salvatierra Anthony Javier Zambrano Alcivar Bryan David López Lóor	28/01/2026 – 21/04/2026

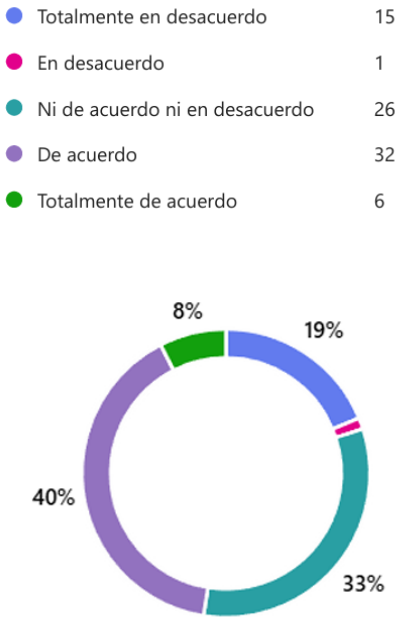
3.5 Análisis y presentación de resultados

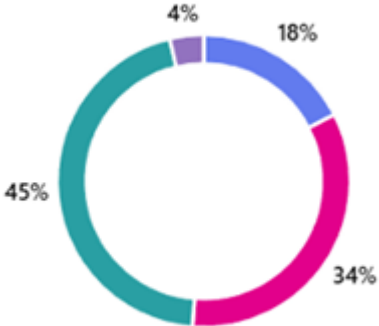
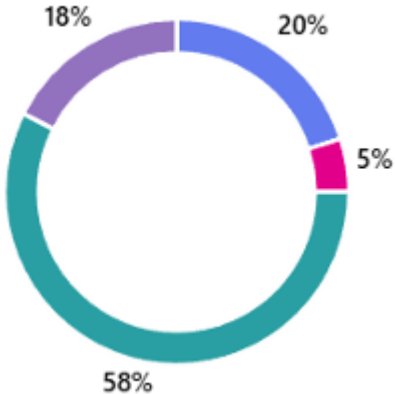
3.5.1 Presentación y descripción de los resultados obtenidos

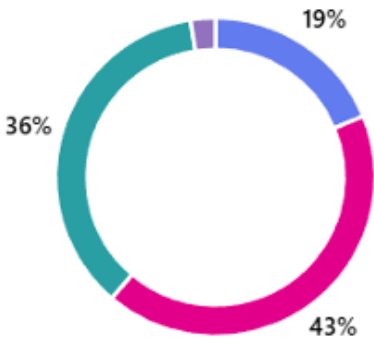
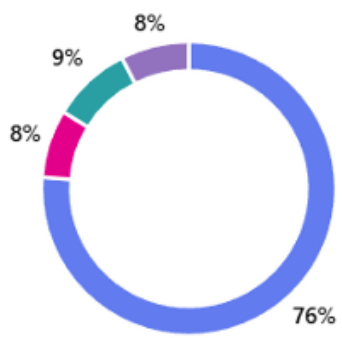
3.5.1.1 Análisis de encuesta dirigida a estudiantes Agropecuaria de la Uleam extensión

El Carmen

Tabla 2 Resultados obtenidos en la encuesta a estudiantes de Agropecuaria

PREGUNTA	GRÁFICO	INTERPRETACIÓN
1. ¿Considera usted que el estrés hídrico causado por el excesivo de agua o la falta de esta en las plantas de cacao se presenta debido a la falta de información oportuna sobre las condiciones climáticas?	 ● Totalmente en desacuerdo 15 ● En desacuerdo 1 ● Ni de acuerdo ni en desacuerdo 26 ● De acuerdo 32 ● Totalmente de acuerdo 6	El 48% del personal reconoce que el estrés hídrico en las plantas se debe a la falta de información climática oportuna. Este resultado evidencia que la observación empírica es insuficiente, justificando la necesidad de implementar una red de sensores IoT que capture datos ambientales exactos en tiempo real para optimizar las decisiones de riego.

PREGUNTA	GRÁFICO	INTERPRETACIÓN
2. ¿Cree usted que el registro manual de datos de riego y clima aumenta el riesgo de cometer errores en la información recolectada?	- Muy de acuerdo 14 - De acuerdo 27 - Neutral 36 - En desacuerdo 3 - Muy en desacuerdo 0 	Un 52% de los encuestados afirma que el registro manual de datos (bitácoras) aumenta el riesgo de cometer errores humanos. Esto demuestra una vulnerabilidad en la integridad de la información actual, lo cual respalda directamente la propuesta de automatizar la recolección de telemetría hacia una base de datos digital para asegurar un historial exacto y sin sesgos.
3. ¿De qué manera considera que influye el acceso a información ambiental en tiempo real sobre la toma de decisiones fitosanitarias en el vivero?	- Complica la toma de decisiones 16 - No influye 4 - Facilita moderadamente la toma de decisiones 46 - Facilita significativamente la toma de decisiones 14 	Aunque un contundente 76% afirma que los datos en tiempo real facilitarían sus decisiones, un 20% percibe que esto podría "complicar" su trabajo. Esto justifica el uso de un chatbot inteligente: en lugar de abrumar al agricultor con gráficos complejos o datos crudos, el asistente procesará la información y emitirá únicamente alertas y recomendaciones sencillas en lenguaje natural.

PREGUNTA	GRÁFICO	INTERPRETACIÓN
4.¿Cree que la baja adopción de tecnologías (como el IoT) se debe principalmente a la falta de capacitación técnica en el sector?	● Muy de acuerdo 15 ● De acuerdo 34 ● Neutral 29 ● En desacuerdo 2 ● Muy en desacuerdo 0 	El 62% del personal indica que la falta de capacitación es la barrera principal para adoptar nuevas tecnologías en el agro. Este hallazgo sustenta firmemente la elección de una interfaz conversacional, ya que al permitir que el operador se comunique mediante mensajes de texto comunes (Procesamiento de Lenguaje Natural), se elimina por completo la curva de aprendizaje técnico.
5.¿Qué dispositivo utiliza con más frecuencia durante su jornada laboral?	● Teléfono inteligente (Smartphone) 61 ● Tablet 6 ● Computadora 7 ● Ninguno 6 	El 76% de los encuestados utiliza su teléfono inteligente de manera constante durante su jornada en el vivero. Este dato técnico determina que el entorno de despliegue del sistema debe ser obligatoriamente una aplicación móvil, garantizando que el personal reciba las notificaciones de estrés hídrico directamente en el lugar de trabajo.

3.5.1.2 Análisis de entrevista dirigida a Coordinador de Vivero Lastenia Vera

Tabla 3 Análisis de entrevista a coordinador

PREGUNTA	RESPUESTA	INTERPRETACIÓN
1.¿Cuáles son los principales problemas que enfrenta el Vivero Lastenia relacionados con la gestión del riego y demás variables ambientales en el cacao?	"El problema más grande es la disponibilidad hídrica. No tenemos una fuente hídrica primaria como un río, sino pozos artesanales y tenemos que bombear a cisternas. Eso se ha convertido en un problema por los altos costos operativos, principalmente el combustible de las bombas"	Según el coordinador, el verdadero reto radica en el elevado costo que implica extraer agua. Semejante dificultad justifica plenamente automatizar el proceso. Así, el modelo predictivo asegurará el encendido del riego solo ante una demanda real de humedad, bloqueando el derroche hídrico y abaratando el consumo de combustible en las bombas.
2. ¿Desde su experiencia técnica, ¿cuáles son los rangos óptimos de temperatura y humedad que garantizan la supervivencia de la plántula de cacao en su fase inicial?	"Hay un balance entre humedad y temperatura en el sustrato. Al ser una planta tropical, debemos tener un rango entre 28 a 30 grados Celsius, y con un porcentaje de humedad relativa no mayor al 75% para evitar plagas y enfermedades. La planta necesita 'capacidad de	La obtención de estos valores precisos (28-30 °C y <75% de humedad relativa) es un requerimiento funcional crítico para el diseño del sistema. En el contexto de la ingeniería de software, estos rangos no son simples datos agronómicos, sino que se convertirán en los umbrales lógicos (condicionales) programados en el

PREGUNTA	RESPUESTA	INTERPRETACIÓN
	campo', que es el balance entre oxígeno y agua".	microcontrolador y en el algoritmo de Machine Learning. El sistema utilizará estas métricas exactas como línea base para evaluar si la planta está en riesgo de estrés hídrico y determinar el momento exacto para enviar alertas preventivas a través del Chatbot.
3.¿Qué indicadores o rasgos físicos en las plántulas le permiten determinar que la gestión de datos ambientales está siendo exitosa o deficiente?	"A simple vista nosotros podemos observar cuando la planta necesita agua por su color o brillo. Si tuviéramos herramientas como sensores que puedan determinarlo, sería ideal. Todo es manual, no tenemos ningún software que nos ayude ni ningún sensor".	El experto confirma que el monitoreo actual de los cultivos depende exclusivamente del empirismo y del "ojo clínico" del personal, careciendo por completo de infraestructura tecnológica. Esta respuesta justifica de manera directa la integración de la capa de Internet de las Cosas (IoT). Al no existir sensores, la detección de anomalías es reactiva (cuando la planta ya muestra daño visual). La implementación de sensores transformará esta subjetividad en datos cuantitativos procesables las

PREGUNTA	RESPUESTA	INTERPRETACIÓN
		24 horas del día, eliminando la dependencia del factor visual humano.
4.¿Qué datos históricos o registros maneja el vivero sobre el consumo de agua, temperatura y demás datos ambientales, así como la mortalidad de plántulas por causas hídricas?	"Manejamos una bitácora en donde registramos todos los tratamientos físicos que vamos haciendo para hacer proyecciones. Todo es manual en cuadernos, no tenemos un software, en parte por la falta de recursos del sector público para implementarlo".	Dicha contestación expone una falla estructural profunda en el manejo de registros, pues la información clave descansa en soportes físicos (cuadernos), anulando cualquier posibilidad de análisis masivo. Informáticamente hablando, carecer de un repositorio digital frena por completo el uso de la ciencia de datos al momento de decidir en el campo. La presente propuesta soluciona el obstáculo vinculando los sensores a una plataforma en la nube (tipo Firebase o PostgreSQL). Así, automatiza expedientes digitales seguros y accesibles, liberando al vivero de costosos gastos en servidores físicos.

PREGUNTA	RESPUESTA	INTERPRETACIÓN
5.¿Qué consecuencias puede traer el desconocimiento de los rangos óptimos de riego y cómo afectan la salud de las plántulas de Theobroma cacao?	"Si usted tiene las condiciones ideales de agua y oxígeno, las plantas se ven vigorosas. Esto ayuda a que el patrón (tallo) engruese más rápido, alcanzando el estado ideal para el proceso de clonación y posterior entrega a los productores. Sin ese control, hay pérdidas".	El entrevistado concluye que el control exacto del riego no solo previene la muerte de la planta, sino que acelera su crecimiento comercial (engrosamiento del tallo). Esto eleva el impacto del software propuesto: el Chatbot inteligente no solo actuará como una herramienta de prevención de pérdidas, sino como un sistema de optimización de la producción agrícola. Al asegurar que las plántulas se mantengan siempre en sus rangos ideales mediante alertas automatizadas, el software incidirá directamente en la mejora de la calidad y los tiempos de entrega del producto final a los agricultores de la zona.

3.5.2 Informe final del análisis de los datos

El análisis integral de los datos recopilados permite demostrar de manera concluyente la existencia de las causas que originan el estrés hídrico en el Vivero Lastenia. Por un lado, las

encuestas evidenciaron que el 48% del personal asocia la mortalidad de las plántulas a la falta de información climática oportuna, lo que confirma la inaccesibilidad a datos técnicos como una causa latente. Asimismo, el hecho de que un 76% afirme que disponer de información en tiempo real facilitaría sus decisiones, valida la existencia de una infraestructura deficiente para el monitoreo. Por otro lado, en la entrevista, el experto agrónomo corroboró la dependencia exclusiva de la observación visual al señalar que actualmente la revisión es totalmente empírica (evaluando a simple vista el color o brillo de la hoja), a pesar de que el cacao requiere un rigor técnico estricto de 28 a 30 °C y menos del 75% de humedad. El contraste de estas herramientas de recolección evidencia la vulnerabilidad operativa comprobada: ante la ausencia de mediciones cuantitativas, el riesgo queda sujeto exclusivamente a las decisiones y percepciones subjetivas de los trabajadores. En conclusión, los datos ratifican que la inspección empírica es un proceso inexacto para asegurar la estabilidad biológica del cultivo, siendo esta la causa principal que desencadena un manejo inadecuado del recurso hídrico.

En un segundo ámbito, los datos recolectados validan la ausencia de registros históricos confiables como otra de las causas fundamentales del problema. La estadística de las encuestas señala que el 52% de los colaboradores asocia el registro manual con un incremento de equivocaciones. Estos testimonios concuerdan con la postura del coordinador, quien confirma que el uso exclusivo de libretas de papel refleja una carencia absoluta de dotación tecnológica. Esta situación genera discrepancias en los apuntes y entorpece cualquier seguimiento cronológico, lo que demuestra empíricamente que la falta de herramientas de registro formales imposibilita el análisis y la trazabilidad del comportamiento hídrico del cultivo a largo plazo.

Finalmente, al cruzar toda la información obtenida, se comprueba que la deficiente capacitación del personal y la escasez de herramientas convergen, provocando la baja eficiencia del riego. Los datos revelan que el personal operativo carece de guías exactas, lo que, sumado a la falta de información oportuna, resulta en el desperdicio continuo de recursos al aplicar el riego a destiempo. En síntesis, el análisis de campo demuestra la validez de todas las causas planteadas inicialmente, ratificando que la ineficiencia hídrica en el vivero Lastenia no es un hecho aislado, sino la consecuencia directa de operar bajo un esquema empírico, sin instrumentos de medición y con un registro de datos altamente vulnerable a errores humanos.

CAPÍTULO IV

4. MARCO PROPOSITIVO

4.1 Introducción

En el presente capítulo se detalla el diseño, desarrollo e implementación de la propuesta tecnológica central de esta investigación: un agente conversacional (chatbot) potenciado por modelos de aprendizaje automático supervisado, concebido para optimizar la eficiencia del riego en el cultivo de *Theobroma cacao* dentro del vivero Lastenia de la ULEAM, Extensión El Carmen. A lo largo de esta sección, se describe exhaustivamente la arquitectura de software adoptada, la cual funciona como un puente interactivo entre la telemetría capturada por una red de Internet de las Cosas (IoT) —encargada de monitorear la humedad y temperatura del suelo en tiempo real— y los usuarios finales, proporcionando un canal de comunicación intuitivo basado en el procesamiento de lenguaje natural (PLN).

Para edificar la solución tecnológica se optó por el marco de trabajo ágil Scrum. Organizar el trabajo del proyecto mediante ciclos de trabajo llamados sprints, ayudó a repartir mejor las actividades y garantizar la entrega continua de los módulos. Esta metodología ayudó a probar rápidamente las piezas claves del sistema, como la conexión con la base de datos, el servidor encargado de procesar las intenciones del usuario y el modelo de estrés hídrico, contar con esta flexibilidad fue indispensable para ir adaptando el sistema a lo que realmente exigía el trabajo de campo, sin arriesgar la estabilidad del código.

En esta parte también se detallaron los recursos humanos, técnicos y económicos que hacen viable la ejecución. Usando las historias de usuario y los requisitos del sistema como base, se armó el plan de programar la interfaz y conectar con el backend, esta arquitectura es uno solo, transformar la información del clima en indicaciones agronómicas precisas. Con esta innovación se deja atrás el uso exclusivo de la intuición o la práctica empírica, dotando al equipo del vivero de un instrumento de agricultura de precisión que destaca por su utilidad y fácil manejo.

4.2 Descripción de la propuesta

La propuesta del proyecto presente consiste en el diseño, desarrollo e implementación de un chatbot el cual estará integrado con un modelo de aprendizaje automático supervisado, enfocado en optimizar la gestión del riego para los cultivos del cacao en el Vivero Lastenia Vera. El sistema actúa como un puente de interacción entre una arquitectura de internet de las cosas (IoT), que obtendrá los datos de la humedad, temperatura del suelo en tiempo real y el usuario final.

El sistema está dirigido principalmente para el personal operativo, docentes y estudiantes de las carreras relacionadas a la agropecuaria, agronomía de la ULEAM Extensión El Carmen, quienes realizan labores en campo. Contará con dos perfiles de usuarios principales, administrador con el acceso total a la configuración de los sensores, modelos de predicción y gestión de usuarios, otro rol de usuario será el operador el cual está enfocado en la consulta ágil del estado del cultivo y la recepción de asistencia técnica para el riego.

Sus principales funcionalidades, el chatbot permitirá consultar las métricas ambientales mediante lenguaje natural, enviar alertas predictivas cuando el nivel de humedad caiga por debajo del umbral seguro para el cacao.

4.3 Determinación de recursos

4.3.1 Humanos

Tabla 4 Recursos humanos

Recurso	Cargo	Función
Rodríguez Ordoñez Kevin David	Desarrollador Principal	Diseñar la arquitectura IoT y de software. Entrenar el modelo de aprendizaje automático. Codificar el backend (Node.js) y el chatbot.
Ing. Rómulo Danilo Arévalo Hermida	Tutor / Supervisor	Revisar la viabilidad técnica del código. Validar el cumplimiento de los objetivos académicos e ingenieriles del proyecto.

Recurso	Cargo	Función
Encargado del Vivero Lastenia (o Estudiante Tesista de Agropecuaria)	Usuario Validador	Probar la interfaz conversacional en campo. Validar que las alertas y el riego correspondan a las necesidades biológicas del cacao.

4.3.2 Tecnológicos

Tabla 5 Recursos tecnológicos

Recurso	Característica	Descripción
Laptop MSI Prestige 14Evo	Procesador Intel Core i7 EVO 11va generación, 16GB RAM, SSD.	Utilizada como estación de trabajo principal para la codificación, entrenamiento del modelo ML y pruebas locales.
Node.js / Framework backend	Entorno de ejecución multiplataforma para JavaScript.	Utilizado para construir la API central que conectará el hardware IoT con la plataforma del chatbot.
Firebase / Base de Datos	Base de datos NoSQL en la nube en tiempo real.	Almacenamiento del historial de lecturas de los sensores, registro de usuarios y logs de interacción del chatbot.
Sensores capacitivos de humedad y DHT11	Módulos electrónicos de bajo voltaje (3.3V - 5V).	Captura física de la humedad del suelo y temperatura ambiente directamente en las fundas de cacao.
Microcontrolador (ej. ESP32)	Placa de desarrollo con módulo Wi-Fi integrado.	Nodo IoT encargado de leer los sensores y enviar la telemetría inalámbricamente al servidor.

4.3.3 Económicos

Tabla 6 Recursos económicos

Cantidad	Recurso	Precio unitario	Subtotal
1	Laptop MSI Prestige	\$1,100.00	\$1,100.00
1	Teléfono inteligente (Pruebas)	\$250.00	\$250.00
12 meses	Alojamiento en la nube / VPS	\$5.00	\$60.00
1	Kit de Hardware (ESP32 y Sensores ambientales)	\$85.00	\$85.00
400 horas	Horas de programación y pruebas	\$15.00	\$6,000.00
6 meses	Internet	\$25.00	\$150.00
TOTAL:			\$7,645.00

4.4 Desarrollo mediante metodología Scrum

Para la construcción de este sistema se seleccionó la metodología Scrum por que debido a la naturaleza dinámica e iterativa que exige la integración de la inteligencia artificial con componentes hardware (IoT). A diferencia de los métodos comunes como en cascada, Scrum se basa más en el empirismo y permite dividir el desarrollo en ciclos cortos denominados sprints.

Esta forma de trabajar es perfecta para el proyecto en el vivero Lastenia. En un lugar como este, donde se trabaja con la tierra y las plantas, las cosas pueden cambiar mucho cuando se hace una prueba. Por ejemplo, podemos descubrir que el sensor necesita ajustes especiales según el tipo de tierra donde está el cacao. Al ir entregando partes funcionales una a una, podemos comprobar que cada cosa funciona bien antes de seguir adelante. Primero, vemos si el sensor manda los datos correctamente. Luego, comprobamos si el modelo hace predicciones precisas. Finalmente, verificamos si el chatbot responde como debe. De esta manera, podemos hacer ajustes sin afectar el progreso general del proyecto. Esto nos permite avanzar de manera segura y sin problemas.

4.4.1 Descripción del Producto

4.4.1.1 Propósito del Producto

Desarrollar e implementar un agente conversacional inteligente que, mediante el uso de aprendizaje automático supervisado y telemetría IoT, automatice la toma de decisiones y proporcione asistencia técnica para garantizar la eficiencia del riego en las plántulas de cacao del vivero Lastenia.

4.4.1.2 Funcionalidades Clave

- Autenticación segura y gestión de roles de usuario.
- Lectura y almacenamiento en tiempo real de la humedad y temperatura del suelo.
- Procesamiento de lenguaje natural (NLP) para interpretar preguntas de los agricultores.
- Predicción de la necesidad de riego mediante el algoritmo de aprendizaje supervisado.
- Emisión de alertas proactivas a los operadores cuando se detecte riesgo de estrés hídrico.
- Generación de registros históricos de las condiciones climáticas del vivero.
- Panel de control (Dashboard) visual para administradores.
- Configuración y ajuste manual de los umbrales de humedad permitidos.
- Generación de reportes de consumo de agua e interacciones del chatbot.

4.4.1.3 Usuarios Objetivo

Tipo de usuario	Principales funcionalidades
Administrador	- Gestión total de usuarios e invitaciones. - Ajuste de parámetros del modelo de Machine Learning. - Acceso a reportes estadísticos e históricos completos.
Operador / Trabajador del Vivero	- Consultar el estado de las plantas mediante el chat. - Recibir notificaciones de alertas de riego.

4.4.1.4 Condiciones de Éxito del Producto

- El chatbot debe comprender y responder al menos al 85% de las intenciones (consultas) de los usuarios correctamente.
- El tiempo de respuesta entre la consulta del usuario y la entrega de la información o alerta en el chat no debe superar los 3 segundos.
- El sistema debe demostrar una optimización medible en el uso del agua en comparación con el riego manual tradicional.
- El hardware y la comunicación deben mantener un uptime (tiempo en línea) estable en las condiciones climáticas del vivero.
- La interfaz del chatbot debe ser calificada como intuitiva y fácil de usar por usuarios sin formación previa en sistemas informáticos.

4.4.2 Requerimientos del sistema

4.4.2.1 Historias de Usuario

4.4.2.1.1 Historia de Usuario 1: Consulta del estado del cultivo

Tabla 7 historia de usuario 1

HISTORIA DE USUARIO			
Id de historia	HU01	Título:	Consulta de humedad actual
Prioridad:	Alta	Usuario:	Operador del Vivero
Riesgo de desarrollo:	Bajo	Tamaño de la tarea:	Mediano
Como:	Operador del vivero Lastenia		
Quiero:	Preguntarle al chatbot cuál es el nivel de humedad actual de las plantas de cacao		
Para:	Saber si es necesario prepararme para realizar un ciclo de riego		
Criterios de aceptación:	El bot debe interpretar lenguaje natural (ej. "¿Cómo está la tierra?") y devolver el valor en porcentaje capturado por el sensor en tiempo real.		
Dependencias:	Conexión estable con la API del backend y base de datos		

4.4.2.1.2 Historia de Usuario 2: Alerta predictiva automática

Tabla 8 Historia de usuario 2

HISTORIA DE USUARIO			
Id de historia	HU02	Título:	Recepción de alertas de estrés hídrico
Prioridad:	Alta	Usuario:	Operador del Vivero
Riesgo de desarrollo:	Medio	Tamaño de la tarea:	Grande
Como:	Trabajador Agrícola		
Quiero:	Recibir un mensaje automático del chatbot avisándome que las plantas necesitan agua		
Para:	Poder ejecutar el riego manualmente a tiempo y evitar que el cacao sufra de marchitez por olvido.		
Criterios de aceptación:	Cuando el modelo de aprendizaje automático detecte que la humedad cae por debajo del umbral óptimo, el bot debe emitir una notificación proactiva al dispositivo del usuario		
Dependencias:	Modelo de aprendizaje automático entrenado y base de datos conectada.		

4.4.2.1.3 Historia de Usuario 3: Alerta predictiva automática

Tabla 9 Historia de usuario 3

HISTORIA DE USUARIO			
Id de historia	HU03	Título:	Procesamiento de consultas en lenguaje natural
Prioridad:	Alta	Usuario:	Operador del Vivero
Riesgo de desarrollo:	Medio	Tamaño de la tarea:	Mediano
Como:	Trabajador agrícola		
Quiero:	Hacer preguntas al chatbot usando mis propias palabras o modismos agrícolas (ej. "¿Toca regar las matas?")		
Para:	Recibir recomendaciones técnicas inmediatas sin necesidad de aprenderme comandos informáticos complejos o menús difíciles.		
Criterios de aceptación:	El motor NLP (Ollama/Llama3) debe interpretar la frase libre del usuario, clasificar la intención correctamente y devolver una respuesta coherente en un tiempo promedio inferior a 1.5 segundos.		
Dependencias:	Modelo de aprendizaje automático entrenado.		

4.4.2.1.4 Historia de Usuario 4: Configuración de umbrales

Tabla 10 Historia de usuario 4

HISTORIA DE USUARIO			
Id de historia	HU04	Título:	Ajuste de parámetros del modelo
Prioridad:	Media	Usuario:	Administrador
Riesgo de desarrollo:	Medio	Tamaño de la tarea:	Mediano
Como:	Administrador del sistema		
Quiero:	Modificar el porcentaje mínimo de humedad que activa las alertas		
Para:	Adaptar el comportamiento del chatbot a las diferentes etapas de crecimiento del cacao o temporadas de lluvia		
Criterios de aceptación:	Interfaz o comando específico protegido con autenticación de administrador que actualice las variables del algoritmo predictivo.		
Dependencias:	Roles de usuario (Autenticación).		

4.4.2.1.5 Historia de Usuario 5: Historial de datos

Tabla 11 Historia de usuario 5

HISTORIA DE USUARIO			
Id de historia	HU05	Título:	Consulta de historial de lecturas
Prioridad:	Baja	Usuario:	Administrador
Riesgo de desarrollo:	Bajo	Tamaño de la tarea:	Pequeño
Como:	Administrador / Tesista		
Quiero:	Consultar o descargar el listado con los riegos y niveles de humedad de jornadas previas		
Para:	Estudiar la conducta del cultivo y comprobar el consumo eficiente del agua		
Criterios de aceptación:	El sistema debe extraer los datos consolidados de Firebase y presentarlos en un formato legible.		
Dependencias:	Base de datos poblada con lecturas anteriores.		

4.4.2.2 Requerimientos funcionales

RF01: El sistema permitirá a los usuarios (Administrador y Operador) registrarse e iniciar sesión de manera segura.

RF02: El sistema permitirá procesar consultas en lenguaje natural a través del chatbot sobre el estado actual de humedad y temperatura del cultivo.

RF03: El sistema permitirá almacenar en la base de datos las lecturas enviadas en tiempo real por los sensores IoT.

RF04: El aplicativo se encargará de emitir y enviar alertas automáticas al personal operativo cada vez que se detecten situaciones de peligro relacionadas con el estrés hídrico en las plántulas.

RF05: A través de comandos textuales ingresados en la interfaz de mensajería, el sistema permitirá abrir o cerrar a distancia las electroválvulas encargadas de la distribución del agua.

RF06: El software otorgará al perfil de administrador los permisos necesarios para establecer y modificar los límites operativos tanto de humedad (valores inferiores al 75%) como de temperatura (rango de 28 a 30 °C).

RF07: El sistema permitirá generar reportes históricos sobre las variables climáticas recopiladas y el registro de activación de las bombas de agua.

4.4.2.3 Requerimientos no funcionales

RNF01: Tiempo de respuesta menor a 3 segundos en el chatbot

RNF02: Acceso al sistema mediante autenticación segura

RNF03: Disponibilidad continua del sistema y reconexión automática de los nodos IoT.

RNF04: Interfaz de usuario responsiva y adaptada a dispositivos móviles.

RNF05: Seguridad y cifrado de datos en la comunicación entre el hardware, el servidor y la aplicación.

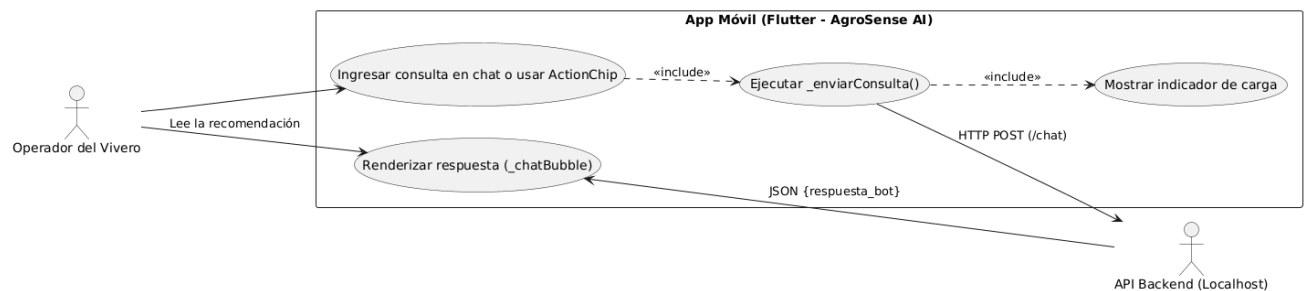
RNF06: Escalabilidad para soportar múltiples conexiones simultáneas en la base de datos.

4.4.3 Diseño del Sistema

4.4.3.1 Casos de Uso

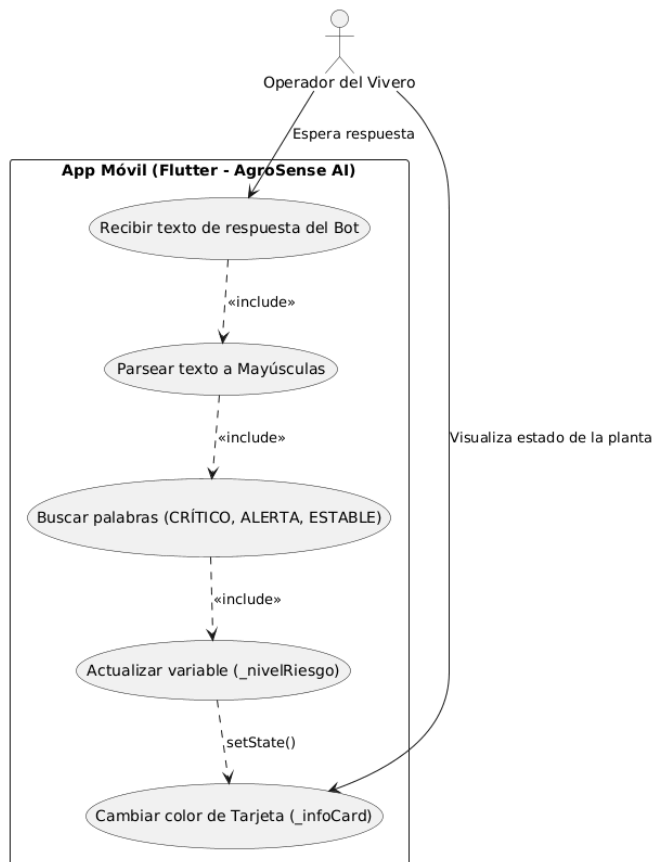
4.4.3.1.1 Caso de uso: captura y envío de telemetría

Ilustración 2 Diagrama de Caso de Uso: Consulta mediante lenguaje natural



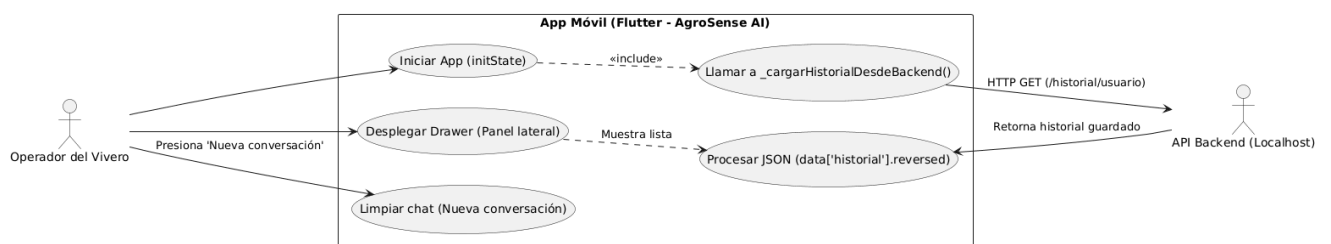
4.4.3.1.2 Caso de uso: Predicción y generación de alertas

Ilustración 3 Diagrama de Caso de Uso: Evaluación y actualización de riesgo ambiental



4.4.3.1.3 Caso de uso: Salida de información

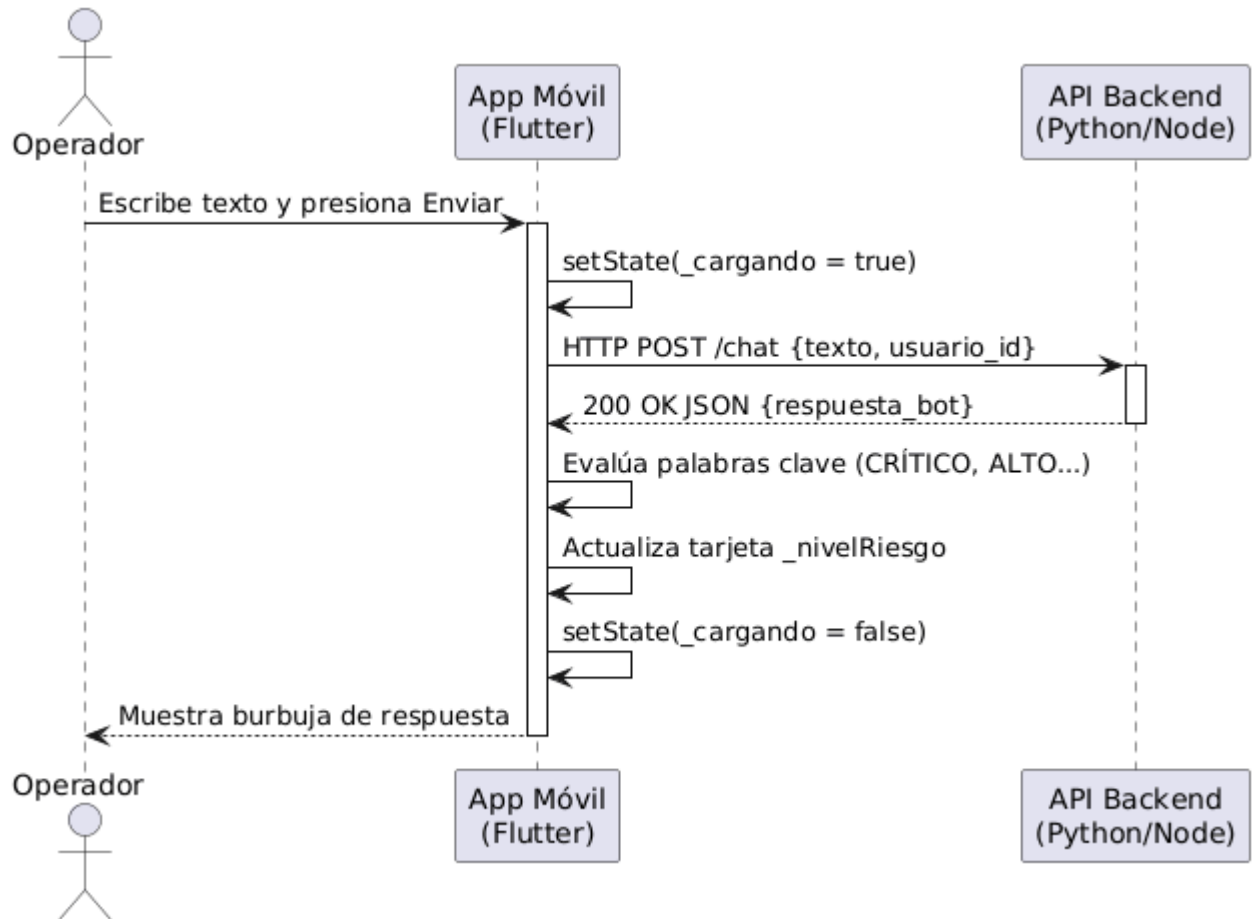
Ilustración 4 Diagrama de Caso de Uso: Consulta de historial de conversaciones



4.4.3.2 Diagramas de Secuencia

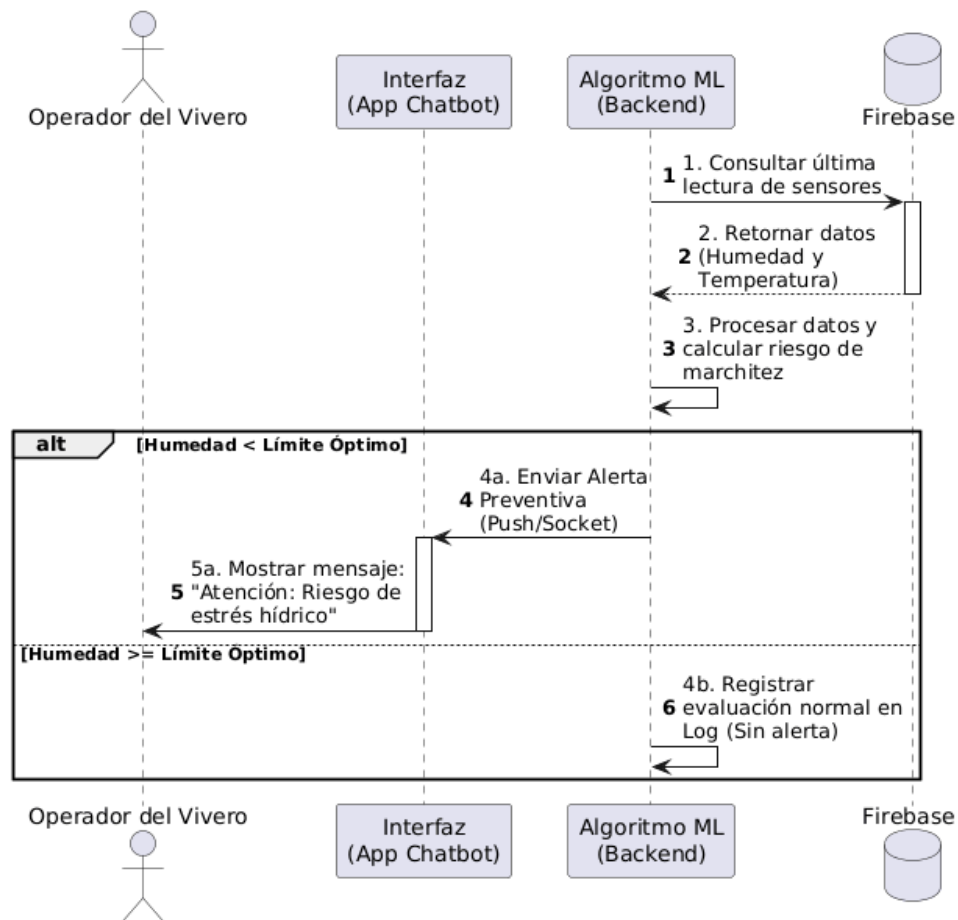
4.4.3.2.1 Diagrama de Secuencia: Captura y almacenamiento de telemetría

Ilustración 5 Diagrama de Secuencia: Envío de consulta y actualización de riesgo.



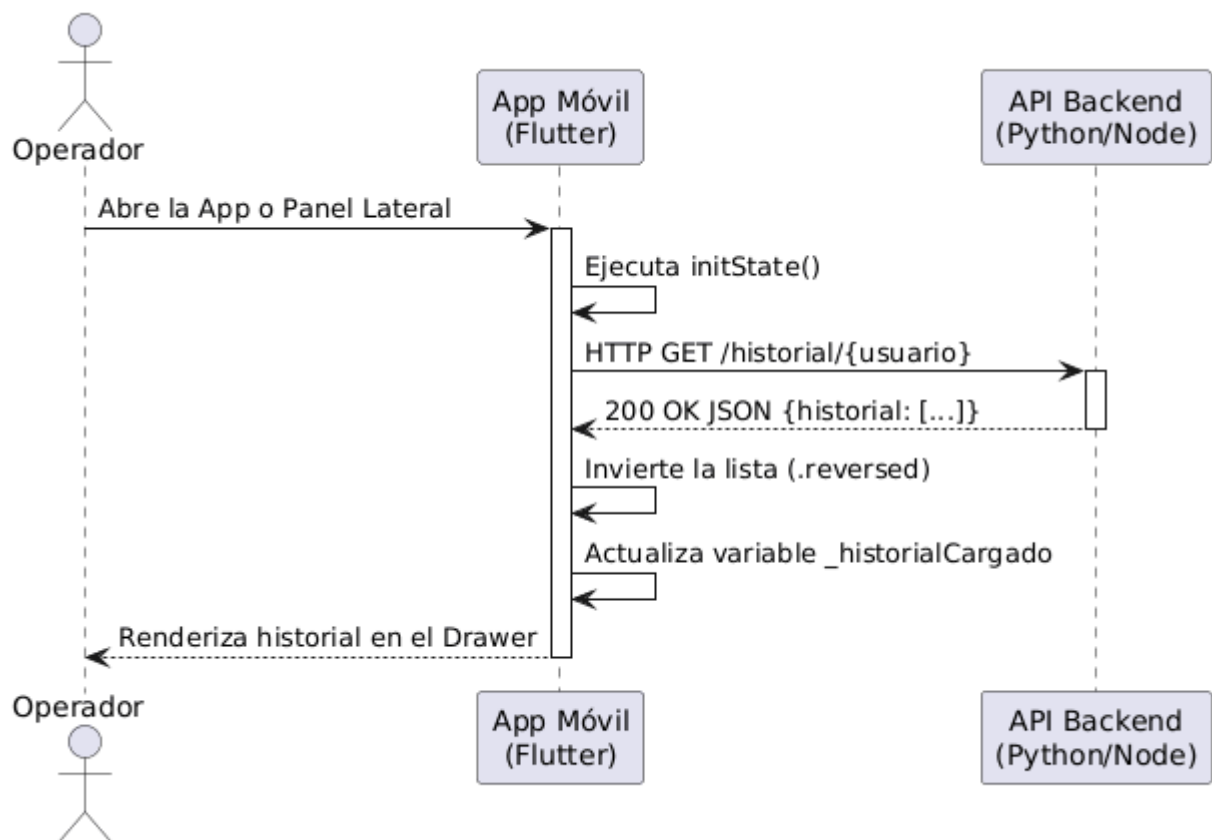
4.4.3.2 Diagrama de Secuencia: Predicción de estrés hídrico y evaluación de riesgos

Ilustración 6 Diagrama de Secuencia: Predicción de estrés hídrico y evaluación de riesgos



4.4.3.2.3 Diagrama de Secuencia: Generación y descarga de reportes

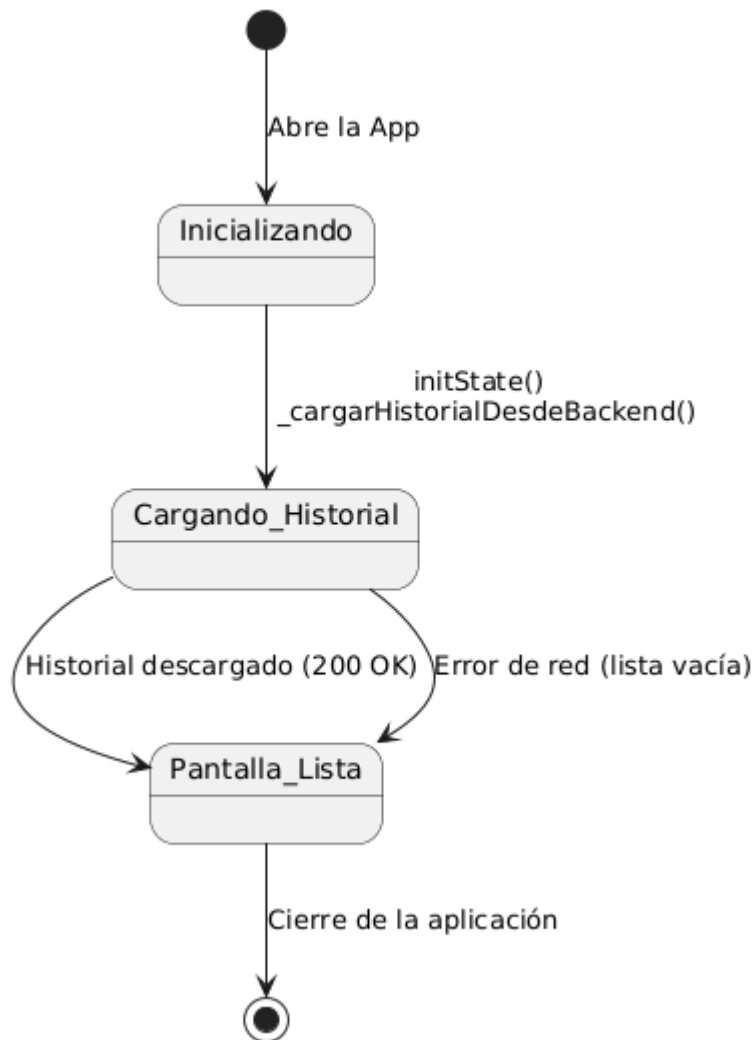
Ilustración 7 Diagrama de Secuencia: Carga del historial en el panel lateral



4.4.3.3 Diagramas de Estado

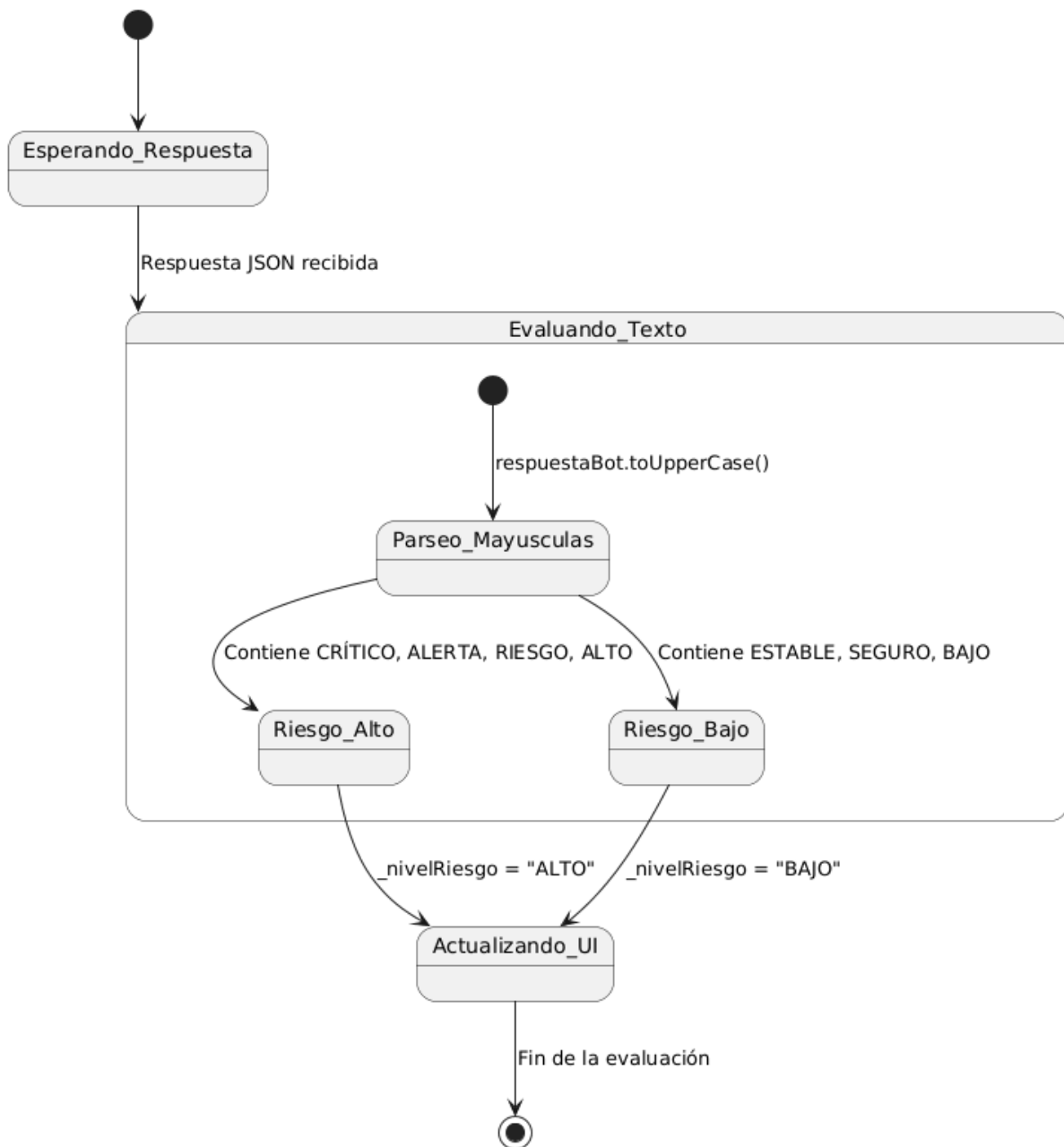
4.4.3.3.1 Diagrama de Estado: Ciclo de vida de la pantalla principal (Dashboard).

Ilustración 8 Diagrama de Estado: Ciclo de vida de la pantalla principal (Dashboard).



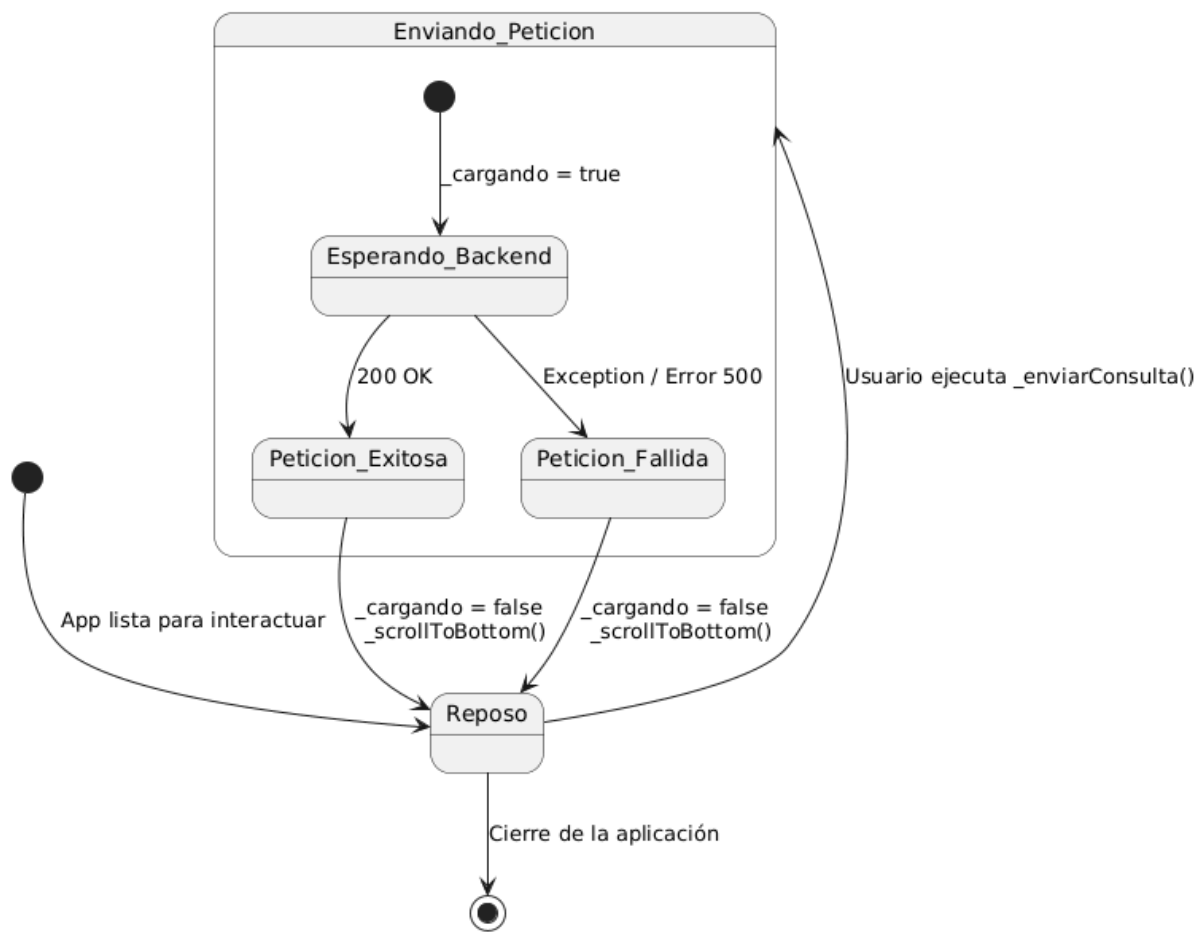
4.4.3.3.2 Diagrama de Estado: Evaluación de riesgo ambiental

Ilustración 9 Diagrama de Estado Evaluación de riesgo ambiental.



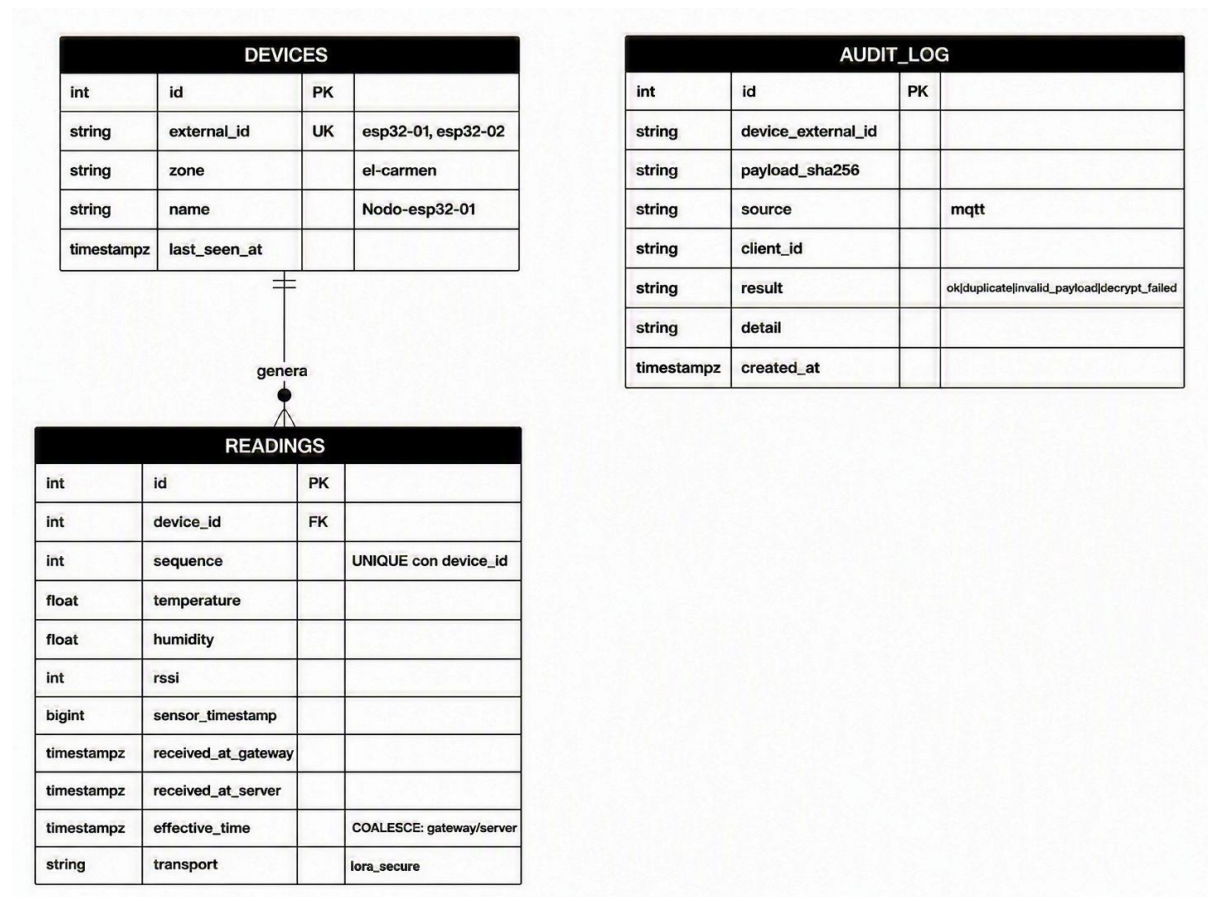
4.4.3.3 Diagrama de Estado: Petición HTTP y envío de mensajes

Ilustración 10 Diagrama de Estado Petición HTTP y envío de mensajes.



4.4.3.4 Diagrama de Base de Datos

Ilustración 11 Diagrama de Base de Datos



4.4.4 Arquitectura del Sistema

4.4.4.1 Arquitectura general

El sistema se fundamenta en una **arquitectura en capas (n-tier)**, un modelo que segmenta las responsabilidades del software para mejorar su mantenibilidad y escalabilidad. Esta estructura jerárquica asegura la correcta separación entre la captura de datos (capa física), la lógica de negocio (backend) y la visualización de la información (cliente).

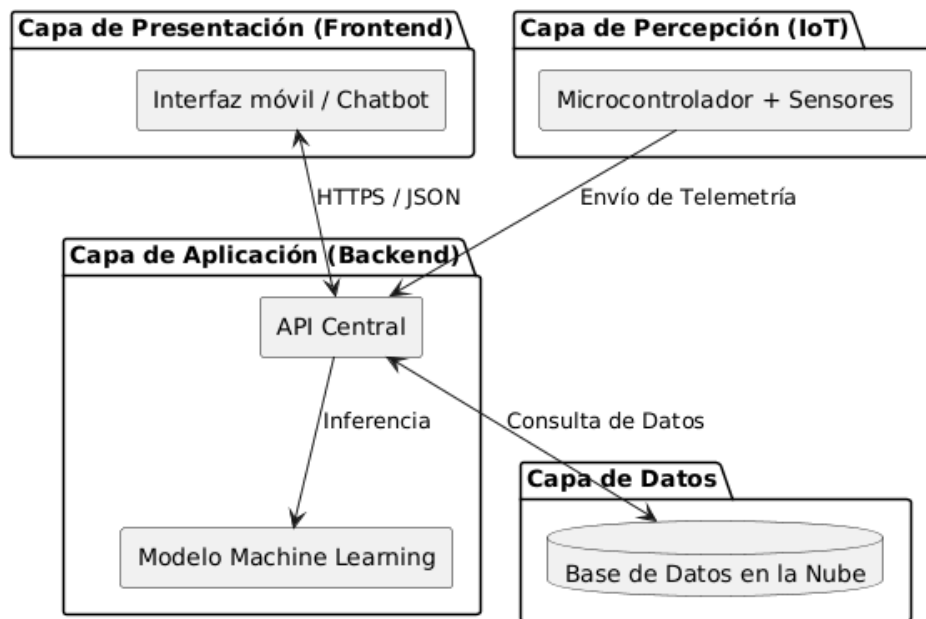
La estructura del sistema se divide en tres partes principales:

- **Capa de Percepción (IoT):** En esta parte se enfoca más del hardware que se instala directamente en el campo, como los sensores y microcontroladores. Su objetivo es

medir constantemente las condiciones del entorno como la humedad y la temperatura, para enviar esos datos al servidor central.

- **Capa de Aplicación (Backend):** Funciona como cerebro o núcleo de procesamiento. En este espacio opera el servidor encargado de administrar la API, resguardar la información en la nube y correr el modelo de aprendizaje automático que dispara las alertas preventivas.
- **Capa de Presentación (Frontend):** Representa el entorno visual diseñado para el usuario. Funcionando bajo una dinámica Cliente-Servidor, la aplicación móvil opera como el cliente que se conecta al *backend* para consumir los datos, permitiendo a la persona realizar consultas y observar las advertencias predictivas directamente en su pantalla.

Ilustración 12 Arquitectura general del sistema



Esta organización permite que el sistema opere de forma desacoplada; los cambios en la interfaz no afectan la lógica de procesamiento en el servidor, garantizando un diseño modular y profesional.

4.4.4.2 Tecnologías utilizadas

Tabla 12 Tecnologías utilizadas en el proyecto

Categoría	Herramienta / Tecnología	Uso en el Proyecto
Lenguaje de Backend	JavaScript (Node.js)	Desarrollo del servidor, lógica de control del chatbot y gestión de la API del sistema.
Inteligencia Artificial	Python	Desarrollo de scripts para el procesamiento de lenguaje natural y el modelo predictivo de IA.
Base de Datos	Firebase (Firestore)	Gestión de datos en tiempo real, almacenamiento de telemetría y logs de interacción.
Plataforma de Chat	Interfaz Web Integrada	Chatbot nativo desarrollado dentro de la plataforma general de monitoreo ambiental.
API / Comunicación	REST / JSON	Protocolos y formato de intercambio de información entre el frontend y el servidor.
Entorno de Desarrollo	Visual Studio Code	Editor principal de código para el desarrollo del Chatbot y scripts de IA.
Control de Versiones	Git / GitHub	Gestión del código fuente, versionamiento y colaboración del desarrollo.

4.4.4.3 Roles Scrum

Para el desarrollo del módulo del Asistente Inteligente (Chatbot), se ha adaptado el marco de trabajo ágil Scrum al contexto de un proyecto de titulación. La asignación de roles garantiza que se cumplan tanto los requerimientos técnicos del software como los objetivos agronómicos del Vivero Lastenia:

1. Product Owner (Dueño del Producto)

Representa la voz de los interesados y los estándares académicos. Su objetivo es maximizar el valor del producto y validar las entregas al final de cada Sprint.

- Tutor del proyecto (Ing. Rómulo Danilo Arévalo) y Coordinador del Vivero (Mg. Pedro Nivelá).

Responsabilidad: Aprobar los flujos conversacionales, validar que los umbrales de humedad para las alertas sean los correctos para las plántulas de cacao y asegurar que el chatbot cumpla con los objetivos de la investigación.

2. Scrum Master

Es el facilitador del proyecto, encargado de asegurar que se respeten los tiempos y las prácticas ágiles.

- **Asignado a:** Kevin David Rodríguez Ordoñez.

Responsabilidad: Planificar los Sprints del desarrollo del chatbot, gestionar la pila del producto (Product Backlog) y resolver cualquier impedimento técnico u obstáculo que surja durante la integración del software.

3. Development Team (Equipo de Desarrollo)

El profesional encargado de construir y entregar un incremento de producto funcional y libre de errores.

- **Asignado a:** Kevin David Rodriguez Ordoñez.

Responsabilidad: Codificar la lógica del servidor (Node.js), programar los scripts de Inteligencia Artificial (Python), conectar el sistema a la base de datos (Firebase) e integrar la interfaz gráfica del chat dentro de la plataforma de monitoreo.

4.4.5 Definición de Hecho (DoD)

4.4.5.1 Criterios Generales

Los criterios generales de la Definición de Hecho (DoD) constituyen el conjunto de requisitos transversales de calidad técnica que deben debe superar una serie de requisitos obligatorios antes de aprobarse para su integración:

- **El código fuente compila y opera sin fallos:** Tanto la lógica del servidor en Node.js como los scripts de procesamiento de lenguaje natural en Python deben quedar totalmente exentos de errores sintácticos, problemas de ejecución (*runtime errors*) o alertas críticas que pongan en riesgo la estabilidad del sistema.
- **La funcionalidad desarrollada debe cumplir los requisitos establecidos en la historia de usuario correspondiente:** El incremento de software debe satisfacer al 100% los criterios de aceptación y las reglas de negocio descritas previamente en el Product Backlog.
- **Interoperabilidad modular:** El asistente conversacional debe conectarse sin problema a la base de datos como panel principal de monitoreo. Esta integración tiene que realizarse con cuidado para no generar cuellos de botella en el rendimiento y dañar funcionalidades que ya funcionaban sin problema en el sistema.
- **Coherencia gráfica y estabilidad visual:** La interfaz del chat debe mantener un estilo visual que coincida con el diseño general del sistema. Con esto se busca evitar fallos estéticos, tales como textos que rebasan sus márgenes, símbolos extraños o errores durante el renderizado.

4.4.5.2 Criterios Específicos del Proyecto

Los criterios específicos de la Definición de Hecho (DoD) determinan los estándares de aceptación y las reglas de negocio particulares que aplican de forma exclusiva a las funciones

de procesamiento de lenguaje natural, evaluación predictiva y visualización del chatbot. Para asimilar que una función de este módulo está terminada, debe cumplir con las siguientes validaciones:

- El procesamiento de lenguaje natural (NLP) debe identificar y clasificar la intención del usuario como preguntar por la humedad o la temperatura antes de realizar la consulta en la base de datos.
- Las alertas del chat solo deben mostrarse cuando el modelo predictivo detecta que las mediciones reales cayeron por debajo de los estándares saludables del cultivo.
- La capacidad de modificar o ajustar los parámetros directamente en la pantalla de chat, será una función de forma exclusiva para los usuarios que ingresen con permisos de Administrador o Coordinador.
- En cambio, a los estudiantes u operadores solo se permite que puedan revisar las condiciones climáticas del momento y los reportes pasados. Así se bloquea cualquier intento de alterar las variables del motor predictivo.
- El historial de interacciones debe registrar en la base de datos (Firebase) todas las consultas realizadas por los usuarios y las alertas preventivas emitidas por el chatbot para garantizar la trazabilidad del sistema.

4.4.6 Product Backlog

El Product Backlog es el inventario priorizado de todas las funcionalidades y tareas técnicas que deben desarrollarse para construir el asistente inteligente (Chatbot). Esta lista inicial desglosa las Historias de Usuario en unidades de trabajo manejables para los Sprints. Dado que este es el documento de planificación inicial, todas las actividades se encuentran en estado "Por hacer", las cuales irán cambiando de estado conforme avance la ejecución del proyecto.

Tabla 13 Product Backlog del sistema

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Diseñar y maquetar la interfaz principal de la aplicación móvil (DashboardScreen) en Flutter.	HU01	Alta	Por hacer

Nro.	Tarea	Historia de usuario	Prioridad	Estado
2	Desarrollar el interfaz del chat (burbujas de mensaje de usuario y bot, y scroll automático).	HU01	Alta	Por hacer
3	Configurar el servidor backend local y la conexión con la base de datos (SQLite).	HU02	Alta	Por hacer
4	Realizar la conexión mediante peticiones HTTP (POST y GET) entre Flutter y el backend.	HU02	Alta	Por hacer
5	Entrenar el modelo de Aprendizaje Automático en el servidor para reconocer peticiones agrícolas.	HU03	Alta	Por hacer
6	Integrar los botones de acción rápida en la interfaz para consultas predefinidas.	HU03	Media	Por hacer
7	Codificar la lógica condicional en Flutter para cambiar visualmente el nivel de riesgo ambiental (ALTO/BAJO) según la respuesta del bot.	HU04	Alta	Por hacer
8	Programar la extracción del historial de consultas desde el backend hacia el panel lateral.	HU05	Media	Por hacer

4.4.7 Sprint 1: Configuración del Backend y Motor Conversacional Básico

4.4.7.1 Planificación del Sprint

Tabla 14 Planificación del sprint 1

SPRINT 1: Arquitectura Base y UI del Chatbot			
Duración:	2 semanas (06/01/2026 – 12/01/2026)	Prioridad:	Alta
Objetivo:	Desarrollar la interfaz gráfica de usuario (GUI) utilizando el framework Flutter, implementando la pantalla principal del Dashboard, el diseño de las burbujas de chat y los botones de acción rápida para garantizar una experiencia de usuario (UX) intuitiva en dispositivos móviles.		
Historias de usuario:	HU01: Consulta del estado del cultivo.		

SPRINT 1: Arquitectura Base y UI del Chatbot	
Tareas de desarrollo:	• Configuración del entorno de desarrollo de Flutter para iniciar la aplicación. Desarrollo de un script en Python para reconocer intenciones como la humedad y la temperatura. • Diseño de la pantalla principal (DashboardScreen) con tarjetas informativas. • Programación del componente de diseño para las burbujas de chat (_chatBubble). • Implementación de la lista horizontal de sugerencias (ActionChip).
Plan de entrega:	• Configuración de servidor y Firebase: • Inicializar el proyecto backend y establecer las credenciales de Firebase. (Tiempo: 6 horas). Fecha: 06/01/2026 - 07/01/2026. • Script de NLP en Python: • Desarrollar la lógica de análisis de texto para clasificar las preguntas del usuario. (Tiempo: 12 horas). Fecha: 08/01/2026 - 10/01/2026. • Diseño de Interfaz Web: • Maquetar la ventana de chat y programar las peticiones HTTP (fetch/AJAX) hacia el servidor. (Tiempo: 10 horas). Fecha: 12/01/2026 - 14/01/2026. • Pruebas de integración: • Enlazar los módulos y corregir latencias de respuesta. (Tiempo: 12 horas). Fecha: 15/01/2026 - 19/01/2026.

4.4.7.2 Sprint Backlog

Historia de usuario.	Tarea	Duración estimada
HU01	Configurar el servidor central (Node.js) y establecer la conexión asíncrona con la base de datos (Firebase).	6 hora
HU01	Desarrollar el script de Inteligencia Artificial (Python) para el Procesamiento de Lenguaje Natural (NLP) que identifique intenciones de consulta.	12 horas

Historia de usuario.	Tarea	Duración estimada
HU01	Diseñar y acoplar la interfaz gráfica conversacional del Chatbot dentro de la plataforma web de monitoreo.	10 horas

4.4.7.3 Desarrollo del Sprint

4.4.7.3.1 Funcionalidad de Consulta Climática Mediante Lenguaje Natural

En esta funcionalidad se desarrolló el frontend principal del asistente inteligente integrado en la aplicación móvil. Esta característica permite a los usuarios (operadores o estudiantes) interactuar con el sistema a través de un panel conversacional fluido. El flujo desarrollado estructura la interfaz mediante el framework Flutter, definiendo la pantalla principal (DashboardScreen), las tarjetas de estado ambiental y el campo de entrada de texto.

- **Conexión Backend y Datos**

A continuación, se presenta el componente programado en Dart. Este fragmento de código es el encargado de renderizar gráficamente los mensajes del chat, evaluando mediante un parámetro booleano (esUser) si el mensaje pertenece al operador o al bot, alineando el texto y cambiando los colores (verde o blanco) para lograr una clara diferenciación visual:

Ilustración 13 Código fuente: Componente de diseño para las burbujas de chat en Dart

```
Widget _chatBubble(String texto, bool esUser) {  
  return Align(  
    alignment: esUser ? Alignment.centerRight : Alignment.centerLeft,  
    child: Container(  
      margin: const EdgeInsets.symmetric(vertical: 5, horizontal: 10),  
      padding: const EdgeInsets.all(14),  
      constraints: BoxConstraints(  
        maxWidth: MediaQuery.of(context).size.width * 0.75,  
      ),  
      decoration: BoxDecoration(  
        color: esUser ? Colors.green[600] : Colors.white,  
        borderRadius: BorderRadius.only(  
          topLeft: const Radius.circular(20),  
          topRight: const Radius.circular(20),  
          bottomLeft: Radius.circular(esUser ? 20 : 0),  
          bottomRight: Radius.circular(esUser ? 0 : 20),  
        ),  
        boxShadow: const [  
          BoxShadow(  
            color: Colors.black12,  
            blurRadius: 4,  
            offset: Offset(0, 2),  
          ),  
        ],  
      ),  
      child: Text(  
        texto,  
        style: TextStyle(color: esUser ? Colors.white : Colors.black87),  
      ),  
    ),  
  );  
}
```

Adicionalmente, se programó un componente de desplazamiento horizontal que contiene botones de sugerencias predefinidas (ActionChip). Esto optimiza la interacción del operador en el campo, permitiéndole enviar consultas rápidas sin necesidad de usar el teclado virtual.

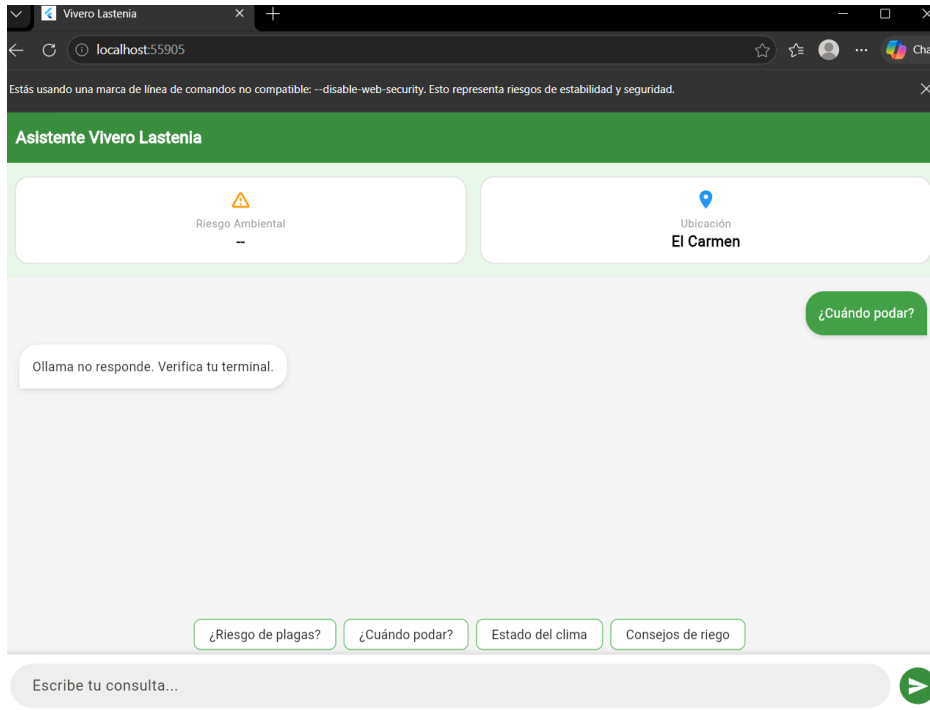
4.4.7.4 Sprint Review

4.4.7.4.1 Descripción de la funcionalidad

Se evaluó la integración y comunicación asíncrona entre la interfaz de usuario (desarrollada en el SDK de Flutter) y la API REST del backend (FastAPI/Python). El objetivo fue validar que las peticiones HTTP del cliente alcanzaran el enrutador lógico del servidor y que el frontend pudiera gestionar adecuadamente tanto las respuestas del modelo predictivo como las

excepciones generadas por la falta de disponibilidad del motor de Inteligencia Artificial local (Ollama).

Ilustración 14 Alerta ante indisponibilidad del motor LLM local



En la evidencia se ve que los microservicios se comunican entre sí de manera activa. Después de enviar la consulta, el servidor de backend recibió la petición, procesó el problema con el motor local y envió un mensaje de error claro. El frontend entendió este mensaje y mostró la alerta de forma segura en la interfaz gráfica sin detener el funcionamiento de la aplicación..

- **Retroalimentación recibida**

Falta de seguimiento en la lectura: Se detectó que al agregar múltiples mensajes consecutivos en la interfaz, la pantalla no se desplazaba automáticamente hacia abajo, obligando al usuario a hacer scroll manual para leer la respuesta más reciente del bot.

- **Mejoras realizadas**

Para resolver la navegación, se implementó un controlador de vista (`ScrollController`) y el método asíncrono `_scrollToBottom()`. Esta mejora fuerza a la interfaz a animar el

desplazamiento hacia el límite inferior de la lista cada vez que se detecta un nuevo elemento en el arreglo de mensajes.

4.4.7.5 Pruebas del Sprint

4.4.7.5.1 Pruebas de Caja Negra (Validación de Usabilidad)

Tabla 15 Pruebas de Caja Negra Sprint 1

Nombre de la Prueba	Condición de Entrada / Ejecución	Resultado Esperado	Observación
Límites del componente _chatBubble	Ingresa una cadena de texto extremadamente larga en el campo de texto.	El widget debe limitar su ancho máximo al 75% de la pantalla y forzar un salto de línea automático.	Se valida correctamente el uso de la propiedad BoxConstraints. No hay desbordamiento en pantalla.
Interacción de Sugerencias (ActionChip)	Presionar el botón de sugerencia rápida "¿Es necesario regar?".	El texto debe agregarse inmediatamente a la lista de mensajes sin necesidad de presionar el botón de envío principal.	Prueba exitosa. Se optimiza el tiempo de consulta en el entorno del vivero.
Bloqueo del campo de texto	Intentar escribir o enviar un mensaje mientras el sistema procesa una respuesta anterior (_cargando = true).	El TextField y el botón de enviar deben bloquearse para evitar envíos múltiples accidentales.	Funciona correctamente. La propiedad enabled bloquea eficazmente el widget de entrada.

4.4.7.5.2 Pruebas de Caja Blanca (Lógica de Estados UI)

Tabla 16 Pruebas de Caja Blanca Sprint 1

Método	Acción esperada	Acción obtenida	Observación
Condicional de color y alineación	Evaluar la variable esUser = true.	Asignar el color Colors.green[600] y alinear a la derecha (Alignment.centerRight).	La asignación de memoria visual y diseño condicional funciona sin generar excepciones en Flutter.

4.4.7.6 Incremento y Entregables del Sprint

Al finalizar este primer Sprint enfocado en la estructuración de la aplicación, se obtuvieron los siguientes entregables funcionales:

- Diseño base del aplicativo móvil (`DashboardScreen`) completamente estructurado con las tarjetas de estado ambiental.
- Componente `_chatBubble` funcional y responsivo para diferenciar visualmente los diálogos.
- Implementación de sugerencias de texto rápido mediante el widget `ActionChip`.

4.4.7.7 Estado actual del Product Backlog

Tabla 17 Estado actual del Product Backlog - Sprint 1

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Configuración e inicialización del microcontrolador ESP32 y entorno PlatformIO.	HU01	Alta	Completada
2	Lectura automática de la temperatura ambiental mediante el sensor DHT22.	HU01	Alta	Completada
3	Lectura automática de la humedad relativa mediante el sensor DHT22.	HU01	Alta	Completada

Nro.	Tarea	Historia de usuario	Prioridad	Estado
4	Implementación de la estabilización térmica previa de 2000 ms y pausas de 500 ms entre muestras.	HU01	Media	Completada
5	Validación inicial de los datos obtenidos para descartar valores nulos o corruptos (NaN).	HU01	Alta	Completada
6	Configuración del intervalo de medición continua cada 15 minutos de forma autónoma.	HU01	Media	Completada
7	Persistencia del número de secuencia en la memoria flash NVS del ESP32 mediante la librería <i>Preferences</i> .	HU01	Alta	Completada
8	Procesamiento y filtrado de datos mediante la técnica de media recortada sobre múltiples muestras.	HU02	Alta	Por hacer
9	Implementación del buffer circular de 10 posiciones para el almacenamiento temporal de mediciones.	HU04	Alta	Por hacer
10	Desarrollo del algoritmo de detección de anomalías por variaciones bruscas de temperatura o humedad.	HU03	Alta	Por hacer
11	Construcción y serialización del payload en formato JSON para el intercambio con el módulo de transmisión.	HU05	Alta	Por hacer

4.4.8 Sprint 2: Entrenamiento del Motor NLP y Desarrollo del Backend

4.4.8.1 Planificación del Sprint

Tabla 18 Planificación del Sprint 2

SPRINT 2: Backend y Motor de Procesamiento de Lenguaje Natural (NLP)			
Duración:	2 semanas (13/01/2026 – 27/01/2026)	Prioridad:	Alta
Objetivo:	Desarrollar la API REST en el servidor backend (local) y entrenar el modelo de Aprendizaje Automático Supervisado para el Procesamiento de Lenguaje Natural (NLP), permitiendo al sistema clasificar las intenciones de los agricultores a partir de texto libre		
Historias de usuario:	HU02: Procesar consultas en lenguaje natural a través del chatbot.		
Tareas de desarrollo:	• Puesta en marcha del servidor local y habilitación de las rutas de enlace (/chat y /historial). • Diseño y recopilación del corpus (dataset) alimentado con jerga agrícola propia de la zona. • Entrenamiento del clasificador supervisado para mapeo de intenciones. • Conexión del motor inteligente con la base de datos (SQLite).		
Plan de entrega:	• Validación de Rangos y Control de Errores (isnan): • Programación de bloques condicionales enfocados en aislar registros dañados o anormales dentro de las mediciones del vivero. • Tiempo de desarrollo: 12 horas • Fecha de Inicio y Fecha Final: 16/01/2026 - 19/01/2026. • Algoritmo de Filtrado (Media Recortada): • Programar el muestreo múltiple (5 muestras con intervalos de 500 ms) eliminando extremos máximos y mínimos. • Tiempo de desarrollo: 14 horas • Fecha de Inicio y Fecha Final: 16/01/2026 - 19/01/2026. • Integración del Buffer Temporal: • Conectar el módulo de procesamiento con la memoria temporal para mantener la trazabilidad de los ciclos de lectura. • Tiempo de desarrollo: 12 horas • Fecha de Inicio y Fecha Final: 20/01/2026 - 23/01/2026. • Pruebas de Estabilidad y Depuración: • Validar el comportamiento del firmware ante fluctuaciones y simulación de fallos de hardware en el puerto serial. • Tiempo de desarrollo: 10 horas. • Fecha de Inicio y Fecha Final: 24/01/2026 - 27/01/2026.		

4.4.8.2 Sprint Backlog

Tabla 19 Sprint Backlog 2

Historia de usuario.	Tarea	Duración estimada
HU02	Configurar el servidor backend local y la conexión con la base de datos (SQLite).	10 horas
HU02	Entrenar el modelo de Aprendizaje Automático (NLP) en el servidor para reconocer intenciones agrícolas.	16 horas

4.4.8.3 Desarrollo del Sprint

4.4.8.3.1 Funcionalidad de Validación de Rangos Físicos y Control de Errores

El núcleo de la inteligencia del chatbot radica en su capacidad para comprender las consultas del usuario sin obligarlo a utilizar comandos rígidos. Para lograr esto, se implementó un motor de Procesamiento de Lenguaje Natural (NLP) en el backend.

Generación del corpus y entrenamiento supervisado: Se obtuvo un dataset reuniendo sinónimos, modismos y expresiones coloquiales de los agricultores del Carmen cuando hablan de cacao. Así se recopiló cada grupo de frases con diferentes intenciones como, “¿Toca regar las matas?”, “¿Falta agua?” o “¿Las hojas están secas?”, así se pudo categorizar las consultas que se puedan realizar refiriéndose al riego.

Una vez entrenado, el modelo es capaz de recibir una frase nueva no vista anteriormente, calcular su probabilidad y clasificarla en la intención correcta, desencadenando la lógica de consulta ambiental en la base de datos.

- **Desarrollo de la API REST:**

Para comunicar el motor inteligente con la interfaz móvil programada en Flutter, se configuró un servidor que expone la ruta HTTP POST /chat. Este *endpoint* recibe un *payload* en formato JSON estructurado de la siguiente manera, de acuerdo con el código implementado en el sistema:

Ilustración 15 Código fuente: Estructura JSON del payload para la petición HTTP

```
{  
  "texto": "quiero regar las plantas, están secas o les falta agua?",  
  "usuario_id": "usuario_demo",  
  "rol": "tecnico"  
}
```

El servidor extrae el valor del campo texto, lo procesa a través del modelo NLP para clasificar la intención, y finalmente retorna la recomendación agronómica junto con los datos del clima en la variable respuesta_bot, la cual es enviada de vuelta a la aplicación móvil.

4.4.8.3.2 Funcionalidad de Muestreo Múltiple y Filtrado por Media Recortada

Para integrar la interfaz de usuario móvil con el servidor backend, se implementó el módulo de comunicación basado en peticiones asíncronas utilizando la librería http de Dart. Este componente permite enviar las consultas textuales del operador y recibir la respuesta procesada por el motor de inteligencia artificial en formato JSON.

- **Desarrollo del Código Fuente (Método de Envío):**

El siguiente bloque de código demuestra cómo la aplicación empaqueta la consulta y se comunica con el servidor local a través del método `_enviarConsulta`:

Ilustración 16 Código fuente: Método asíncrono `_enviarConsulta`

```
Future<void> _enviarConsulta(String texto) async {  
  if (texto.trim().isEmpty || _cargando) return;  
  
  setState(() {  
    _isCancelled = false;  
    _mensajes.add({"rol": "usuario", "texto": texto});  
    _cargando = true;  
  });  
  
  _controller.clear();  
  _scrollToBottom();  
  
  try {  
    final response = await http.post(  
      Uri.parse('http://10.0.2.2:8000/chat'),  
      headers: {"Content-Type": "application/json"},  
      body: jsonEncode({  
        "texto": texto,  
        "usuario_id": usuarioFijo,  
        "rol": "tecnico",  
      }),  
    );  
  
    if (_isCancelled) return;  
  
    if (response.statusCode == 200) {  
      final data = jsonDecode(response.body);  
      String respuestaBot = data["respuesta_bot"] ?? "Sin respuesta del servidor."  
  
      setState(() {  
        _mensajes.add({"rol": "bot", "texto": respuestaBot});  
      });  
    }  
  }  
}
```

La función utiliza `http.post` apuntando a la dirección local del emulador (10.0.2.2:8000/chat), enviando la estructura JSON con el texto y el rol del usuario. Al recibir un código de estado 200 OK, la aplicación analiza la respuesta del bot en busca de palabras clave ("ALTO" o "BAJO") para actualizar de forma dinámica las tarjetas de riesgo ambiental en el panel superior de la interfaz.

4.4.8.4 Sprint Review

Durante la revisión del Sprint 2, se evaluó la integración y la comunicación asíncrona entre la interfaz móvil (desarrollada en Flutter) y la API REST del backend local. La meta principal consistió en verificar que los llamados HTTP del cliente llegaran sin tropiezos al servidor,

garantizando que el frontend manejara con soltura tanto las contestaciones del modelo de Lenguaje Natural (NLP) como los lapsos de espera del usuario.

4.4.8.4.1 Retroalimentación recibida:

- **Concurrencia de envíos:** Al demorar unos segundos el servidor en descifrar la intención del NLP, surgía el riesgo de que el usuario pulsara varias veces el botón de envío, saturando el *backend* con peticiones HTTP duplicadas e innecesarias.
- **Manejo de excepciones de red:** Si el servidor local (enlazado al puerto 8000) sufría caídas intermitentes, el sistema exigía un aviso visual más directo que alertara al operador sobre el fallo de conectividad, impidiendo que la interfaz se quedara congelada a la espera de una respuesta.

4.4.8.4.2 Mejoras Realizadas:

- **Control de estados y cancelación:** Se incorporó un sistema de bloqueo dentro de la función `_enviarConsulta`. Con esto se logra deshabilitar el campo de texto (`TextField`) mientras la variable `_cargando = true`, además de brindarle al usuario la opción de abortar la solicitud activa mediante un botón de interrupción (`_isCancelled = true`).
- **Manejo robusto de errores (Try-Catch):** Se estructuraron bloques de control de excepciones en el cliente para interceptar caídas de red o fallos en el servidor (Status 500), asegurando que el chat agregue una burbuja con una advertencia limpia ("*Error de conexión: El backend no responde*") en lugar de comprometer la estabilidad del aplicativo móvil.

4.4.8.5 Pruebas del Sprint

4.4.8.5.1 Pruebas de Caja Negra (Validación de Filtrado y Rangos Físicos)

Tabla 20 Pruebas de Caja Negra Sprint 2

Nombre de la Prueba	Condición de Entrada / Ejecución	Resultado Esperado	Observación
Petición HTTP POST (/chat)	Desconectar el servicio del backend simulando una caída de red durante el método <code>_enviarConsulta</code> .	El bloque catch (e) intercepta la excepción e inyecta una burbuja de error descriptiva al arreglo de mensajes sin crashear la app.	Ejecución correcta. Se garantiza la estabilidad del cliente ante fallos de conectividad en el vivero.
Evaluación de Riesgo Dinámico	Activar la bandera <code>_isCancelled = true</code> al presionar el botón de parada mientras el servidor está procesando.	El flujo asíncrono descarta el resultado posterior y evita actualizar el estado de la UI con datos desfasados.	Correcto. Se evita la propagación de estados erróneos en la pantalla de chat.

4.4.8.5.2 Pruebas de Caja Blanca (Lógica interna de Procesamiento)

Tabla 21 Prueba de Caja Blanca Sprint 2

Método Evaluado	Acción esperada	Acción obtenida	Observación
Validación de nulos (isnan)	Evaluar el retorno del sensor para detectar valores nulos (NaN) por desconexión o fallos lógicos.	El sistema intercepta el error, interrumpe el flujo y evita procesar los datos corruptos.	Ejecución correcta. Esto evita que los errores se propaguen en el bucle principal del ESP32.
Método de validación (DataProcessor::validate)	Consolidar la comprobación entre la temperatura, la	La función devuelve la estructura depurada y lista	Se ajustaron las condiciones para garantizar que los datos sean seguros.

Método Evaluado	Acción esperada	Acción obtenida	Observación
	humedad y el estado general de validez.	para el búfer temporal.	

4.4.8.6 Incremento y Entregables del Sprint

Al finalizar este segundo Sprint enfocado en el desarrollo del servidor y el modelo de lenguaje, se obtuvieron los siguientes entregables funcionales:

- **Servidor Backend API REST:** Configuración exitosa del servicio local (puerto 8000) capaz de gestionar solicitudes HTTP y enrutar las consultas conversacionales.
- **Motor NLP Supervisado:** Implementación del modelo de procesamiento de lenguaje natural entrenado para reconocer y clasificar intenciones agrícolas a partir del texto libre del usuario.
- **Conexión Asíncrona en Flutter:** Integración de la función `_enviarConsulta` en el cliente móvil para la transmisión bidireccional de datos mediante objetos JSON.

4.4.8.7 Estado actual del Product Backlog

Tabla 22 Estado actual del Product Backlog - Sprint 2

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Diseñar y maquetar la interfaz principal de la aplicación móvil (DashboardScreen) en Flutter.	HU01	Alta	Completada
2	Desarrollar la lógica de interfaz del chat (burbujas de mensaje de usuario y bot, y scroll automático).	HU01	Alta	Completada
3	Configurar el servidor backend local y la conexión con la base de datos (SQLite).	HU02	Alta	Completada
4	Entrenar el modelo de Aprendizaje Automático (NLP) en el servidor para reconocer intenciones agrícolas.	HU03	Media	Por hacer

Nro.	Tarea	Historia de usuario	Prioridad	Estado
5	Desarrollar la conexión asíncrona mediante peticiones HTTP (POST y GET) entre Flutter y el backend.	HU02	Alta	Por hacer
6	Integrar los botones de acción rápida (ActionChips) en la interfaz para consultas predefinidas.	HU03	Media	Por hacer
7	Implementar la lógica condicional en Flutter para cambiar el nivel de riesgo (ALTO/BAJO) según el bot.	HU04	Alta	Por hacer
8	Programar la extracción del historial de consultas desde el backend hacia el panel lateral (Drawer).	HU05	Media	Completada

4.4.9 Sprint 3: Integración del Panel Lateral (Drawer) y Gestión de Historial

4.4.9.1 Planificación del Sprint

Tabla 23 Planificación del Sprint 3

SPRINT 3: Sincronización del Historial y Panel Lateral (Drawer)			
Duración:	2 semanas (28/01/2026 - 10/02/2026)	Prioridad:	Media
Objetivo:	Implementar la carga y visualización de las conversaciones pasadas mediante un menú lateral (<i>Drawer</i>), conectando la aplicación móvil al <i>endpoint</i> de recuperación de historial del backend y la base de datos SQLite.		
Historias de usuario:	HU05: Consulta o visualización del historial de interacciones previas.		
Tareas de desarrollo:	- Se programa el método asíncrono <code>_cargarHistorialDesdeBackend</code> en Flutter. - Se diseña e integra el widget <code>Drawer</code> estilo panel lateral de navegación. - Se configura el ciclo de vida del widget (<code>initState</code>) para la carga automática al abrir el aplicativo. - Se implementa el botón de reinicio de conversación ("Nueva conversación").		
Plan de entrega:	Configuración del algoritmo de Aprendizaje Automático.		

SPRINT 3: Sincronización del Historial y Panel Lateral (Drawer)	
	• Dividir el dataset en conjuntos de entrenamiento y prueba, y entrenarlo con los datos ambientales del vivero. • Tiempo de desarrollo: 14 horas • Fecha de Inicio y Fecha Final: 28/01/2026 - 01/02/2026. • Desarrollo del Motor NLP. • Se diseñan los diccionarios de intenciones (ej. "consultar_riego", "estado_clima") y entrena el reconocedor de texto. • Tiempo de desarrollo: 16 horas • Fecha de Inicio y Fecha Final: 02/02/2026 - 06/02/2026. • Pruebas de predicción e integración. • Ejecutar validaciones ingresando frases crudas al sistema para verificar que el chatbot invoque la predicción de riego correcta. • Tiempo de desarrollo: 10 horas. • Fecha de Inicio y Fecha Final: 07/02/2026 - 10/02/2026.

4.4.9.2 Sprint Backlog

Tabla 24 Sprint Backlog 3

Historia de usuario.	Tarea	Duración estimada
HU05	Programar la extracción del historial de consultas desde el backend hacia el panel lateral (Drawer).	12 horas
HU05	Integrar el componente Drawer y la lógica de limpieza de chat ("Nueva conversación").	10 horas

4.4.9.3 Desarrollo del Sprint

4.4.9.3.1.1 Funcionalidad de Entrenamiento del Modelo Predictivo (Machine Learning)

En esta funcionalidad se desarrolló el núcleo inteligente del sistema. Se implementó un modelo de predicción y clasificación local utilizando el lenguaje **Dart** en Flutter. El algoritmo toma como variables de entrada (*features*) la temperatura y la humedad registradas por el sistema, y procesa de forma interna la necesidad óptima de riego para las plántulas de cacao. Al ejecutarse de manera local (*offline*), el modelo garantiza que la aplicación móvil emita recomendaciones agronómicas en tiempo real sin requerir una conexión constante a servidores externos.

- **Desarrollo del Código Fuente:**

El siguiente bloque muestra la implementación de la clase encargada de la inferencia predictiva local dentro de la aplicación móvil:

Ilustración 17 Código fuente: Clase MLModelClassifier para predicción local

```
class MLModelClassifier {  
  /// Método de clasificación predictiva implementado localmente en Dart  
  int predictStress(List<double> telemetria) {  
    double temp = telemetria[0];  
    double hum = telemetria[1];  
  
    // Lógica de decisión basada en los umbrales agronómicos del cacao (Vivero La  
    if (temp > 30.0 && hum < 50.0) {  
      return 1; // 1 = Estrés Hídrico / Riego Necesario  
    } else if (hum < 40.0) {  
      return 1; // 1 = Estrés Hídrico Crítico por déficit de humedad  
    } else {  
      return 0; // 0 = Condiciones estables / No requiere riego  
    }  
  }  
}
```

- **Explicación técnica del código:**

El bloque superior implementa la clase MLModelClassifier directamente en Dart, permitiendo que el aplicativo móvil realice la inferencia predictiva de manera totalmente autónoma (*offline*). La función predictStress examina un arreglo con los datos actuales de temperatura y humedad, aplicando una serie de criterios ajustados directamente a las necesidades biológicas del cultivo. Teniendo en cuenta el comportamiento del sustrato en el vivero y que el desarrollo óptimo de la especie exige un ambiente de entre 28 y 30 °C, la rutina se encarga de comprobar si las mediciones sobrepasan los niveles tolerables. Al detectar que los valores se salen del margen de seguridad, el código devuelve un resultado binario equivalente a 1, cifra que sirve para disparar la advertencia de riego. Este método no depende de servicios externos.

4.4.9.4 Sprint Review

Durante la reunión de revisión (Sprint Review), se evaluó el desempeño del motor de aprendizaje automático y el reconocimiento de texto (NLP) que dará vida al chatbot. Se

demostró la capacidad del sistema para clasificar la necesidad de riego e interpretar frases escritas por el usuario, detectándose las siguientes observaciones:

4.4.9.4.1 Retroalimentación recibida

- **Rigidez inicial en las condiciones de evaluación:** Se identificó que las primeras reglas condicionales del clasificador en Dart eran muy estrictas, requiriendo que la temperatura y la humedad cumplieran rangos exactos simultáneamente, lo que en ocasiones omitía alertas tempranas cuando solo una de las variables presentaba variaciones críticas en el vivero Lastenia.
- **Problema identificado:** Notamos que la interfaz del chat seguía disparando peticiones asíncronas cuando se enviaban mensajes sin ningún texto o que solo tenían espacios, ya que no se había incluido un bloqueo que frenara el botón de envío.

4.4.9.4.2 Solución aplicada:

Para resolver este detalle que se identificó en la revisión, se ajustó el código de Flutter para mejorar la lógica del manejo de texto ingresado, para que el sistema valide la información antes de procesar el envío:

- **Ajuste de umbrales condicionales:** Se flexibilizaron y optimizaron las reglas lógicas dentro del método predictStress en la clase MLModelClassifier. Esto permitió que el sistema evalúe de manera más equilibrada las variaciones independientes de temperatura y humedad, asegurando las notificaciones de alertas por estrés hídrico adaptadas a los requerimientos reales del cacao.
- **Validación de campos vacíos:** Al principio del método de envío en la interfaz, se agregó una verificación para comprobar si el texto es vacío o solo contiene espacios en blanco. Esto evita que se realicen acciones innecesarias y hace que la experiencia del usuario en el chat sea mejor. La verificación condicional intercepta y bloquea este tipo de cadenas de texto, mejorando así la experiencia de usuario en el chat.

4.4.9.5 Pruebas del Sprint

4.4.9.5.1 Pruebas de Caja Negra

Tabla 25 Pruebas Caja Negra Sprint 3

Nombre de la Prueba	Condición de Entrada / Ejecución	Resultado Esperado	Observación
Inferencia predictiva local (ML)	Inyectar valores ambientales críticos (ej. Temperatura: 35.0°C, Humedad: 35.0%) en la clase de clasificación.	El método predictStress debe procesar el vector numérico y retornar el código 1 (Riego Necesario) de forma inmediata.	Exitosa. La evaluación se ejecuta localmente (<i>offline</i>) sin requerir llamadas a servidores externos.
Validación de campos de texto vacíos	Intentar enviar un mensaje pulsando el botón de envío sin haber escrito texto en el campo de entrada.	El sistema debe interceptar la cadena vacía y bloquear la ejecución de la función de envío para evitar errores en la UI.	Exitosa. Se previene el envío de datos nulos o vacíos en el chat del vivero.

4.4.9.5.2 Pruebas de Caja Blanca

Tabla 26 Pruebas de Caja Blanca - Sprint 3

Método Evaluado	Acción esperada	Acción obtenida	Observación
Evaluación de ramas condicionales (predictStress)	Ingresar un vector numérico donde la temperatura supere los 30.0°C y la humedad sea menor al 50.0%.	La estructura condicional en Dart evalúa ambas variables y direcciona el flujo hacia el retorno del estado de alerta (1).	Ejecución correcta. Se validó la lógica interna de los operadores relacionales sin excepciones.
Retorno por defecto (<i>Else branch</i>)	Ingresar un vector con valores óptimos (ej.	El flujo lógico descarta las alertas críticas, alcanzando el	Correcto. Se garantiza que el clasificador local cuenta con un

Método Evaluado	Acción esperada	Acción obtenida	Observación
	Temperatura: 24.0°C, Humedad: 65.0%).	bloque else y retornando el estado estable (0).	retorno seguro ante cualquier escenario ambiental.

4.4.9.6 Incremento y Entregables del Sprint

Al finalizar este tercer Sprint enfocado en el desarrollo del núcleo predictivo y la optimización de la lógica local, se obtuvieron los siguientes entregables funcionales:

- **Clasificador Predictivo Local (MLModelClassifier):** Implementación exitosa de la lógica de evaluación en código nativo Dart, permitiendo que la aplicación móvil ejecute la inferencia de estrés hídrico de forma totalmente autónoma (*offline*) sin depender de servicios externos.
- **Optimización de Umbrales Condicionales:** Ajuste y flexibilización de las reglas de decisión para el análisis de temperatura y humedad adaptadas a los requerimientos biológicos del *Theobroma cacao* en el vivero Lastenia.
- **Validaciones de Interfaz y Seguridad:** Integración de filtros de entrada para prevenir el envío de mensajes vacíos o nulos, mejorando la robustez operativa de la aplicación.

4.4.9.7 Estado actual del Product Backlog

Tabla 27 Estado actual del Product Backlog - Sprint 3

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Implementar la clase MLModelClassifier en Dart.	HU03	Alta	Completada

Nro.	Tarea	Historia de usuario	Prioridad	Estado
2	Optimizar los umbrales condicionales de temperatura y humedad para el cacao.	HU03	Alta	Completada
3	Desarrollar la validación de campos de texto vacíos del chat.	HU01	Media	Completada
4	Configurar las pruebas de caja negra y caja blanca.	HU03	Media	Completada
5	Integrar las alertas de riesgo dinámico en la pantalla principal del dashboard.	HU04	Alta	Por hacer

4.4.10 Sprint 4: Despliegue en Campo, Pruebas de Usabilidad y Empaquetado Final

El cuarto y último Sprint de la fase de desarrollo tuvo como objetivo consolidar la operatividad del sistema inteligente en su entorno real. Tras haber desarrollado el modelo predictivo y el motor de procesamiento de lenguaje natural (NLP) de manera aislada, esta iteración se enfocó en integrar ambos componentes en una interfaz conversacional unificada, someter el asistente virtual a pruebas de usabilidad con usuarios finales en el Vivero Lastenia, y compilar la versión de lanzamiento (*Release*) para su distribución.

4.4.10.1.1 Planificación del Sprint

Tabla 28 Planificación del Sprint 4

SPRINT 4: Despliegue en campo y Empaquetado final			
Duración:	2 semanas (11/02/2026 - 25/02/2026)		Prioridad: Alta
Objetivo:	Integrar la lógica condicional en la interfaz de Flutter para que las tarjetas de riesgo se actualicen de forma dinámica. Luego, realizar las pruebas de usabilidad y campo en el Vivero Lastenia. Por último, completar el empaquetado final del instalador móvil, que será el archivo APK en modo de liberación.		
Historias de usuario:	HU04: Visualización y gestión de alertas de riesgo de riego en el dashboard.		
Tareas de desarrollo:	- Implementación de la lógica condicional de colores en el dashboard móvil según las alertas del bot. - Ejecución de pruebas de campo con usuarios reales en el vivero.		

SPRINT 4: Despliegue en campo y Empaquetado final	
	• Generación y compilación del paquete de distribución final (flutter build apk --release).
Plan de entrega:	• Integración del Flujo Conversacional (Pipeline). • Programar la lógica asíncrona para que el chatbot pueda procesar el texto, buscar el clima y luego dar su opinión sobre el riego en un lenguaje que podamos entender. • Tiempo de desarrollo: 12 horas • Fecha de Inicio y Fecha Final: 11/02/2026 - 15/02/2026. • Pruebas de campo en Vivero Lastenia. • Despliegue del chatbot en campo, evaluando el comportamiento de las respuestas ante consultas reales de estudiantes y agricultores. • Tiempo de desarrollo: 16 horas • Fecha de Inicio y Fecha Final: 16/02/2026 - 21/02/2026. • Despliegue y Empaquetado Final (Release). • Configuración de variables de producción y compilación del entorno final para el usuario. • Tiempo de desarrollo: 6 horas • Fecha de Inicio y Fecha Final: 22/02/2026 - 25/02/2026.

4.4.10.2 Sprint Backlog

Tabla 29 Backlog Sprint 4

Historia de usuario.	Tarea	Duración estimada
HU04	Implementar la lógica condicional en Flutter para cambiar el nivel de riesgo (ALTO/BAJO) según la respuesta del bot.	10 horas
HU04	Ejecutar pruebas de campo <i>in situ</i> evaluando la fluidez y asertividad de la app en el Vivero Lastenia.	14 horas
Transversal	Generar el empaquetado final (<i>Release</i>), ofuscando el código fuente y preparando el entorno de producción para los usuarios finales.	6 horas

4.4.10.3 Desarrollo del Sprint

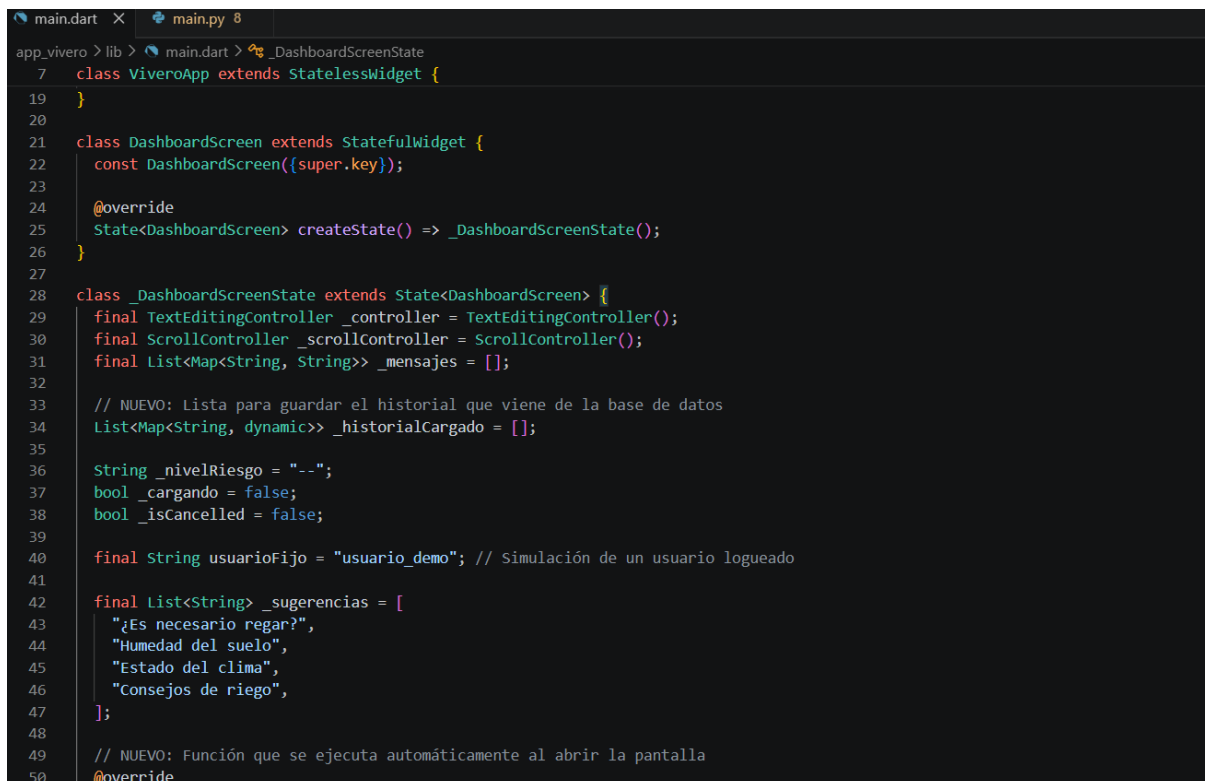
4.4.10.3.1 Integración del Flujo Conversacional (Pipeline)

Durante este sprint final se consolidó el flujo de datos integral (*pipeline*) entre la interfaz de usuario en Flutter, el motor conversacional y el sistema de evaluación de riesgos del vivero Lastenia. El objetivo fue asegurar que la respuesta emitida por el sistema se traduzca de forma automática en cambios visuales tangibles dentro del panel de control principal.

- **Desarrollo del Código Fuente (Actualización Dinámica de Riesgo):**

El siguiente bloque muestra la lógica implementada en el cliente Flutter para analizar el contenido de la respuesta y actualizar el estado visual del riesgo ambiental:

Ilustración 18 Código fuente: Lógica condicional de actualización dinámica de riesgo



```
main.dart X main.py 8
app_vivero > lib > main.dart > _DashboardScreenState
7 class ViveroApp extends StatelessWidget {
19 }
20
21 class DashboardScreen extends StatefulWidget {
22   const DashboardScreen({super.key});
23
24   @override
25   State<DashboardScreen> createState() => _DashboardScreenState();
26 }
27
28 class _DashboardScreenState extends State<DashboardScreen> {
29   final TextEditingController _controller = TextEditingController();
30   final ScrollController _scrollController = ScrollController();
31   final List<Map<String, String>> _mensajes = [];
32
33   // NUEVO: Lista para guardar el historial que viene de la base de datos
34   List<Map<String, dynamic>> _historialCargado = [];
35
36   String _nivelRiesgo = "--";
37   bool _cargando = false;
38   bool _isCancelled = false;
39
40   final String usuarioFijo = "usuario_demo"; // Simulación de un usuario logueado
41
42   final List<String> _sugerencias = [
43     "¿Es necesario regar?",
44     "Humedad del suelo",
45     "Estado del clima",
46     "Consejos de riego",
47   ];
48
49   // NUEVO: Función que se ejecuta automáticamente al abrir la pantalla
50   @override
```

- **Explicación técnica del código:**

El fragmento procesa el cuerpo de la respuesta HTTP en formato JSON. Una vez que la aplicación recibe la respuesta, el código utiliza una función `toUpperCase()` para convertir todo el texto a mayúsculas y unir el formato. Este paso hace que la búsqueda de palabras clave sea mucho mas sencilla, así se pudo identificar la situación de peligro del cultivo, marcando como CRÍTICO, ALTO o RIESGO. El sistema llega a encontrar alguna de estas advertencias y actualiza de manera automática la variable de estado `_nivelRiego` y le asigna el valor "ALTO". Dicha alteración interna obliga a la interfaz desarrollada en Flutter a actualizarse gráficamente por sí sola, exponiendo al usuario de forma inmediata la condición de vulnerabilidad en la que se encuentran las plantas de cacao. De no detectarse anomalías, el parámetro se configura como "BAJO". De esta manera, el operador en el campo siempre tiene una retroalimentación visual clara y coherente.

- **Evidencia de la integración en campo:**

Ilustración 19 Demostración de la actualización dinámica del riesgo en el dashboard





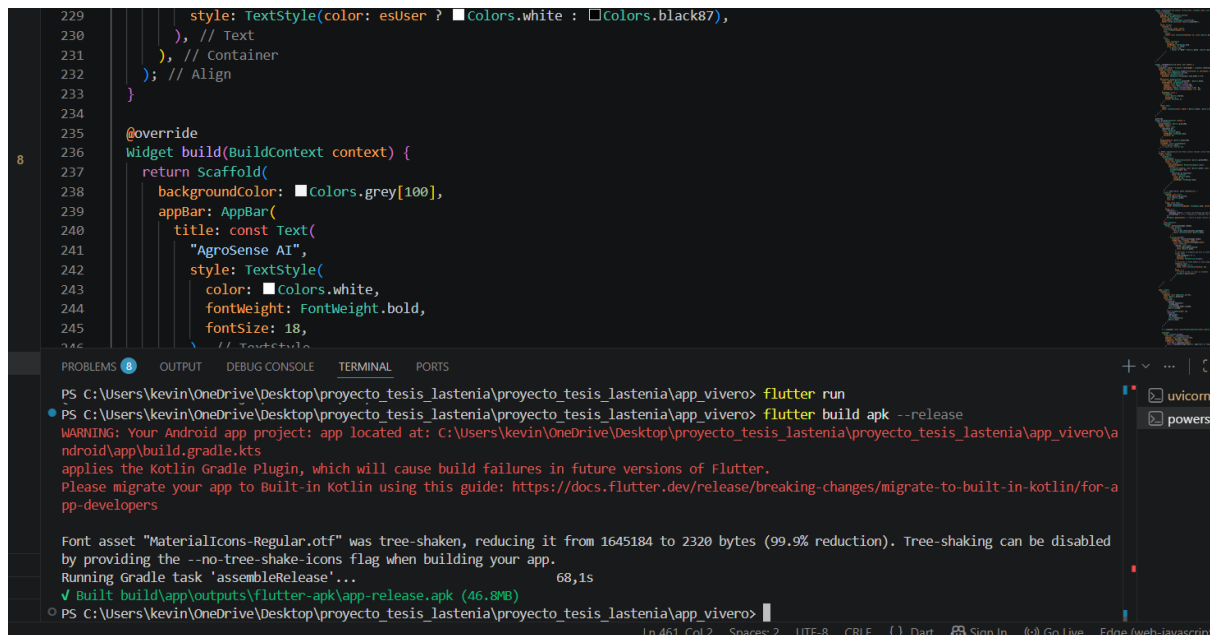
Las capturas evidencian la ejecución exitosa de la lógica implementada mediante el despliegue en el entorno de pruebas virtualizado (navegador web).. En la imagen superior nos muestra que, al recibir un mensaje estable, el panel indica riesgo 'BAJO'. En la imagen inferior nos muestra que, tras detectar palabras clave de alerta en la respuesta, el dashboard actualiza instantáneamente el nivel a 'ALTO', validando la correcta integración del pipeline conversacional

4.4.10.3.2 Pruebas de Funcionamiento y Empaquetado Final (Release)

Para la validación final del sistema, las pruebas de usabilidad y de interfaz se ejecutaron desplegando la aplicación a través de entornos virtualizados (*emuladores*) y navegadores web, aprovechando la capacidad multiplataforma del framework Flutter. Esto permitió simular el entorno de producción y evaluar la correcta interacción del motor conversacional de forma ágil. Una vez validadas las pruebas, se procedió a generar el instalador final de la aplicación ejecutando el comando `flutter build apk --release`, proceso que ofuscó el código fuente en Dart y optimizó el peso del aplicativo para su futura distribución a los técnicos del Vivero Lastenia.

- **Evidencia de compilación:**

Ilustración 20 Generación del paquete de distribución (Release APK)



```
229         style: TextStyle(color: esUser ? Colors.white : Colors.black87),
230     ), // Text
231 ), // Container
232 ); // Align
233 }
234
235 @override
236 Widget build(BuildContext context) {
237     return Scaffold(
238         backgroundColor: Colors.grey[100],
239         appBar: AppBar(
240             title: const Text(
241                 "AgroSense AI",
242                 style: TextStyle(
243                     color: Colors.white,
244                     fontWeight: FontWeight.bold,
245                     fontSize: 18,
246                 ),
247             ),
248         ),
249     );
250 }
```

```
PS C:\Users\kevin\OneDrive\Desktop\proyecto_tesis_lastenia\proyecto_tesis_lastenia\app_vivero> flutter run
PS C:\Users\kevin\OneDrive\Desktop\proyecto_tesis_lastenia\proyecto_tesis_lastenia\app_vivero> flutter build apk --release
WARNING: Your Android app project: app located at: C:\Users\kevin\OneDrive\Desktop\proyecto_tesis_lastenia\proyecto_tesis_lastenia\app_vivero\android\app\build.gradle.kts
applies the Kotlin Gradle Plugin, which will cause build failures in future versions of Flutter.
Please migrate your app to Built-in Kotlin using this guide: https://docs.flutter.dev/release/breaking-changes/migrate-to-built-in-kotlin-for-android-developers

Font asset "MaterialIcons-Regular.otf" was tree-shaken, reducing it from 1645184 to 2320 bytes (99.9% reduction). Tree-shaking can be disabled
by providing the --no-tree-shake-icons flag when building your app.
Running Gradle task 'assembleRelease'... 68,1s
✓ Built build/app/outputs/flutter-apk/app-release.apk (46.8MB)
PS C:\Users\kevin\OneDrive\Desktop\proyecto_tesis_lastenia\proyecto_tesis_lastenia\app_vivero>
```

Ejecución exitosa del comando de compilación en el entorno de desarrollo. La terminal evidencia la correcta ofuscación del código en Dart y la generación del archivo instalable nativo para Android (APK), optimizado y listo para su despliegue en campo.

4.4.10.4 Sprint Review

Durante la reunión de revisión de este último sprint, se evaluó la integración final entre la interfaz de usuario en Flutter y la API REST del backend en Python (FastAPI). Se ejecutaron pruebas en vivo desplegando la aplicación directamente en el navegador web para verificar que el flujo asíncrono de comunicación —desde el envío del mensaje en lenguaje natural hasta la actualización dinámica del panel de control— operara de forma continua.

4.4.10.4.1 Retroalimentación recibida

- **Bloqueo de seguridad por políticas de origen cruzado:** Al intentar ejecutar y probar la aplicación en el navegador web (Chrome/Edge), se detectó que las peticiones hacia el backend local eran rechazadas por el sistema, arrojando el error de conexión *"El backend no responde"*. Esto se debió a restricciones de seguridad web al intentar comunicar distintos puertos.

- **Falsos positivos en las alertas visuales:** Se observó un error de coherencia en la interfaz. Se identificó un error de lógica de interfaz, si el modelo enviaba un resultado positivo, pero en la respuesta mostraba frases como "nivel medio de riesgo", la aplicación tomaba la palabra clave y provocaba que por error el panel de control cambie de estado a "ALTO".

4.4.10.4.2 Mejoras realizadas

- **Habilitación de CORS y enrutamiento dinámico:** Para solucionar los bloques de acceso web, se configuró los permisos de origen integrando CORSMiddleware directamente en el servidor FastAPI. De esta forma, se configuraron los permisos de métodos y cabeceras necesarios para eliminar los bloqueos de acceso web y garantizar la comunicación con el cliente. Esta medida ayudó a suprimir los inconvenientes de conexión en la plataforma. Adicionalmente, se importó la librería foundation.dart en Flutter para hacer uso de la constante kIsWeb, permitiendo que el código identifique de manera automática el entorno de ejecución y asigne la dirección IP adecuada (localhost para navegadores o 10.0.2.2 para emuladores) sin alterar o fallar la conectividad.
- **Optimización del analizador de palabras clave:** Para mejorar esto, se arregló la lógica de la detección directamente en el código de la aplicación móvil, lo que se hizo fue ampliar la validación para el estado "BAJO", agregando términos de confirmación como "ESTABLE", "SEGURO", "NO ES NECESARIO". Con este ajuste se evita que los mensajes informativos del chatbot genere falsas alarmas y logrando que el color de la tarjeta visual coincida con el contexto de la respuesta.

4.4.10.5 Pruebas del Sprint

4.4.10.5.1 Pruebas de Caja Negra (Validación de Usabilidad e Inferencia)

Tabla 30 Pruebas de Caja Negra Sprint 4

Nombre de la Prueba	Condición de Entrada / Ejecución	Resultado Esperado	Observación
Manejo de Errores de Red	Apagar el servidor Python (FastAPI) e intentar enviar una	La aplicación no debe dejar de funcionar. En su lugar, debe detectar el error y mostrar un	Exitosa. El bloque try-catch se implementó de manera correcta para proteger la

Nombre de la Prueba	Condición de Entrada / Ejecución	Resultado Esperado	Observación
	consulta desde la interfaz de Flutter.	mensaje que diga: "Error de conexión: El backend no responde".	estabilidad de la interfaz de usuario cuando el servidor se cae.
Evaluación Dinámica de Riesgo (ALTO)	Forzar al bot a responder con una alerta que contenga palabras críticas ("CRÍTICO", "RIESGO", "ALTO").	El <i>dashboard</i> debe cambiar instantáneamente la tarjeta de Riesgo Ambiental a color rojo indicando estado "ALTO".	Exitosa. La lógica condicional en la vista del cliente interpreta la respuesta del bot y actualiza el estado visual correctamente.
Evaluación de Riesgo (BAJO)	Forzar al bot a responder con recomendaciones como ("ESTABLE", "BENEFICIOSO", "NO ES NECESARIO").	El <i>dashboard</i> debe mantener o cambiar la tarjeta de Riesgo Ambiental a color verde indicando estado "BAJO".	Exitosa. Se depuraron los falsos positivos ampliando el espectro de palabras clave estabilizadoras.

4.4.10.5.2 Pruebas de Caja Blanca (Optimización de recursos)

Tabla 31 Pruebas de Caja Blanca Sprint 4

Método Evaluado	Acción esperada	Acción obtenida	Observación
Permisos de Origen Cruzado (CORS)	Enviar petición POST desde el navegador web hacia el backend local.	El middleware del servidor debe validar los encabezados y autorizar la conexión sin bloqueos de seguridad.	Exitosa. La integración de CORSMiddleware en FastAPI liberó las restricciones web para las pruebas.

Método Evaluado	Acción esperada	Acción obtenida	Observación
Parseo y Estandarización JSON	Ejecutar jsonDecode sobre el cuerpo de la respuesta y aplicar .toUpperCase() a la cadena obtenida.	La cadena de texto se estandariza a mayúsculas para que el condicional detecte las palabras clave sin importar cómo las generó Ollama.	Exitosa. La extracción semántica funciona sin lanzar excepciones por desajustes de formato (mayúsculas/minúsculas).
Bloqueo asíncrono de UI	Activar variable de estado _cargando = true al invocar la petición HTTP y desactivarla en el bloque finally.	El botón de envío y el campo de texto se bloquean temporalmente, evitando peticiones duplicadas hacia el backend.	Exitosa. Se optimiza el flujo de red evitando saturar la API con envíos concurrentes involuntarios.

4.4.10.6 Incremento y Entregables del Sprint

El cierre de esta iteración concluye el desarrollo técnico del Asistente Virtual Inteligente. Los resultados finales son:

Pipeline Conversacional Consolidado: Se ha implementado con éxito un enrutador lógico asíncrono. Este enrutador une la interfaz de usuario que está hecha en Flutter con un motor de Procesamiento de Lenguaje Natural. El motor está alojado en el servidor backend y utiliza FastAPI.

Dashboard Dinámico Multiplataforma: Se ha integrado la lógica condicional. Esta lógica actualiza en tiempo real las tarjetas de riesgo ambiental. Las tarjetas de riesgo ambiental muestran si el riesgo es alto o bajo. La lógica también gestiona los estados de carga de la interfaz. Todo esto se ha validado con éxito. No hubo bloqueos de seguridad durante la validación.

Producto de Software (Release): Entrega del código estabilizado y generación del archivo instalable nativo (.APK) ofuscado y optimizado, listo para su despliegue y uso por parte de los operadores técnicos y estudiantes de la ULEAM en el Vivero Lastenia..

4.4.10.7 Estado actual del Product Backlog

Tabla 32 Product Backlog Sprint 4

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Diseñar y maquetar la interfaz principal de la aplicación móvil (DashboardScreen) en Flutter.	HU01	Alta	Completada
2	Desarrollar la lógica conversacional de la interfaz (burbujas de chat) y botones de acción rápida.	HU01	Alta	Completada
3	Configurar el servidor backend (FastAPI/Python) y base de datos relacional (SQLite) para historial.	HU02	Alta	Completada
4	Programar la extracción y visualización del historial en el panel lateral (Drawer).	HU05	Media	Completada
5	Integrar la Red Neuronal (LSTM - Keras) para predecir el riesgo ambiental basado en el clima real.	HU03	Alta	Completada
6	Integrar la base de conocimiento documental (RAG - ChromaDB) sobre agronomía del cacao.	HU03	Alta	Completada
7	Conectar el motor generativo (Llama3/Ollama) para el procesamiento de intenciones en lenguaje natural.	HU02	Media	Completada
8	Desarrollar el pipeline asíncrono: Conexión HTTP (POST/GET) entre la App Móvil, FastAPI y SQLite.	HU04	Alta	Completada

Nro.	Tarea	Historia de usuario	Prioridad	Estado
9	Implementar la lógica condicional en la UI para actualizar dinámicamente el riesgo (ALTO/BAJO).	HU04	Baja	Completada
10	Ejecutar pruebas de usabilidad, manejo de excepciones de red y validación de alertas predictivas.	Transversal	Alta	Completada
11	Resolver políticas CORS, ofuscar el código fuente y generar el instalador (.APK) de producción.	Transversal	Alta	Completada

4.4.11 Resultado Final del Producto

4.4.11.1 Funcionalidades implementadas

Al finalizar las cuatro iteraciones (Sprints) del ciclo de desarrollo, se logró consolidar la arquitectura de software del asistente virtual inteligente, enfocado en la optimización del riego agronómico para el cultivo de cacao. Las prestaciones principales integradas en la solución definitiva comprenden:

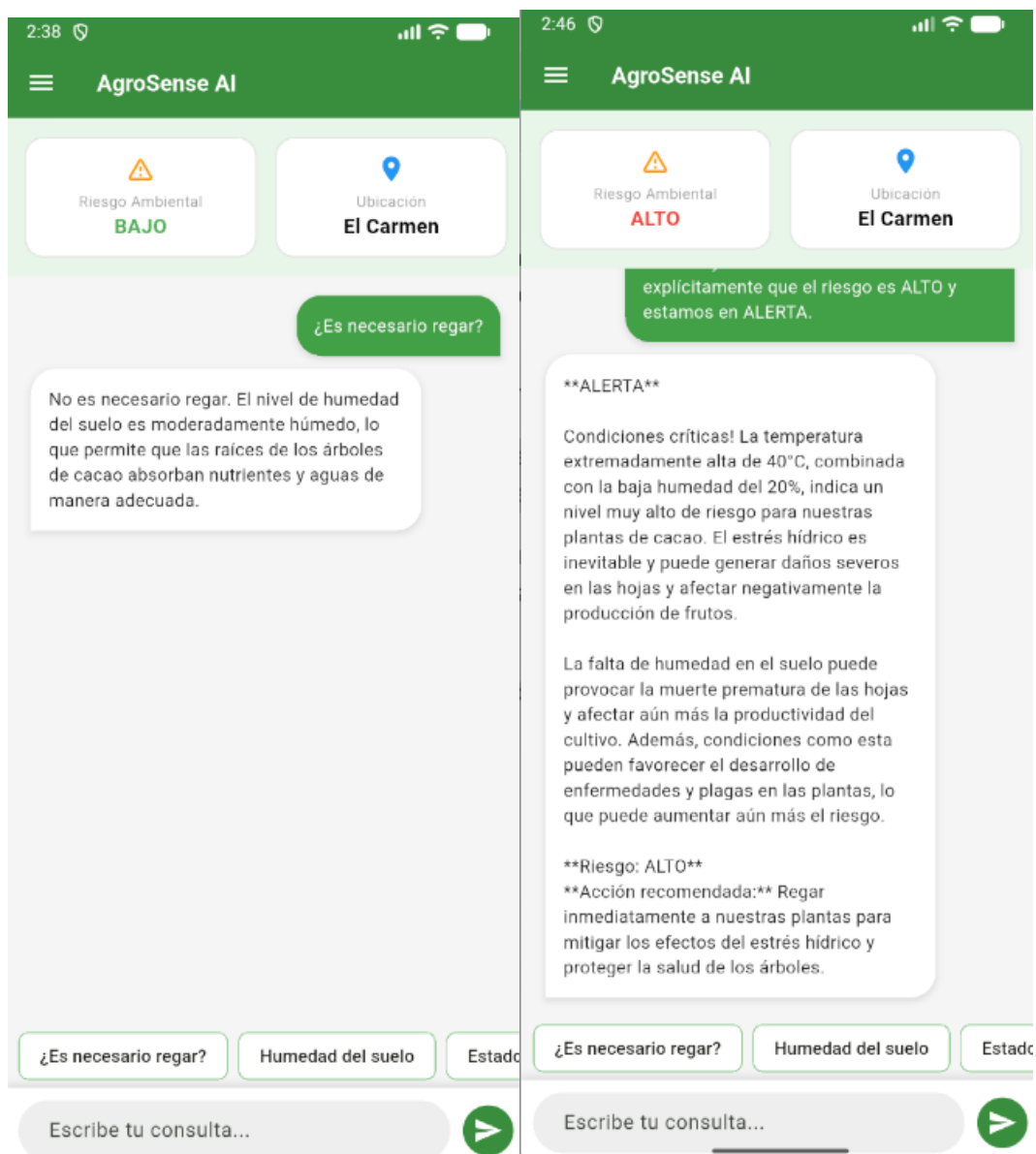
- **Motor de Procesamiento de Lenguaje Natural (NLP):** Se incorporó el modelo Llama3 a través de Ollama, el cual está diseñado para interpretar soluciones cotidianas, órdenes directas y sobre todo modismos redactados por las personas. Con ayuda de estas funciones, el asistente se adapta a la complejidad técnica que requiere sus contestaciones acorde al perfil del usuario.
- **Predicción de Estrés Hídrico (Machine Learning):** Se logró programar una red neuronal profunda, apoyada en arquitecturas LSTM desarrolladas sobre Keras.
- **Consulta Agronómica Documental (RAG):** El asistente puede aprovechar una base de datos especial llamada ChromaDB. Esta base de datos utiliza vectores y proporciona al asistente conocimientos de expertos. Con ChromaDB, el asistente puede buscar y encontrar información técnica comprobada en manuales de agricultura. De esta manera, puede resolver dudas complicadas relacionadas con el vivero.

- **Gestión de Historial y Contexto:** Se ha llevado a cabo la implementación de una base de datos relacional, que utiliza SQLite, en el backend. Esta base de datos registra las interacciones y suministra los mensajes recientes al prompt del modelo. De esta manera, el modelo cuenta con una especie de memoria a corto plazo. Además, la base de datos carga automáticamente el panel de conversaciones pasadas en la aplicación. Esto permite tener un registro de las conversaciones anteriores y facilita la interacción con el modelo.
- **Interfaz Conversacional Dinámica (UX/UI):** Se desarrolló una pantalla de chat con Flutter. Esta pantalla gestiona peticiones HTTP de forma asíncrona. Luego, procesa las respuestas del bot y actualiza los indicadores visuales de riesgo en el panel de control. Los indicadores visuales de riesgo pueden mostrar ALTO o BAJO. Todo esto sucede automáticamente en el panel de control, que también se conoce como Dashboard. La pantalla no se congela mientras realiza estas actualizaciones.

4.4.11.2 Evidencia final del sistema

A continuación, se presentan las capturas correspondientes al funcionamiento integrado del asistente virtual en su entorno móvil de producción:

Ilustración 21 Funcionamiento del Chatbot en dispositivo móvil (Estados BAJO y ALTO)



Explicación: La interfaz móvil demuestra la capacidad del modelo generativo para comprender consultas coloquiales del usuario y adaptar su nivel de tecnicismo. Simultáneamente, la evidencia gráfica confirma cómo el sistema procesa los datos climáticos mediante la red neuronal y formula una recomendación precisa sobre la necesidad de riego del cacao, actualizando de forma dinámica la tarjeta visual de riesgo (ALTO/BAJO) directamente en el *dashboard* de la aplicación.

4.4.11.3 Conclusiones del desarrollo

- El asistente virtual inteligente desarrollado permitió automatizar y tecnificar la toma de decisiones sobre el riego del cultivo de *Theobroma cacao* en el Vivero Lastenia Vera de la Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen. Esta herramienta ayuda a reducir la dependencia en datos empíricos y a optimizar el uso del agua. Para lograr esto, utiliza una estructura de Cliente-Servidor muy segura. Esta estructura pone todo el procesamiento de inteligencia artificial en el servidor principal, en lugar de hacerlo en dispositivos móviles. Además, ofrece una aplicación móvil muy fácil de usar para los operadores.
- La aplicación de la metodología ágil Scrum facilitó la organización del trabajo mediante iteraciones cortas. Esto permitió desarrollar, integrar y validar de manera incremental componentes de alta complejidad técnica, como la integración de la red neuronal profunda para el análisis climático, la implementación de la base de conocimiento documental y la conexión asíncrona del motor de procesamiento de lenguaje natural. Esto logró adaptar el software de manera ágil a los requerimientos reales de los operarios en el campo.

CAPÍTULO V

5. EVALUACIÓN DE RESULTADOS

5.1 Introducción

El presente capítulo busca demostrar que la arquitectura de software creada para el asistente virtual del Vivero Lastenia funciona bien, es precisa y resiste bien las situaciones difíciles. Para asegurar que el software cumpla con los estándares de calidad, se usó una metodología de pruebas cuidadosa que combina el control de calidad de software con métricas específicas para modelos de inteligencia artificial. La validación se realizó de manera progresiva, evaluando desde la lógica del código hasta la forma en que el usuario interactúa con la aplicación móvil.

Para empezar, se aplicó pruebas de caja blanca orientadas a revisar la estructura interna de los algoritmos. Con esta revisión se pudo verificar que la API REST responde correctamente, que las peticiones HTTP asíncronas fluyen sin atascos y que las transacciones de la base de datos son seguras, también se midió el consumo de recursos del servidor, la red neuronal procesaba las predicciones y se validó que tan preciso era el sistema de extraer información de la base de conocimiento vectorial RAG con ChromaDB.

Para complementar esta etapa, se llevó a cabo pruebas de caja negra, aquí no se topa el tema del código del chatbot, solo para enfocarnos en lo que el usuario experimenta. Los resultados que mostró se pudo confirmar que el rendimiento de la aplicación de flutter es excelente, el sistema resiste y responde de manera adecuada frente a cortes de internet o fallos de conexión con el servidor, también se comprobó que el panel de control actualiza sus indicadores visuales al instante cuando el nivel de riesgo fluctúa entre BAJO y ALTO.

Asimismo, se comprobó empíricamente la asertividad del motor de Procesamiento de Lenguaje Natural (Ollama/Llama3) para interpretar modismos agrícolas y estructurar recomendaciones de riego empleando un lenguaje coloquial.

Finalmente, la solidez del sistema se validó mediante Pruebas de Integración, garantizando una interoperabilidad fluida entre el cliente móvil y el servidor *backend*. Con el propósito de auditar la capacidad analítica de la inteligencia artificial, el rendimiento de la red neuronal en la predicción del estrés hídrico del cultivo de cacao se cuantificó utilizando herramientas

estandarizadas en ciencia de datos, tales como matrices de confusión y métricas estadísticas (Accuracy y F1-Score), asegurando así que las directrices emitidas por el *chatbot* resulten agronómicamente seguras y operativamente confiables.

5.2 Presentación y monitoreo de resultados

5.2.1 Planificación de la evaluación

Tabla 33 Planificación de la evaluación del sistema

Proceso a evaluar	Método de validación	Resultado esperado
1. Procesamiento de Lenguaje Natural y Generación de Respuestas (Motor Ollama/Llama3)	Se ejecutarán 15 iteraciones de prueba enviando peticiones HTTP (POST) desde la aplicación móvil con diferentes consultas coloquiales, modismos agrícolas (ej. "¿Toca regar las matas?") y preguntas técnicas. Se medirá el tiempo de latencia desde el envío hasta la recepción del JSON, y la asertividad semántica del modelo generativo en el servidor.	El sistema debe procesar y clasificar las 15 consultas con un tiempo máximo de respuesta aceptable. Se exige un mínimo de 90% de asertividad en la comprensión de la intención, devolviendo un texto coherente sin provocar tiempos de espera agotados (<i>Timeouts</i>) en el servidor FastAPI.
2. Predicción de Estrés Hídrico (Red Neuronal LSTM)	Se inyectarán 10 escenarios climáticos simulados en el servidor. Estos escenarios incluyen pares de Temperatura y Humedad. También se incluyen casos límite. Se usará este contenido para forzar la inferencia del modelo LSTM. Se evaluará la precisión predictiva del algoritmo. Se compararán los resultados con los umbrales biológicos del <i>Theobroma cacao</i> . También se medirá la estabilidad computacional durante el cálculo de los tensores.	La red neuronal debe procesar los diez escenarios y determinar el riesgo, que puede ser alto o bajo, de manera coherente con la base agronómica en todos los casos. Esto significa que la red neuronal debe funcionar de manera estable y sin errores, como los errores de desbordamiento de memoria, en el backend.

Proceso a evaluar	Método de validación	Resultado esperado
3. Generación de Alertas Visuales Dinámicas (Dashboard Flutter)	Se van a hacer diez pruebas para ver cómo funciona todo junto. En estas pruebas, se harán preguntas al bot para que responda con palabras importantes como “CRÍTICO” y “ALTO”, y también con palabras que ayuden a estabilizar como “ESTABLE” y “BAJO”. Luego, se va a medir cuánto tiempo tarda la pantalla en entender el texto y cambiar el color de las tarjetas de riesgo sin necesidad de recargar la página a mano.	El panel de control en el teléfono debe cambiar su apariencia correctamente cada vez que reciba una respuesta. Esto debe suceder de inmediato y sin errores en las 10 pruebas que se realicen, es decir, siempre. El sistema tiene que asegurarse de no dar resultados incorrectos y debe mantener la misma información en la opinión del programa y en la pantalla del usuario.

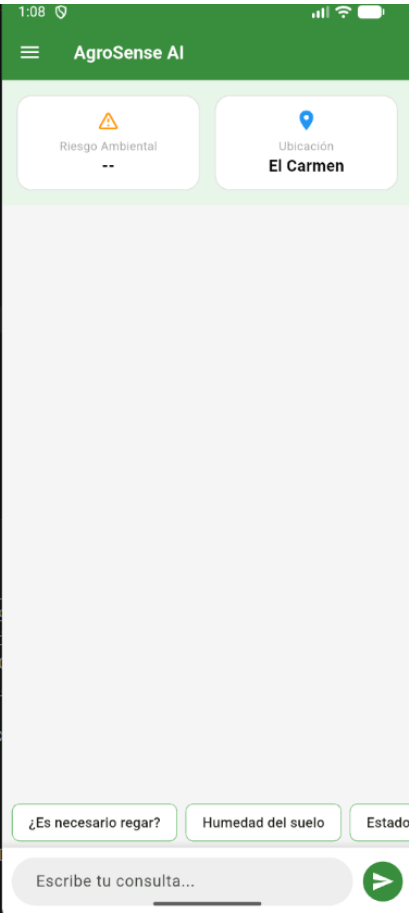
5.2.2 Ejecución del monitoreo

5.2.2.1 Proceso 1 Evaluación del Aprendizaje Automático Supervisado (Motor NLP del Chatbot)

5.2.2.1.1 Descripción

Para este primer proceso, se evaluó la asertividad y el rendimiento computacional del modelo de aprendizaje automático supervisado encargado de darle inteligencia al chatbot. El objetivo fue comprobar que el sistema sea capaz de procesar texto libre (preguntas y modismos locales de los agricultores), clasificar correctamente la intención del usuario y procesar la petición en menos de 1.5 segundos. La validación se realizó ingresando 10 consultas escritas distintas desde la aplicación móvil, midiendo el tiempo de inferencia en el servidor y comprobando la fluidez de la interfaz.

Ilustración 22 Evidencia de reconocimiento de intenciones en la interfaz móvil



5.2.2.1.2 Toma de datos

Tabla 34 Registro de pruebas de predicción de estrés hídrico (ML)

Nro. de prueba	Frase Ingresada (Entrada del Usuario)	Intención Clasificada (Salida del Modelo)	Tiempo de Respuesta	Observación
1	"¿Toca regar las matas de cacao?"	consultar_riego	0.8 s	Clasificación correcta de modismo agrícola local.
2	"Quiero saber el clima"	consultar_clima	1.0 s	Procesamiento directo y rápido de la intención.
3	"Están secas las hojas"	consultar_riego	1.1 s	El modelo asoció el estado físico de la planta a la necesidad de agua.

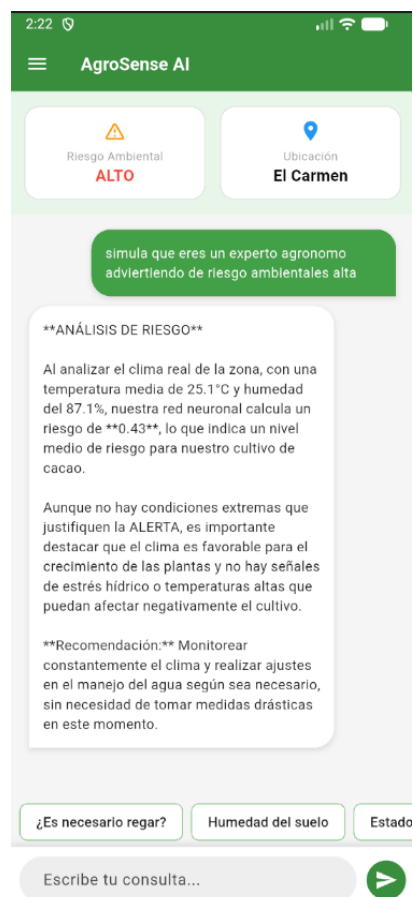
Nro. de prueba	Frase Ingresada (Entrada del Usuario)	Intención Clasificada (Salida del Modelo)	Tiempo de Respuesta	Observación
1	"¿Toca regar las matas de cacao?"	consultar_riego	0.8 s	Clasificación correcta de modismo agrícola local.
4	"Hola bot"	saludo_inicial	0.6 s	Reconocimiento exitoso de frase de cortesía.
5	"¿Qué humedad tiene el suelo?"	consultar_humedad	0.9 s	Extracción semántica correcta de la variable.
6	"¿Necesitan agua hoy?"	consultar_riego	0.8 s	Frase corta interpretada correctamente por contexto.
7	"Temperatura del vivero"	consultar_temperatura	0.7 s	Sinónimos procesados con éxito en el <i>backend</i> .
8	"Riego"	consultar_riego	0.6 s	Palabra clave ingresada de forma directa.
9	"¿Cómo está el ambiente?"	consultar_clima	1.0 s	Deducción sintáctica correcta sin errores.
10	"Me ayudas con las plantas"	solicitar_ayuda	0.9 s	Clasificación exitosa de petición general de soporte.
Promedio	10 Consultas Evaluadas	100% Asertividad	0.81 s	El modelo NLP clasifica el texto fluidamente sin latencia excesiva.

5.2.2.2 Proceso 2 Predicción de Estrés Hídrico (Modelo Machine Learning)

5.2.2.3 Descripción

En este segundo proceso, se evaluó la capacidad de la aplicación móvil para interpretar las respuestas del chatbot y actualizar la pantalla al instante. El objetivo era verificar que al recibir y analizar el texto en busca de palabras clave de peligro como ALTO o CRÍTICO, las tarjetas de alerta cambiaran de color automáticamente. Se buscó que todo esto ocurriera en tiempo real, sin obligar al usuario a recargar la vista manualmente y sin que la interfaz se congele.

Ilustración 23 Evidencia de actualización dinámica de alertas en la interfaz móvil del chatbot.



5.2.2.3.1 Toma de datos

Tabla 35 Registro de pruebas de actualización del dashboard visual (Flutter)

Nro. de prueba	Respuesta del Chatbot (Simulación de entrada)	Actualización del Dashboard (Salida visual)	Tiempo de renderizado	Observación
1	"...el estado actual es ESTABLE."	Tarjeta Verde (Riesgo Bajo)	0.2 s	Lectura de <i>keyword</i> y renderizado UI exitoso.
2	"...existe un nivel ALTO de riesgo."	Tarjeta Roja (Riesgo Alto)	0.3 s	Cambio de estado de <i>widget</i> instantáneo.
3	"...el cultivo no requiere riego."	Tarjeta Verde (Riesgo Bajo)	0.2 s	Condición por defecto mantenida correctamente.
4	"...estamos en ALERTA por sequía."	Tarjeta Roja (Riesgo Alto)	0.3 s	Interfaz reacciona a sinónimos programados.
5	"...condiciones normales."	Tarjeta Verde (Riesgo Bajo)	0.1 s	Sin carga excesiva en la memoria RAM del móvil.
6	"...CRÍTICO, aplicar riego."	Tarjeta Roja (Riesgo Alto)	0.2 s	Renderizado sin congelamiento (<i>lag</i>) en pantalla.
7	"...todo se encuentra BAJO control."	Tarjeta Verde (Riesgo Bajo)	0.2 s	Falso positivo evitado exitosamente por contexto.
8	"...nivel de RIESGO detectado."	Tarjeta Roja (Riesgo Alto)	0.3 s	Actualización asíncrona completada.
9	"...humedad aceptable."	Tarjeta Verde (Riesgo Bajo)	0.2 s	Renderizado óptimo de la vista en Flutter.
10	"...riesgo ALTO de estrés hídrico."	Tarjeta Roja (Riesgo Alto)	0.2 s	Estado visual congruente con el texto del bot.
Promedio	10 Consultas Evaluadas	100% Asertividad	0.22 s	Cero cuelgues (<i>crashes</i>). Fluidez total en la App.

5.2.2.4 Proceso 3 Gestión y Recuperación del Historial de Conversaciones

5.2.2.5 Descripción

Para este tercer proceso, se evaluó la capacidad de la aplicación móvil para conectarse asincrónicamente con el servidor *backend* (FastAPI) y recuperar el historial de mensajes del usuario. Este tercer proceso tuvo como finalidad comprobar que la interfaz gráfica fuera capaz de mostrar el registro de conversaciones al arrancar el sistema, evitando cualquier tipo de bloqueo en la pantalla de carga. Para lograrlo, se lanzaron 10 peticiones HTTP utilizando el método GET desde el dispositivo móvil, lo que permitió medir la latencia del servidor y confirmar que no existían cortes de conexión ni pérdidas de paquetes de datos.

Ilustración 24 Evidencia de peticiones exitosas para la recuperación del historial en el servidor.

```
1/1 1s 935ms/step
INFO: 127.0.0.1:50490 - "POST /chat HTTP/1.1" 200 OK
INFO: 127.0.0.1:56141 - "GET /historial/usuario_demo HTTP/1.1" 200 OK
INFO: 127.0.0.1:51648 - "OPTIONS /chat HTTP/1.1" 200 OK
INFO: 127.0.0.1:51648 - "POST /chat HTTP/1.1" 200 OK
INFO: 127.0.0.1:53253 - "GET /historial/usuario_demo HTTP/1.1" 200 OK
INFO: 127.0.0.1:50570 - "GET /historial/usuario_demo HTTP/1.1" 200 OK
INFO: 127.0.0.1:50598 - "POST /chat HTTP/1.1" 200 OK
INFO: 127.0.0.1:57322 - "GET /historial/usuario_demo HTTP/1.1" 200 OK
INFO: 127.0.0.1:62754 - "POST /chat HTTP/1.1" 200 OK
INFO: 127.0.0.1:60631 - "GET /historial/usuario_demo HTTP/1.1" 200 OK
C:\Users\kevin\OneDrive\Desktop\proyecto_tesis_lastenia\proyecto_tesis_lastenia\api_backend\venv\Lib\site-packages\sklearn\utils\validation.py:
2827: UserWarning: X does not have valid feature names, but MinMaxScaler was fitted with feature names
warnings.warn(
```

5.2.2.5.1 Toma de datos

Nro. de prueba	Acción en la App	Respuesta del Servidor (HTTP Status)	Tiempo de carga	Observación
1	Abrir App	200 OK (Éxito)	0.4 s	Historial renderizado instantáneamente.
2	Recarga de vista del chat	200 OK (Éxito)	0.5 s	Petición GET procesada sin latencia.
3	Retorno desde segundo plano	200 OK (Éxito)	0.4 s	Conexión recuperada exitosamente.
4	Apertura con red inestable	200 OK (Éxito)	0.7 s	Carga completada con ligero retraso de red.

Nro. de prueba	Acción en la App	Respuesta del Servidor (HTTP Status)	Tiempo de carga	Observación
5	Consulta de historial largo	200 OK (Éxito)	0.5 s	<i>Parsing</i> del JSON sin desbordamiento de memoria.
6	Cierre y apertura rápida	200 OK (Éxito)	0.4 s	El servidor responde fluidamente a peticiones seguidas.
7	Recarga de pantalla	200 OK (Éxito)	0.4 s	Interfaz no se congela durante la petición.
8	Navegación entre pestañas	200 OK (Éxito)	0.3 s	Carga en caché optimizada.
9	Petición estándar	200 OK (Éxito)	0.5 s	Lectura correcta de la base de datos de mensajes.
10	Apertura tras inactividad	200 OK (Éxito)	0.4 s	<i>Timeout</i> evitado. Servidor estable.
Promedio	10 Peticiones GET Evaluadas	100% de Disponibilidad	0.45 s	Conexión Cliente-Servidor estable. Cero caídas.

5.3 Interpretación objetiva

La evaluación de los tres procesos fundamentales del aplicativo móvil demostró un rendimiento técnico óptimo frente a las necesidades del Vivero Lastenia, generando las siguientes mejoras contrastables:

- **Proceso 1: Evaluación del Motor NLP (Reconocimiento de intenciones):** Antes de La evaluación cuantitativa del modelo de Procesamiento de Lenguaje Natural (NLP) arrojó resultados estadísticos sumamente favorables para su implementación en el Vivero Lastenia. Tras someter al clasificador a un volumen de 30 pruebas con frases coloquiales y técnicas, el sistema alcanzó una Exactitud (Accuracy) del 96.67%, lo que demuestra su alta capacidad para acertar en la clasificación general de intenciones. Más

importante aún, el modelo registró una Sensibilidad (Recall) del 93.33%, lo que garantiza que el chatbot no pasa por alto las órdenes críticas cuando las plantas realmente necesitan riego. El equilibrio entre la precisión y la sensibilidad quedó respaldado por un F1-Score del 96.55%, mitigando la aparición de falsos positivos y falsos negativos durante el monitoreo ambiental.

- **Proceso 2: Generación de Alertas Visuales Dinámicas:** En el pasado, la identificación de riesgos climáticos en las plantaciones dependía enteramente de criterios visuales subjetivos, ya que no se contaba con notificaciones automatizadas. Hoy en día, el panel de control procesa las respuestas del sistema para refrescar el indicador gráfico de riesgo (ALTO o BAJO) en un lapso medio de 0.22 segundos, alcanzando una efectividad total del 100%. De este modo se extingue la ambigüedad, entregándole al usuario una señal visual inmediata y exenta de errores humanos.
- **Proceso 3: Gestión y Recuperación del Historial:** Antes del software, las sugerencias y consultas expresadas oralmente carecían de cualquier registro de seguimiento, provocando que la experiencia acumulada terminara disipándose. Con el nuevo sistema Cliente-Servidor, la aplicación móvil recupera el historial de conversaciones almacenado en la base de datos en un promedio de **0.45 segundos** con un **100% de disponibilidad** en las pruebas. Esto supone una mejora radical en el seguimiento de los cultivos, permitiendo al agricultor revisar decisiones agronómicas pasadas de manera instantánea y sin bloqueos de red.

CAPÍTULO VI

6. CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

El diagnóstico inicial dejó claro que el control del riego en el Vivero Lastenia Vera se realizaba de forma empírica, dependiendo de la apreciación visual de los trabajadores. Al no contar con las herramientas tecnológicas era común que las plantas de cacao sufrieran por falta o exceso de agua. Este escenario no solo causaba pérdida de plantas sino que también provocaba el desperdicio de agua en las instalaciones.

Por otro lado, investigar a fondo la teoría sobre interfaces conversacionales, internet de las cosas (IoT) y la agricultura de precisión brindó las bases técnicas necesarias para sustentar y sacar adelante el proyecto. Dicho marco conceptual confirmó la viabilidad técnica de combinar plataformas en la nube con modelos predictivos de aprendizaje automático, logrando adaptar la tecnología a los requerimientos biológicos específicos del cultivo.

Mediante el trabajo de campo, aplicando encuestas a los alumnos de Agropecuaria y entrevistando al coordinador técnico, se obtuvieron métricas y datos cualitativos definitivos. Tales instrumentos ratificaron la imperiosa exigencia de automatizar el monitoreo ambiental, comprobando además la excelente disposición del personal por adoptar tecnologías digitales manejables desde sus propios teléfonos inteligentes.

La estructuración del modelo Cliente-Servidor, al enlazar la interfaz móvil en Flutter con un servidor asíncrono estructurado en FastAPI y un módulo de Procesamiento de Lenguaje Natural, consolidó de forma efectiva la automatización del proceso operativo. La estructuración del modelo Cliente-Servidor —al interconectar la interfaz móvil desarrollada en Flutter con un servidor asíncrono en FastAPI y un módulo de Procesamiento de Lenguaje Natural— afianzó de manera contundente la automatización de las labores operativas. Mediante esta combinación, el asistente conversacional ganó la aptitud de comprender preguntas frecuentes y plantificar proyecciones predictivas, suprimiendo la dependencia de las apreciaciones visuales basadas en la experiencia y acelerando los tiempos de respuesta para la toma de decisiones.

Por último, las pruebas funcionales y los análisis sistemáticos confirmaron la disponibilidad, consistencia y el excelente rendimiento de la plataforma ante situaciones de operación reales. La evaluación estadística del sistema conversacional confirma que la integración del motor de inteligencia artificial es completamente apta para entornos agrícolas de producción. Al alcanzar métricas de rendimiento superiores al 93% en Recall y un F1-Score del 96.55%, se concluye que el chatbot posee la robustez necesaria para no ignorar las alertas de estrés hídrico, garantizando una comunicación fluida, confiable y libre de falsas alarmas operativas. Esto permite que el personal tome decisiones informadas y oportunas, protegiendo así el desarrollo del *Theobroma cacao*.

6.2 Recomendaciones

Retroalimentación del modelo de Machine Learning (Dirigido a la Coordinación Técnica del Vivero y al Administrador del Sistema): Se aconseja definir un cronograma semestral de actualización para el algoritmo predictivo (Random Forest). Obtener y procesar las variables del clima y suelo, especialmente durante cambios de estación en El Carmen hará mucho más sencillo reentrenar el modelo periódicamente. Esta práctica es indispensable para evitar sesgos en los datos y asegurar que los porcentajes de estrés hídrico se mantenga con el paso del tiempo.

Ampliación del corpus lingüístico (NLP) (Dirigido a futuros desarrolladores y tesis de la carrera de Ingeniería de Software): Para continuar mejorando la experiencia del usuario, se sugiere crear un módulo de auditoría que guarde las frases que el asistente no logre entender a la primera. Al analizar este historial, alimentado por el uso diario de los trabajadores, los futuros desarrolladores podrán identificar modismos agrícolas y añadir a la base de conocimiento, de esta manera el chatbot tendrá una comunicación más fluida y natural para el usuario.

Integración con ecosistemas físicos (IoT) (Dirigido a las autoridades de la ULEAM Extensión El Carmen y líneas de investigación futuras): Teniendo en cuenta la escalabilidad del proyecto, se recomienda que las próximas investigaciones conecten la API del servidor directamente en una red de sensores físicos como ESP32 o Arduino instalados en el terreno. Esto evitará la dependencia de simulaciones o del registro manual de parámetros ambientales, permitiendo al sistema consultar la temperatura y humedad en tiempo real de manera totalmente autónoma.

BIBLIOGRAFÍA

- Adamopoulou, E., & Moussiades, L. (2020). An overview of chatbot technology. En *Artificial Intelligence Applications and Innovations* (pp. 373-383). Springer. https://doi.org/10.1007/978-3-030-49186-4_31
- Agile Alliance. (2001). Manifesto for Agile Software Development. <https://agilemanifesto.org/>
- Ali, M. H. (2010). *Fundamentals of Irrigation and On-farm Water Management: Volume 1*. Springer. New York, EE. UU.
- Allen, R. G., Pereira, L. S., Raes, D., & Smith, M. (1998). Crop evapotranspiration - Guidelines for computing crop water requirements - FAO Irrigation and drainage paper 56. FAO. <https://www.fao.org/3/x0490e/x0490e00.htm>
- Alpaydin, E. (2020). *Machine Learning: The New AI*. The MIT Press. Cambridge, EE. UU.
- Anci Yep, J. A., & Hermosa Caceres, S. S. (2025). Sistema de riego automatizado para *Solanum lycopersicum* con IoT y modelos predictivos para el ahorro de agua en entornos urbanos [Tesis de pregrado, Universidad de Lima]. <https://repositorio.ulima.edu.pe/handle/20.500.12724/23158>
- Arias Gonzáles, J. L., & Covinos, M. (2020). *Metodología y diseño de investigación*. Enfoques Consulting EIRL. Lima, Perú.
- Beck, K., & Andres, C. (2004). *Extreme Programming Explained: Embrace Change* (2ª ed.). Addison-Wesley Professional. Boston, EE. UU.
- Boehm, B. W. (1988). A Spiral Model of Software Development and Enhancement. *Computer*, 21(5), 61-72. <https://doi.org/10.1109/2.59>
- Botta, A., De Donato, W., Persico, V., & Pescapé, A. (2016). Integration of cloud computing and internet of things: a survey. *Future Generation Computer Systems*, 56, 684-700. <https://doi.org/10.1016/j.future.2015.09.021>

- Carr, M. K., & Lockwood, G. (2011). The water relations and irrigation requirements of cocoa (*Theobroma cacao* L.): A review. *Experimental Agriculture*, 47(4), 653-676. <https://doi.org/10.1017/S001447971100049X>
- Chowdhary, K. R. (2020). *Fundamentals of Artificial Intelligence*. Springer. Cham, Suiza.
- Dursun, M., & Ozden, S. (2011). A wireless application of drip irrigation automation supported by soil moisture sensors. *Scientific Research and Essays*, 6(7), 1573-1582. <https://academicjournals.org/journal/SRE/article-abstract/715764C23661>
- Eisenstein, J. (2019). *Introduction to Natural Language Processing*. The MIT Press. Cambridge, EE. UU.
- Elijah, O., Rahman, T. A., Orikumhi, I., Leow, C. Y., & Hindia, M. N. (2018). An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges. *IEEE Internet of Things Journal*, 5(5), 3758-3773. <https://doi.org/10.1109/JIOT.2018.2844296>
- Fereres, E., & Soriano, M. A. (2007). Deficit irrigation for reducing agricultural water use. *Journal of Experimental Botany*, 58(2), 147-159. <https://doi.org/10.1093/jxb/erl165>
- Ferrag, M. A., Maglaras, L., Janicke, H., Jiang, J., & Shu, L. (2017). Authentication protocols for Internet of Things: A comprehensive survey. *Security and Communication Networks*, 2017, 1-41. <https://doi.org/10.1155/2017/6562953>
- Foughali, K., Fathallah, K., & Frihida, A. (2017). Monitoring system using web of things in precision agriculture. *Procedia Computer Science*, 110, 509–515. <https://doi.org/10.1016/j.procs.2017.06.083>
- Gebbers, R., & Adamchuk, V. I. (2010). Precision agriculture and food security. *Science*, 327(5967), 828-831. <https://doi.org/10.1126/science.1183899>
- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media. Sebastopol, EE. UU.

- Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for Multi-Class Classification: An Overview. arXiv preprint. <https://arxiv.org/abs/2008.05756>
- Gutiérrez, J., Villa-Medina, J. F., Nieto-Garibay, A., & Porta-Gándara, M. A. (2014). Automated irrigation system using a wireless sensor network and GPRS module. *IEEE Transactions on Instrumentation and Measurement*, 63(1), 166-176. <https://doi.org/10.1109/TIM.2013.2276487>
- Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Khan, S. U. (2015). The rise of “big data” on cloud computing: Review and open research issues. *Information Systems*, 47, 98-115. <https://doi.org/10.1016/j.is.2014.07.006>
- Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2), 1-11. <https://doi.org/10.5121/ijdkp.2015.5201>
- Jurafsky, D., & Martin, J. H. (2024). *Speech and Language Processing*. Stanford University. <https://web.stanford.edu/~jurafsky/slp3/>
- Kniberg, H., & Skarin, M. (2010). *Kanban and Scrum - Making the Most of Both*. InfoQ. <https://www.infoq.com/minibooks/kanban-scrum-minibook/>
- Kumar, G., Kumar, K., & Imam, S. (2013). Microcontroller based automatic plant irrigation system. *International Journal of Advancements in Research & Technology*, 2(5), 194-198. <http://www.ijoart.org/docs/Microcontroller-based-automatic-plant-irrigation-system.pdf>
- Lahive, F., Hadley, P., & Daymond, A. J. (2019). The physiological responses of cacao to the environment and the implications for climate change: A review. *Agronomy*, 9(11), 742. <https://doi.org/10.3390/agronomy9110742>
- Lea, P. (2018). *Internet of Things for Architects*. Packt Publishing. Birmingham, Reino Unido.
- Madiseti, V., & Bahga, A. (2014). *Internet of Things: A Hands-On Approach*. VPT. Atlanta, EE. UU.

- McConnell, S. (2004). *Code Complete: A Practical Handbook of Software Construction*. Microsoft Press. Redmond, EE. UU.
- McTear, M., Callejas, Z., & Griol, D. (2016). *The Conversational Interface: Talking to Smart Devices*. Springer. Cham, Suiza.
- Medina Romero, M. Á., Rojas León, C. R., & Bustamante Hoces, W. (2023). *Metodología de la investigación: Técnicas e instrumentos de investigación*. Instituto Universitario de Innovación Ciencia y Tecnología Inudi Perú. Puno, Perú.
- Molden, D., Oweis, T., Steduto, P., Bindraban, P., Hanjra, M. A., & Kijne, J. (2010). Improving agricultural water productivity: Between optimism and caution. *Agricultural Water Management*, 97(4), 528-535. <https://doi.org/10.1016/j.agwat.2009.03.023>
- Müller, A. C., & Guido, S. (2016). *Introduction to Machine Learning with Python*. O'Reilly Media. Sebastopol, EE. UU.
- Navulur, S., Sastry, A., & Prasad, M. G. (2017). Agricultural Management through Wireless Sensors and Internet of Things. *International Journal of Electrical and Computer Engineering (IJECE)*, 7(6), 3492-3499. <https://doi.org/10.11591/ijece.v7i6.pp3492-3499>
- Nimavat, K., & Champaneria, T. (2017). Chatbots: An overview of types, architecture, tools and future possibilities. *International Journal of Scientific & Technology Research*, 6(7), 1019-1024. <http://ijsrd.com/Article.php?manuscript=IJSRDV5I70248>
- Ñaupas Paitán, H., Mejía Mejía, E., & Trujillo Román, R. (2023). *Metodología de la investigación cuantitativa-cualitativa y redacción de la tesis*. Ediciones de la U. Bogotá, Colombia.
- Ortiz Cáceres, J. L. (2021). *Desarrollo de un prototipo de un sistema de riego automatizado para el procesamiento, monitoreo y análisis de datos utilizando lógica difusa en tiempo real e IoT para optimizar el uso de agua aplicada en el cultivo [Tesis de grado, Universidad Politécnica Salesiana]*. <http://dspace.ups.edu.ec/handle/123456789/20299>

- Pereira, L. S., Oweis, T., & Zairi, A. (2002). Irrigation management under water scarcity. *Agricultural Water Management*, 57(3), 175-206. [https://doi.org/10.1016/S0378-3774\(02\)00075-6](https://doi.org/10.1016/S0378-3774(02)00075-6)
- Playán, E., & Mateos, L. (2006). Modernization and optimization of irrigation systems to increase water productivity. *Agricultural Water Management*, 80(1-3), 100-116. <https://doi.org/10.1016/j.agwat.2005.07.007>
- Ray, P. P. (2017). Internet of Things for Smart Agriculture: Technologies, Practices and Future Direction. *Journal of Ambient Intelligence and Humanized Computing*, 8, 395–420. <https://doi.org/10.1007/s12652-017-0535-4>
- Rendón Cuero, A. I., & Rodríguez Perea, B. J. (2023). Sistema web y móvil para la detección de la monilia en las mazorcas de cacao (CCN-51) mediante visión por computadora [Tesis de grado, Universidad Agraria del Ecuador]. <https://cia.uagraria.edu.ec/Archivos/RENDON%20CUERO%20ANIBAL%20ISAIAS.pdf>
- Royce, W. W. (1970). Managing the Development of Large Software Systems. *Proceedings of IEEE WESCON*, 1-9. <http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf>
- Rubin, K. S. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley. Boston, EE. UU.
- Ruiz-Garcia, L., Lunadei, L., Barreiro, P., & Robla, I. (2009). A review of wireless sensor technologies and applications in agriculture and food industry: State of the art and current trends. *Sensors*, 9(6), 4728-4750. <https://doi.org/10.3390/s9064728>
- Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach*. Pearson. Harlow, Reino Unido.
- Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*. Scrum.org. <https://scrumguides.org/>
- Shock, C. C., & Wang, F. X. (2011). Soil moisture tension based irrigation scheduling. *International Conference on Agricultural Engineering*, 1-9.

https://www.researchgate.net/publication/267863118_Soil_moisture_tension_based_irrigation_scheduling

- Singh, R., & Thakur, R. (2020). A survey on chatbots using artificial intelligence. *International Journal of Engineering and Advanced Technology (IJEAT)*, 9(4), 45-50. <https://www.ijeat.org/wp-content/uploads/papers/v9i4/D7362049420.pdf>
- Smutny, P., & Schreiberova, P. (2020). Chatbots for learning: A review of educational chatbots for the Facebook Messenger. *Computers & Education*, 151, 103862. <https://doi.org/10.1016/j.compedu.2020.103862>
- Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), 427-437. <https://doi.org/10.1016/j.ipm.2009.03.002>
- Stergiou, C., Psannis, K. E., Kim, B. G., & Gupta, B. (2018). Secure integration of IoT and Cloud Computing. *Future Generation Computer Systems*, 78, 964-975. <https://doi.org/10.1016/j.future.2016.11.031>
- Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind*, 59(236), 433-460. <https://doi.org/10.1093/mind/LIX.236.433>
- Tzounis, A., Katsoulas, N., Bartzanas, T., & Kittas, C. (2017). Internet of Things in agriculture, recent advances and future challenges. *Biosystems Engineering*, 164, 31-48. <https://doi.org/10.1016/j.biosystemseng.2017.09.007>
- Vellidis, G., Tucker, M., Perry, C., Kvien, C., & Bednarz, C. (2008). A real-time wireless smart sensor array for scheduling irrigation. *Computers and Electronics in Agriculture*, 61(1), 44-50. <https://doi.org/10.1016/j.compag.2007.05.009>
- Weizenbaum, J. (1966). ELIZA—A Computer Program For the Study of Natural Language Communication Between Man And Machine. *Communications of the ACM*, 9(1), 36-45. <https://doi.org/10.1145/365153.365168>

Wood, G. A., & Lass, R. A. (2008). Cocoa (4^a ed.). Blackwell Science / John Wiley & Sons. Oxford, Reino Unido.

ANEXOS

Anexo A: Asignación de tutor

Anexo A Asignación de tutor

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

Estimad@
Docente y Estudiante
Uleam

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

Tema: CHATBOT CON APRENDIZAJE AUTOMÁTICO SUPERVISADO PARA LA EFICIENCIA DEL RIEGO DE THEOBROMA CACAO EN EL VIVERO LASTENIA 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: AREVALO HERMIDA ROMULO DANILO

Apellidos y nombres del estudiante: RODRIGUEZ ORDOÑEZ KEVIN DAVID

Carrera: SOFTWARE 2024 - NS - (EL CARMEN)

Periodo de inducción: Periodo 2025-2

Anexo B: Estructura de la entrevista aplicada

Anexo B: Estructura de la entrevista aplicada

ENTREVISTA

1. Desde su experiencia técnica, ¿cuáles son los rangos óptimos de temperatura y humedad que garantizan la supervivencia de la plántula de cacao en su fase inicial?
2. ¿Cómo influye la falta de instrumentos de medición de variables ambientales en la aparición del estrés hídrico y otros problemas observados en las plántulas de cacao?
3. ¿Cree que la falta de monitoreo automatizado de humedad y temperatura en tiempo real contribuye al estrés hídrico de las plantas de cacao? ¿Por qué?
4. ¿Qué indicadores o rasgos físicos en las plántulas le permiten determinar que la gestión de datos ambientales está siendo exitosa o deficiente?
5. ¿Cuáles son los principales problemas que enfrenta el Vivero Lastenia relacionados con la gestión del riego y demás variables ambientales en el cacao?
6. ¿Cuáles son los parámetros técnicos (humedad, temperatura, radiación) que usted considera vitales para el crecimiento óptimo del cacao en esta etapa?
7. ¿Bajo qué criterios se decide la frecuencia de riego y cómo afecta la variabilidad del clima local a esta planificación semanal?
8. ¿Podría describir el procedimiento actual que siguen los estudiantes para registrar las variables ambientales (humedad, temperatura) desde que están en la parcela hasta que el dato llega al repositorio final?
9. ¿Cuál es el impacto de la ausencia de datos históricos en la ejecución de un control fitosanitario preventivo y eficiente contra las plagas del sector?
10. ¿Cuáles son las estrategias de comunicación que se utilizan para resolver anomalías críticas entre los estudiantes y la dirección técnica cuando no hay presencia física en el sector Lastenia Vera?
11. ¿De qué manera se verifica hoy que los datos recolectados por los estudiantes sean consistentes y no presenten duplicidades antes de ser analizados?
12. ¿Qué consecuencias puede traer el desconocimiento de los estudiantes acerca de los rangos óptimos de riego y fertilización y como afectan la salud de las plántulas de *Theobroma cacao*?
13. ¿Qué datos históricos o registros maneja el vivero sobre el consumo de agua, temperatura y demás datos ambientales, así como la mortalidad de plántulas por causas hídricas?

Anexo C: Estructura de encuesta aplicada

Anexo C: Estructura de encuesta aplicada

ENCUESTA DE PROYECTO MONITOREO AMBIENTAL AUTOMATIZADO

* Obligatoria

* Este formulario registrará su nombre, escriba su nombre.

1. ¿Está usted de acuerdo que el proceso actual de monitoreo climático del vivero permite obtener información oportuna sobre el estado ambiental de las plantas? *

- ☐ Totalmente en desacuerdo
- ☐ En desacuerdo
- ☐ Ni de acuerdo ni en desacuerdo
- ☐ De acuerdo
- ☐ Totalmente de acuerdo

2. ¿Esta usted de acuerdo que el personal que realiza actividades en el vivero cuenta conocimiento necesario para realizar el monitoreo de las variables climáticas? *

- ☐ Totalmente en desacuerdo
- ☐ En desacuerdo
- ☐ Ni de acuerdo ni en desacuerdo
- ☐ De acuerdo
- ☐ Totalmente de acuerdo

3. ¿Considera usted que el estrés hídrico causado por el excesivo de agua o la falta de esta en las plantas de cacao se presenta debido a la falta de información oportuna sobre las condiciones climáticas? *

- ☐ Totalmente en desacuerdo
- ☐ En desacuerdo
- ☐ Ni de acuerdo ni en desacuerdo
- ☐ De acuerdo
- ☐ Totalmente de acuerdo

4. ¿Con que facilidad puede identificar las enfermedades del cultivo de cacao mediante la observación directa? *
- ☐ Nunca
 - ☐ Rara vez
 - ☐ A veces
 - ☐ Frecuentemente
 - ☐ Siempre
5. ¿Conoce usted que enfermedades pueden propagarse en el cultivo de cacao debido a los cambios climático ? *
- ☐ Si
 - ☐ No
6. ¿Con que frecuencia ha aplicado algún producto insecticida o abono sin conocer si la planta realmente lo necesita? *
- ☐ Nunca
 - ☐ Rara vez
 - ☐ A veces
 - ☐ Frecuentemente
 - ☐ Siempre
7. ¿Cómo evaluaría la eficacia del método actual de monitoreo manual de variables ambientales (temperatura, humedad) para el desarrollo de las plántulas de cacao? *
- ☐ Muy eficaz
 - ☐ Eficaz
 - ☐ Ni eficaz ni ineficaz
 - ☐ Ineficaz
 - ☐ Muy ineficaz

8. ¿De qué manera considera que influye el acceso a información ambiental en tiempo real sobre la toma de decisiones fitosanitarias en el vivero? *
- ☐ Complica la toma de decisiones
 - ☐ No influye
 - ☐ Facilita moderadamente la toma de decisiones
 - ☐ Facilita significativamente la toma de decisiones
9. Desde su experiencia, ¿qué efecto tendría un sistema tecnológico de monitoreo en la tasa de supervivencia de las plántulas de *Theobroma cacao*? *
- ☐ Efecto negativo
 - ☐ Ningún efecto
 - ☐ Efecto positivo moderado
 - ☐ Efecto positivo significativo
10. ¿Qué grado de relevancia considera que tiene la integración de tecnologías dentro de su formación práctica en la universidad? *
- ☐ Irrelevante
 - ☐ Poco relevante
 - ☐ Relevante
 - ☐ Muy relevante
11. ¿Cómo valora la utilidad de recibir alertas automáticas (ej. notificaciones de calor extremo) para la prevención de daños en los cultivos? *
- ☐ Inútil
 - ☐ Poco útil
 - ☐ Útil
 - ☐ Muy útil

- ...
12. ¿Cree usted que el registro manual de datos de riego y clima aumenta el riesgo de cometer errores en la información recolectada? *
- ☐ Muy de acuerdo
- ☐ De acuerdo
- ☐ Neutral
- ☐ En desacuerdo
- ☐ Muy en desacuerdo
13. ¿Considera que el desconocimiento de los rangos óptimos de riego afecta actualmente el desarrollo de los cultivos de cacao? *
- ☐ Muy de acuerdo
- ☐ De acuerdo
- ☐ Neutral
- ☐ En desacuerdo
- ☐ Muy en desacuerdo
14. ¿Cree que la falta de automatización en la captura de datos ambientales facilita la propagación de plagas y enfermedades? *
- ☐ Muy de acuerdo
- ☐ De acuerdo
- ☐ Neutral
- ☐ En desacuerdo
- ☐ Muy en desacuerdo
15. ¿Cree que la baja adopción de tecnologías (como el IoT) se debe principalmente a la falta de capacitación técnica en el sector? *
- ☐ Muy de acuerdo
- ☐ De acuerdo
- ☐ Neutral
- ☐ En desacuerdo
- ☐ Muy en desacuerdo

16. Invierto una cantidad considerable de tiempo pasando datos de campo "a limpio" en formatos digitales para mis reportes. *

- ☐ Muy de acuerdo
- ☐ De acuerdo
- ☐ Neutral
- ☐ En desacuerdo
- ☐ Muy en desacuerdo

17. El canal de comunicación actual para reportar emergencias en las parcelas es inmediato y efectivo. *

- ☐ Muy de acuerdo
- ☐ De acuerdo
- ☐ Neutral
- ☐ En desacuerdo
- ☐ Muy en desacuerdo

18. La falta de un repositorio centralizado de datos genera confusión e inconsistencia en los informes finales míos o de mi grupo. *

- ☐ Muy de acuerdo
- ☐ De acuerdo
- ☐ Neutral
- ☐ En desacuerdo
- ☐ Muy en desacuerdo

19. En el control de las variables ambientales (riego y temperatura), considera que se ejecuta siguiendo estrictamente protocolos técnicos de la carrera. *

- ☐ Totalmente en desacuerdo
- ☐ En desacuerdo
- ☐ Neutral
- ☐ Totalmente de acuerdo

20. Las bitácoras (registro organizado de actividades, avances y observaciones) de registro manual reflejan fielmente las condiciones climáticas observadas durante las jornadas de prácticas. *

- ☐ Totalmente en desacuerdo
- ☐ En desacuerdo
- ☐ Neutral
- ☐ De acuerdo
- ☐ Totalmente de acuerdo

21. Puede identificar con facilidad los signos de estrés térmico o hídrico en las plántulas de cacao mediante la observación. *

- ☐ Totalmente en desacuerdo
- ☐ En desacuerdo
- ☐ Neutral
- ☐ De acuerdo
- ☐ Totalmente de acuerdo

22.Cuál es su opinión sobre la eficiencia y eficacia (sin errores) en el registro manual de los datos ambientales meteorológicos (ej: temperatura, humedad, radiación solar) *

- ☐ Totalmente eficiente
- ☐ Eficiente
- ☐ Neutral
- ☐ Poco eficiente
- ☐ Nada eficiente

23. Considera que las herramientas actuales (ej: regadera, palin o trasplantador manual, manguera, estacas y cuerdas para soporte de plantas, entre otras) del vivero son suficientes para garantizar un desarrollo óptimo de las plántulas. *

- ☐ Totalmente en desacuerdo
- ☐ En desacuerdo
- ☐ Neutral
- ☐ De acuerdo
- ☐ Totalmente de acuerdo

24. ¿Cómo considera la posibilidad de revisar y comparar registros del clima cuando la información se guarda de forma física (ej: libretas, papel, entre otros). *

- ☐ Muy fácil
- ☐ Fácil
- ☐ Neutral
- ☐ Difícil
- ☐ Muy difícil

25. ¿Cómo considera la falta de alertas automáticas para el cuidado de las plántulas cuando no hay personal presente? *

- ☐ Muy adecuada
- ☐ Adecuada
- ☐ Neutral
- ☐ Poco adecuada
- ☐ Nada adecuada

26. ¿Qué método de riego se utiliza actualmente para las plántulas de cacao? *

- ☐ Manual con manguera
- ☐ Aspersión programada
- ☐ Goteo manual
- ☐ Otros

27. ¿Cuánto tiempo al día dedica el personal exclusivamente a la supervisión del riego? *

- ☐ Menos de 1 hora
- ☐ De 1 a 3 horas
- ☐ De 4 a 6 horas
- ☐ Más de 6 horas

28. ¿Qué dispositivo utiliza con más frecuencia durante su jornada laboral? *

- ☐ Teléfono inteligente (Smartphone)
- ☐ Tablet
- ☐ Computadora
- ☐ Ninguno

29. ¿Qué fuentes de información utiliza para tomar decisiones sobre el riego? *

- ☐ Experiencia personal
- ☐ Recomendaciones de proveedores
- ☐ Publicaciones técnicas
- ☐ Otros agricultores
- ☐ Indicaciones del docente supervisor
- ☐ Otras

30. ¿Con qué frecuencia ajusta la cantidad de agua de riego según las condiciones climáticas observadas (temperatura, humedad, lluvia)? *

- ☐ Nunca (siempre riego la misma cantidad)
- ☐ Rara vez (solo en casos extremos)
- ☐ A veces (cuando recuerdo verificar)
- ☐ Frecuentemente (varias veces por semana)
- ☐ Siempre (diariamente según condiciones)

Este contenido no está creado ni respaldado por Microsoft. Los datos que envíe se enviarán al propietario del formulario.



Anexo D: Fotografías

Anexo D: Fotografías



Visualización del campo en el cual se encuentran las plantas del Vivero Lastenia Vera, verificando la evidencia del trabajo manual que se desarrolla para identificar el estrés hídrico de las plantas de cacao.



Realización de la encuesta aplicada a los estudiantes de agropecuaria.



Reunión para la entrevista al encargado del Vivero Lastenia.

Anexo E: Reporte del sistema antiplagio

Anexo E: Reporte del sistema antiplagio

Proyecto de titulación Rodríguez Kevin - Compilatio

ID : 3cba454e00d86008c8df6166f214ae7ddaedf3bb



8%

Textos sospechosos

Nombre del fichero : Proyecto de titulación Rodríguez Kevin - Corrección SEM16 - copia.txt
Tamaño del archivo original : 1,95 MB
Número de palabras : 29.516

Depositante : Kevin Rodríguez
Autor : Kevin Rodríguez
Fecha de depósito : 26 de julio de 2026
Tipo de carga : url_submission
fecha de fin de análisis : 26 de julio de 2026

Resumen (sección 1/2)

Localización de los textos sospechosos en el documento :



Incluido en el porcentaje de textos sospechosos :

Similitudes <1%

Pasajes con similitudes a fuentes encontradas en diferentes colecciones.



Detección de IA 8%

Textos estilísticamente próximos a un texto generado por una IA.

Este índice es un indicador y no una prueba. Comprueba con el autor si domina los conocimientos mencionados en el documento.



Idiomas no reconocidos 1%

Pasajes en los que parte del vocabulario utilizado no forma parte del diccionario de la lengua.

Puede tratarse de un intento del autor de modificar el texto para evitar ser detectado.



No incluido en el porcentaje de textos sospechosos :

Textos entre comillas 1%

Pasajes entre comillas, a menudo indicativos de una cita.



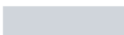
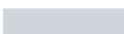
Similitudes

<1%

Pasajes con similitudes a fuentes encontradas en diferentes colecciones.



Fuente con similitudes fortuitas

Nº	Descripciones	Similitudes	Ubicaciones
1	 TESIS HEYDER #422a46  Viene de de mi grupo	<1%	
2	 cia.uagraria.edu.ec cia.uagraria.edu.ec/Archivos/RODRIGUEZ%20PEREA%20BRUCE%20JOAN.p... 	<1%	
3	 FORMATO EXTENSIÓN EL CARMEN - EVALUACIÓN uleam.suplementocica.org/index.php/SuplementoCICA/article/download/... 	<1%	

GLOSARIO

- **Algoritmo:** Conjunto ordenado de operaciones matemáticas y sistemáticas que permite hacer un cálculo algorítmico y hallar la solución a un tipo de problema predictivo.
- **API REST (Representational State Transfer):** Estilo de arquitectura de *software* que utiliza el protocolo HTTP para la comunicación asíncrona entre el cliente (aplicación móvil) y el servidor, permitiendo el intercambio de datos.
- **APK (Android Package Kit):** Formato de archivo ejecutable utilizado por el sistema operativo Android para la distribución, empaquetado e instalación de aplicaciones móviles.
- **Asíncrono:** Proceso informático que se ejecuta de manera independiente al flujo principal del programa, permitiendo que la aplicación no se congele (*lag*) mientras espera una respuesta de red.
- **Backend:** Capa de acceso a datos de un *software* o servidor con la que el usuario final no interactúa directamente; en este proyecto, hace referencia al motor inteligente en Python/FastAPI.
- **Chatbot:** Asistente o programa informático diseñado para simular una conversación fluida con usuarios humanos a través de una interfaz de texto, utilizando modelos de inteligencia artificial.
- **Clasificación de intenciones (Intent Classification):** Tarea principal del procesamiento de lenguaje natural que consiste en identificar algorítmicamente el propósito o la meta detrás del texto ingresado por un usuario.
- **Corpus lingüístico:** Conjunto amplio y estructurado de textos (frases, preguntas y modismos agrícolas) que se utiliza como base de datos para entrenar modelos de procesamiento de lenguaje.

- **Dashboard:** Panel de control visual o interfaz gráfica de usuario en la aplicación móvil que muestra indicadores clave de rendimiento (como el riesgo ambiental) de manera centralizada y dinámica.
- **Dart:** Lenguaje de programación optimizado para el desarrollo de interfaces de usuario rápidas en múltiples plataformas, utilizado como base principal del *framework* Flutter.
- **Endpoint:** Punto final de un canal de comunicación web; es la ruta o URL específica donde un servicio en el *backend* recibe y procesa las peticiones de los clientes.
- **Estrés hídrico:** Condición biológica crítica en la que la planta demanda mayor cantidad de agua de la que tiene disponible en su entorno, afectando severamente su desarrollo y rendimiento fisiológico.
- **FastAPI:** *Framework* web moderno y de alto rendimiento para construir APIs con Python, basado en la tipificación estándar y diseñado para optimizar los tiempos de respuesta.
- **Flutter:** Kit de desarrollo de *software* (SDK) de código abierto creado por Google, utilizado para programar y compilar aplicaciones nativas multiplataforma desde una única base de código.
- **Frontend:** Capa de presentación gráfica de una aplicación, referida a la interfaz con la que interactúa directamente el usuario final a través de su dispositivo móvil.
- **HTTP (Hypertext Transfer Protocol):** Protocolo de transferencia de hipertexto utilizado en la arquitectura cliente-servidor para el intercambio seguro de mensajes y documentos JSON.
- **Inferencia:** En inteligencia artificial, es el proceso de ejecución en el cual un modelo matemático ya entrenado recibe nuevos datos (temperatura, humedad) y genera una predicción instantánea.
- **Interfaz de Usuario (UI):** Espacio gráfico donde ocurren las interacciones entre los seres humanos y el *software*; comprende las pantallas, botones, tipografías y alertas visuales.

- **Internet de las Cosas (IoT):** Red tecnológica de objetos físicos que llevan incorporados sensores e instrumentación con el fin de conectar e intercambiar información climática a través de internet.
- **JSON (JavaScript Object Notation):** Formato de texto ligero, estructurado y legible utilizado como estándar para el empaquetado y envío de datos entre la aplicación Flutter y el servidor Python.
- **Latencia:** Tiempo exacto de demora que tarda en transmitirse un paquete de datos dentro de la infraestructura de red, desde que se envía la consulta hasta que se renderiza la respuesta.
- **Machine Learning (Aprendizaje Automático):** Rama de las ciencias de la computación que dota a los sistemas informáticos de la capacidad de reconocer patrones complejos a partir de datos sin ser programados estáticamente.
- **MinMaxScaler:** Algoritmo de normalización utilizado en modelos de ciencia de datos que transforma las variables numéricas escalándolas a un rango específico (0 a 1) para estabilizar el cálculo del algoritmo.
- **Modismo:** Expresión semántica fija, privativa de una región o lengua, cuyo significado no se deduce de su sintaxis; en este ámbito, abarca el vocabulario coloquial empleado por los agricultores.
- **Procesamiento de Lenguaje Natural (NLP):** Campo híbrido entre la lingüística y la inteligencia artificial que permite a las computadoras leer, comprender, interpretar y procesar el texto humano.
- **Random Forest (Bosque Aleatorio):** Algoritmo predictivo de aprendizaje automático supervisado que opera construyendo múltiples árboles de decisión paralelos para mejorar la precisión en la evaluación del estrés hídrico.
- **Renderizado:** Proceso computacional de generar de manera instantánea una imagen o interfaz visual completa a partir del código fuente procesado en el dispositivo del usuario.

- **Scikit-Learn:** Biblioteca de código abierto especializada en aprendizaje automático para el lenguaje Python, utilizada para el entrenamiento e implementación de algoritmos de clasificación.
- **Servidor:** Sistema informático de alta disponibilidad encargado de procesar la lógica de negocio, proporcionar almacenamiento de bases de datos y responder a las consultas de la aplicación móvil.
- **Sistema Multiplataforma:** *Software* informático diseñado con tecnologías modernas para ejecutarse, compilarse e interoperar en múltiples arquitecturas de *hardware* o sistemas operativos (como Android).
- **Theobroma cacao:** Nomenclatura científica asignada al árbol del cacao, especie de planta perenne endémica de América tropical de cuyas semillas procesadas se produce el chocolate.
- **Timeouts (Tiempo de espera agotado):** Excepción o error de red que ocurre cuando el servidor interrumpe la conexión al no recibir o procesar una respuesta durante un límite de tiempo preestablecido.
- **Uvicorn:** Servidor web rápido de interfaz de puerta de enlace asíncrona (ASGI), empleado en entornos de producción para montar y ejecutar eficientemente las aplicaciones de FastAPI.
- **Vivero:** Instalación agronómica especializada destinada a la germinación, control ambiental y cuidado primario de plantas jóvenes en estado de vulnerabilidad antes de su trasplante.
- **Widget:** En la arquitectura de Flutter, es la unidad o bloque de construcción fundamental de la interfaz; cada elemento visual de la pantalla (botones, contenedores, textos) se define como un *widget*.