



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

ARQUITECTURA EDGE LAKEHOUSE SOBRE LORAWAN PARA LA
GESTIÓN DE DATOS AMBIENTALES METEOROLÓGICOS EN EL
VIVERO LASTENIA VERA ULEAM EL CARMEN

PLUA CHAMORRO ANDERSON STEVEN

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:

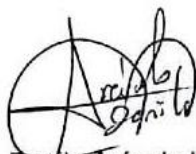
Haber dirigido, revisado y aprobado preliminarmente el Trabajo de Integración Curricular bajo la autoría del estudiante **PLUA CHAMORRO ANDERSON STEVEN**, legalmente matriculado en la carrera de Software, período académico 2026-2027, cumpliendo el total de 384 horas, cuyo tema del proyecto es **"ARQUITECTURA EDGE LAKEHOUSE SOBRE LORAWAN PARA LA GESTIÓN DE DATOS AMBIENTALES METEOROLÓGICOS 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: ARQUITECTURA EDGE LAKEHOUSE SOBRE LORAWAN PARA LA GESTIÓN DE DATOS AMBIENTALES METEOROLÓGICOS EN EL VIVERO LASTENIA VERA ULEAM EL CARMEN

Modalidad: Proyecto integrador

Autor: Plua Chamorro Anderson Steven

Tutor: Ing. Árevalo Hermida Rómulo Danilo

Tribunal de sustentación:

- **Presidente:** A.S Minaya Macias Renelmo Wladimir, Mg

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

- **Miembro:** Ing. López Rodríguez Carlos Vinicio, Mg

Fecha de sustentación:
26 de Agosto de 2026

DECLARACIÓN DE AUTORÍA



La responsabilidad del contenido de este Trabajo de titulación, cuyo tema es: Arquitectura Edge Lakehouse sobre LoRaWAN para la gestión de datos ambientales meteorológicos en el Vivero Lastenia Uleam El Carmen, corresponde exclusivamente a: Plua Chamorro Anderson Steven con CI. 2350052771, y los derechos patrimoniales de la misma corresponden a la Universidad Laica "Eloy Alfaro" de Manabí.

A handwritten signature in blue ink, appearing to read "A. Plua", is written over a horizontal line.

Plua Chamorro Anderson Steven

CI. 2350052771

DEDICATORIA

Este trabajo se lo dedico primeramente a Dios por ayudarme en todo este proceso, desde el inicio hasta el final de toda mi carrera, por ayudarme a crecer y aprender con sabiduría todos los retos que se presentaron y supere a lo largo de mi vida universitaria.

A mi familia, mis papás Jorge Francisco Plua Moran y Gabriela Patricia Chamorro Eches, ya que ellos me apoyaron durante todo este proceso, el sacrificio y el esfuerzo que hicieron para que no me faltara nada, ellos me impulsaron a seguir adelante, a no rendirme, por no dejarme desistir ante las dificultades, demostrándome el amor y el sacrificio que ellos incondicionalmente lo dan, por sus consejos que fueron parte de mi y lo que me estoy formando, enseñándome que todo se logra con esfuerzo y cada esfuerzo tiene su recompensa.

A mi hermana, Liliana Anahi Plua Chamorro, que me apoyo y ah sido testigo de las noches de trabajo y esfuerzo, que siempre eh querido ser un ejemplo para que cumpla todo lo que se proponga y no desista a cumplir sus sueños.

A la persona importante en mi vida que conocí en este proceso universitario, la cual le tengo mucho amor, cariño y aprecio, a Estefany Leonela Murillo Intriago, por apoyarme desde que está en mi vida, a no rendirme y dar lo mejor de mí, por creer en mí y recordarme de lo que soy capaz de hacer y sobre todo por estar para mí incluso cuando las cosas no eran fáciles.

A mi tutor de tesis Ing. Rómulo Danilo Árevalo Hermida por su confianza y la ayuda en este proceso de desarrollo de titulación, por la enseñanza inculcada de los valores y aspectos que debe de tener un profesional.

Anderson Steven Plua Chamorro

AGRADECIMIENTO

Agradezco profundamente a la Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen, por haberme brindado una formación integral a lo largo de la carrera de Ingeniería de Software. A mis docentes y a mi tutor de tesis, Ing. Rómulo Arévalo, por su paciencia, sus valiosos consejos y por compartir su experiencia, la cual fue fundamental para el desarrollo de este trabajo.

Agradezco a mi familia quienes son mi soporte, mi pilar para levantarme una vez más. Gracias por creer en mí y darme la fuerza para afrontar las dificultades que se presentan en la vida como también para culminar esta etapa importante de mi vida.

Finalmente a mis amigos que sin dudar han sido las mejores personas y grandes compañeros de este proceso, a Kevin Rodríguez por ser la amistad la persona que ha estado desde el principio en casi todo juntos; Dereck Castro por estar, ayudarme a entender y superar los límites que tenía pensado; Daniel Pérez por ser un amigo importante que también nos hemos ayudado juntos a no desistir ante las cosas, a esforzarnos más por dar lo mejor de nosotros en cada cosa que hacemos; David Guerrero por ser una gran amistad, que juntos nos hemos apoyado por crecer y creer en nosotros mismos de todo lo que somos capaces de ser y hacer, sin duda estas son las amistades que agradezco y aún quedan muchas más por agradecer y ser parte de quien soy y lo que son y significan para mí.

Anderson Steven Plua Chamorro

ÍNDICE GENERAL

PORTADA.....	I
CERTIFICACIÓN DEL TUTOR.....	III
TRIBUNAL DE SUSTENTACIÓN.....	IV
DECLARACIÓN DE AUTORÍA.....	V
DEDICATORIA	VI
AGRADECIMIENTO	VII
ÍNDICE GENERAL	VIII
ÍNDICE DE TABLAS	XVII
ÍNDICE DE ILUSTRACIONES	XIX
ÍNDICE DE ANEXOS	XXI
RESUMEN	XXII
ABSTRACT.....	XXIII
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	4
1.5	Diagrama causa – efecto del problema	5
1.6	Objetivos	6
1.6.1	Objetivo general.....	6
1.6.2	Objetivos específicos	6
1.7	Justificación.....	6
1.8	Impactos esperados	7
1.8.1	Impacto tecnológico.....	7
1.8.2	Impacto social	7
1.8.3	Impacto ecológico	8
CAPÍTULO II		9
2	MARCO TEÓRICO.....	9
2.1	Antecedentes históricos.....	9
2.2	Antecedentes de investigaciones relacionadas al tema presentado	10
2.3	Definiciones conceptuales.....	11
2.3.1	Arquitectura Edge Lakehouse.....	11
2.3.1.1	Fundamentos de Edge Computing y sus Modelos de Despliegue.....	11
2.3.1.2	Tecnologías de Conectividad IoT de Bajo Consumo LoRaWAN.....	12
2.3.1.3	Conceptos de Componentes y Capas de la Arquitectura Lakehouse.....	12
2.3.1.4	Integración de Edge Computing con Lakehouse: El Patrón Edge Lakehouse	

2.3.1.5	Procesamiento de Datos en el Borde para Streaming Ambiental y Detección de Anomalías	13
2.3.1.6	Gobernanza de Datos y Calidad de la Información en Entornos Lakehouse	14
2.3.1.7	Desafíos de Seguridad y Latencia en las Arquitecturas Distribuidas Edge-Cloud	14
2.3.2	Gestión de datos ambientales.....	15
2.3.2.1	Modelo de Datos para Sensores y Series Temporales Ambientales.....	15
2.3.2.2	Ciclo de Vida de los Datos Ambientales	15
2.3.2.3	Estrategias de Ingesta y Transformación (ETL/ELT) de Datos Heterogéneos	16
2.3.2.4	Calidad, Limpieza y Validación de Datos de Monitoreo Ambiental	16
2.3.2.5	Técnicas de Almacenamiento y Optimización de Consultas para Big Data Ambiental	17
2.3.2.6	Visualización y Reporting de Indicadores Ambientales Clave KPIs	17
2.3.2.7	Reglamento y Requisitos necesarios para la Gobernanza de la Información Ambiental según estándares Locales e Internacionales	18
2.3.3	Metodología de desarrollo Prototipos.....	18
2.4	Conclusiones del marco teórico	19
CAPÍTULO III.....		21
3	MARCO INVESTIGATIVO	21
3.1	Introducción	21
3.2	Tipos de investigación.....	21
3.2.1	Investigación Descriptiva.....	21

3.2.2	Investigación No Experimental.....	22
3.2.3	Investigación Aplicada.....	22
3.2.4	Métodos de investigación	23
3.2.4.1	Método Inductivo-Deductivo	23
3.2.4.2	Método de Análisis-Síntesis	23
3.2.4.3	Método de Observación.....	24
3.3	Fuentes de información de datos	24
3.3.1	Fuentes primarias	24
3.3.1.1	Encuestas	24
3.3.1.2	Entrevista	25
3.4	Estrategia operacional para la recolección de datos.....	25
3.4.1	Población.....	25
3.4.2	Muestra	26
3.4.3	Análisis de las herramientas de recolección de datos a utilizar	27
3.4.3.1	Encuesta.....	27
3.4.3.2	Entrevista	27
3.4.3.3	Estructura de los instrumentos de recolección de datos aplicados	27
3.4.4	Plan de recolección de datos	28
3.5	Análisis y presentación de resultados.....	29
3.5.1	Presentación y descripción de los resultados obtenidos	29
3.5.1.1	Análisis de encuesta dirigida a estudiantes Agropecuaria de la Uleam extensión El Carmen	29

3.5.1.2	Análisis de entrevista dirigida a Coordinador de Vivero Lastenia Vera	32
3.5.2	Informe final del análisis de los datos.....	34
CAPÍTULO IV.....		36
4	MARCO PROPOSITIVO.....	36
4.1	Introducción	36
4.2	Descripción de la propuesta	36
4.3	Determinación de recursos	37
4.3.1	Humanos	37
4.3.2	Tecnológicos	37
4.3.3	Económicos.....	38
4.4	Desarrollo del sistema de trabajo bajo la metodología Scrum.....	39
4.4.1	Descripción del producto	39
4.4.1.1	Propósito del producto.....	39
4.4.1.2	Funcionalidades Clave.....	40
4.4.1.3	Usuarios objetivos	40
4.4.1.4	Criterios de éxito del Producto	41
4.4.2	Requerimientos del sistema	41
4.4.2.1	Historias de Usuario	41
4.4.2.2	Requerimientos funcionales	44
4.4.2.3	Requerimientos no funcionales	45
4.4.3	Diseño del Sistema.....	45

4.4.3.1	Casos de Uso	45
4.4.3.2	Diagramas de Secuencia	47
4.4.3.3	Diagramas de Estado	50
4.4.3.4	Diagrama de Base de Datos.....	52
4.4.4	Arquitectura del sistema	53
4.4.4.1	Arquitectura general	53
4.4.4.2	Tecnologías utilizadas	56
4.4.5	Roles Scrum	57
4.4.6	Definición de Hecho (DoD).....	58
4.4.6.1	Criterios Generales	58
4.4.6.2	Criterios Específicos del Proyecto.....	59
4.4.7	Product Backlog.....	60
4.4.8	Sprint 1: Implementación de la Capa de Percepción y Enlace LoRaWAN.....	61
4.4.8.1	Planificación del Sprint	62
4.4.8.2	Sprint Backlog	63
4.4.8.3	Desarrollo del Sprint.....	63
4.4.8.4	Sprint Review	66
4.4.8.5	Pruebas del Sprint.....	69
4.4.8.6	Incremento y Entregables del Sprint	71
4.4.8.7	Estado actual del Product Backlog	71
4.4.9	Sprint 2: Desarrollo de la Aplicación Móvil y Base de Datos Local (SQLite)..	73

4.4.9.1	Planificación del Sprint	73
4.4.9.2	Sprint Backlog	74
4.4.9.3	Desarrollo del Sprint.....	75
4.4.9.4	Sprint Review	78
4.4.9.5	Pruebas del Sprint.....	79
4.4.9.6	Incremento y Entregables del Sprint	80
4.4.9.7	Estado actual del Product Backlog	80
4.4.10	Sprint 3: Implementación de Alertas Locales (Edge Analytics)	82
4.4.10.1	Planificación del Sprint	82
4.4.10.2	Sprint Backlog	83
4.4.10.3	Desarrollo del Sprint.....	84
4.4.10.4	Sprint Review	86
4.4.10.5	Pruebas del Sprint.....	87
4.4.10.6	Incremento y Entregables del Sprint	89
4.4.10.7	Estado actual del Product Backlog	90
4.4.11	Sprint 4: Despliegue en Campo, Pruebas de Estrés y Empaquetado Final	91
4.4.11.1	Planificación del Sprint	91
4.4.11.2	Sprint Backlog	92
4.4.11.3	Desarrollo del Sprint.....	93
4.4.11.4	Sprint Review	95
4.4.11.5	Pruebas del Sprint.....	96

4.4.11.6	Incremento y Entregables del Sprint	97
4.4.11.7	Estado actual del Product Backlog	98
4.4.12	Resultado Final del Producto	99
4.4.12.1	Funcionalidades implementadas.....	99
4.4.12.2	Evidencia final del sistema	100
4.4.12.3	Conclusiones del desarrollo.....	106
CAPÍTULO V		108
5	EVALUACIÓN DE RESULTADOS	108
5.1	Introducción	108
5.2	Presentación y monitoreo de resultados	108
5.2.1	Planificación de la evaluación	108
5.2.2	Ejecución del monitoreo	110
5.2.2.1	Proceso 1: Alojamiento local de los datos en SQLite	110
5.2.2.2	Proceso 2: Conservación sin conexión y cola de retransmisión.....	113
5.2.2.3	Proceso 3: Consulta del histórico sin conexión	115
5.3	Interpretación objetiva.....	117
CAPÍTULO VI.....		120
6	CONCLUSIONES Y RECOMENDACIONES	120
6.1	Conclusiones	120
6.2	Recomendaciones.....	121
BIBLIOGRAFÍA		123

ANEXOS	127
--------------	-----

ÍNDICE DE TABLAS

Tabla 1: Recolección de Datos	28
Tabla 2: Análisis de encuesta dirigida a estudiantes.....	29
Tabla 3: Análisis de entrevista dirigida a Coordinador	32
Tabla 4: Determinación recursos humanos.....	37
Tabla 5: Determinación de recursos tecnológicos	37
Tabla 6: Determinación de recursos económicos	38
Tabla 7: Tipos de usuarios objetivos	40
Tabla 8: HU1 Monitoreo en tiempo real.....	41
Tabla 9: HU2 Alertas de humedad critica.....	42
Tabla 10: HU3 Reporte de datos históricos	42
Tabla 11: HU4 Gestión de Dispositivos IoT.....	43
Tabla 12: HU5 Bitácora digital de actividades	43
Tabla 13: Tecnologías utilizadas	56
Tabla 14: Roles Srcum.....	57
Tabla 15: Product Backlog.....	60
Tabla 16: Planificación del Sprint 1.....	62
Tabla 17: Sprint Backlog 1.	63
Tabla 18: Pruebas de Caja Negra - Sprint 1.....	69
Tabla 19: Pruebas de Caja Blanca - Sprint 1	70
Tabla 20: Estado actual del Product Backlog - Sprint 1	71

Tabla 21: Planificación del Sprint 2.....	73
Tabla 22: Sprint Backlog 2	74
Tabla 23: Pruebas de Caja Negra - Sprint 2.....	79
Tabla 24: Pruebas de Caja Blanca - Sprint 2	79
Tabla 25: Estado actual del Product Backlog - Sprint 2	80
Tabla 26: Planificación del Sprint 3.....	82
Tabla 27: Sprint Backlog 3.	83
Tabla 28: Pruebas de Caja Negra - Sprint 3.....	87
Tabla 29: Pruebas de Caja Blanca - Sprint 3.	88
Tabla 30: Estado actual del Product Backlog - Sprint 3.	90
Tabla 31: Planificación del Sprint 4.....	91
Tabla 32: Sprint Backlog 4.	92
Tabla 33: Pruebas de Caja Negra - Sprint 4.....	96
Tabla 34: Pruebas de Caja Blanca - Sprint 4.	97
Tabla 35: Estado actual del Product Backlog - Sprint 4.	98
Tabla 36: Planificación de la evaluación del módulo de base de datos móvil.....	108
Tabla 37: Resultados del alojamiento local de datos en SQLite.....	112
Tabla 38: Resultados de la conservación sin conexión y reenvío.....	114
Tabla 39: Resultados de la consulta del histórico sin conexión.....	117

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Diagrama causa - efecto del problema.....	5
Ilustración 2: Diagrama Caso de Uso (Entrada)	46
Ilustración 3: Diagrama Caso de Uso (Proceso)	46
Ilustración 4: Diagrama Caso de Uso (Salida).....	47
Ilustración 5: Diagrama de Secuencia Entada de datos (Captura y Registro)	48
Ilustración 6: Diagrama de Secuencia Procesamiento (Edge Computing y Sincronización) ..	49
Ilustración 7: Diagrama de Secuencia Salida (Visualización y Reportes).....	50
Ilustración 8: Diagrama de Estado Paquete de datos Ambientales.....	51
Ilustración 9: Diagrama de Estado Nodo Sensor IoT	52
Ilustración 10: Diagrama de base de datos SQLite.....	53
Ilustración 11: Arquitectura General del Sistema (Arquitectura Edge Lakehouse sobre LoRaWAN).....	55
Ilustración 12: Ensamblaje físico sensor.....	64
Ilustración 13: Monitor Serial de PlatformIO en Visual Studio Code.....	65
Ilustración 14: Función preparePayload	66
Ilustración 15: Respuesta del Monitor Serial de PlatformIO en Visual Studio Code.....	67
Ilustración 16: Base de datos Local SQLite.....	76
Ilustración 17: Recepción por Bluetooth y alojamiento del paquete en la base local.....	77
Ilustración 18: Funcionalidad de alertas por umbral.....	84
Ilustración 19: Función lógica aviso por umbral	85
Ilustración 20: Lógica DDL del mecanismo Store-and-Forward en SQLite	93

Ilustración 21: Generación de la versión Release (.APK)	94
Ilustración 22: Despliegue in situ	95
Ilustración 23: Estado de conexión activo aplicación.....	101
Ilustración 24: Registros alojados bd móvil.....	102
Ilustración 25: Comportamiento cola de retransmisión	103
Ilustración 26: Consulta valores max y min offline.....	104
Ilustración 27: Aviso de alerta fuera del umbral.....	105
Ilustración 28: APK verificado en campo.....	106
Ilustración 29: Registros ambientales alojados en la base de datos local	111
Ilustración 30: Cola de retransmisión antes y después de restablecer la conexión.....	114
Ilustración 31: Consulta del histórico ambiental sin conexión a internet	116

ÍNDICE DE ANEXOS

Anexo A: Asignación de tutor	127
Anexo B: Evidencia de aplicación de encuestas y entrevistas.....	128
Anexo C: Fotografía	129
Anexo D: Estructura de la entrevista aplicada.....	130
Anexo E: Estructura de la encuesta aplicada	131
Anexo F:Reporte del sistema antiplagio	139

RESUMEN

El presente proyecto de titulación tiene como objetivo general aplicar la arquitectura Edge Lakehouse basada en LoRaWAN y emplearla para gestionar datos referentes a las condiciones meteorológicas en el vivero Lastenia Vera de la Uleam extensión El Carmen. El problema observado es que, de manera habitual, los parámetros ambientales en el vivero se registraban a mano y ocasionalmente, lo cual resultaba en datos incompletos, errores frecuentes en la recolección de la información y la imposibilidad de asegurar su integridad. Asimismo, la inestabilidad de la conexión a Internet, que es propia de las áreas rurales, empeora este escenario y hace más complicado conseguir datos históricos confiables, los cuales son esenciales para tomar decisiones agronómicas. En la investigación intervinieron 88 alumnos de la carrera de agropecuaria, quienes estaban involucrados directamente en el manejo del vivero, además del coordinador responsable de este. Se eligieron 72 estudiantes para la muestra utilizando un procedimiento de muestreo probabilístico con un margen de error del 5 % y una confianza del 95 %. Se emplearon como métodos de recolección de datos una encuesta (que se llevó a cabo con los estudiantes a través de un cuestionario compuesto por 30 preguntas) y una entrevista con el coordinador del vivero Lastenia Vera. Se elaboró una aplicación móvil con una base de datos relacional incorporada en cuatro iteraciones, como solución desarrollada dentro del enfoque Scrum. Esta aplicación, por medio de Bluetooth clásico, recibe paquetes de datos telemétricos y, utilizando un sistema de "almacenamiento y reenvío", guarda estos datos localmente. De esta forma, los usuarios pueden acceder al historial de datos ambientales sin necesidad de estar conectados a la red. Los resultados de las pruebas revelaron que la integridad del almacenamiento de datos llegó al 100 %: no se dejaron fuera ni se duplicaron los 205 paquetes de datos enviados. Asimismo, se restauraron todos los 150 paquetes de datos que llegaron durante las interrupciones de la red, lo cual posibilitó llevar a cabo una consulta exhaustiva de los datos históricos aun cuando el aparato no estaba conectado a Internet.

ABSTRACT

The overall objective of this thesis project is to implement the LoRaWAN-based Edge Lakehouse architecture and use it to manage weather data at the Lastenia Vera nursery at the ULEAM El Carmen extension. The problem observed is that environmental parameters at the nursery were typically recorded manually and only occasionally, resulting in incomplete data, frequent errors in data collection, and an inability to ensure data integrity. The inconsistent internet connectivity common in rural areas further complicates this issue, making it difficult to maintain the reliable historical records necessary for agronomic planning. The research involved the nursery coordinator along with eighty-eight agricultural science students responsible for its management. A probabilistic sample of seventy-two students was selected from this group, utilizing a 95% confidence level and a 5% margin of error. The data collection methods used were a survey (conducted with the students using a questionnaire consisting of 30 questions) and an interview with the nursery coordinator, Lastenia Vera. A mobile application with a built-in relational database was developed over four iterations, as a solution created using the Scrum methodology. This application receives telemetry data packets via classic Bluetooth and, using a “store-and-forward” system, stores this data locally. This allows users to access the environmental data history without needing to be connected to the network. Test results revealed that data storage integrity reached 100%: none of the 205 data packets sent were omitted or duplicated. Furthermore, all 150 data packets received during network outages were restored, enabling a comprehensive query of historical data even when the device was not connected to the Internet.

CAPÍTULO I

1 INTRODUCCIÓN

1.1 Introducción

El cultivo de cacao es una fuente fundamental en la economía de Ecuador, esto significa que la fase de los viveros es crítica para el crecimiento y producción de las plántulas de cacao. actualmente la gestión de datos ambientales meteorológicos (VD) en el vivero Lastenia de la extensión Uleam El Carmen, se basa tradicionalmente de manera manual. Esta metodología trae consigo errores, latencia de información e impide la toma de decisiones inmediata, lo cual resulta en una administración ineficiente de los recursos y una vulnerabilidad elevada para las plántulas.

Para contrarrestar y dar soluciones, se propone implementar una Arquitectura Edge Lakehouse sobre LoRaWAN (VI). Esta solución plantea la integración de tres tecnologías clave: LoRaWAN, para la recolección de datos de los sensores IoT; Edge Computing, para el procesamiento de datos en el sitio y la generación de alertas en tiempo real; y un Lakehouse para un almacenamiento estructurado y análisis de tendencias facilitando la consulta del historial ambiental y la toma de decisiones agronómicas a partir de los datos conservados en el dispositivo.

En combinación de estas dos variables se busca demostrar como este sistema puede ayudar a mejorar los procesos agrícolas, en la germinación y riesgo de los cultivos ante las variables climáticas extremas. Se busca transformar tecnológicamente esta gestión de datos. El objetivo principal se basa en implementar una arquitectura Edge Lakehouse sobre LoRaWAN para la gestión eficiente de datos ambientales meteorológicos en el Vivero Lastenia Uleam El Carmen.

La comprobación en la metodología se realiza utilizando modelos prototipos debido a que es adecuado para los sistemas que tienen interacción con el usuario. Permitiendo que el usuario pruebe la interfaz y las funcionalidades necesarias en una etapa temprana gracias al trabajo directo con los encargados del vivero. La aplicación de esta metodología garantiza que se use óptimamente el sistema, también agrega valor a la creación de prototipos similares de tecnologías computación en el borde e Internet de las cosas lo cual es de gran apoyo para la toma de decisiones agronómicas a los productores.

El presente trabajo constituye una parte fundamental del proyecto integrador “Monitoreo Ambiental Automatizado” en el vivero Lastenia Vera por lo cual es importante desglosar este enorme proyecto. El título y la documentación describe una arquitectura técnica general necesaria para que este ecosistema funcione (conectividad LoRaWAN, servidores, Edge Lakehouse); sin embargo, el alcance de este documento y su contribución técnica directa se limitan estrictamente al módulo 3, “Bases de datos móviles y alojamiento de datos”. Otros miembros del grupo del proyecto integrador se encargan de desarrollar funciones como el diseño de hardware, los protocolos de cifrado y la base de datos centralizada en la nube. Por lo tanto, la metodología y el enfoque técnico de esta propuesta se centran en el procesamiento de datos, la gestión local mediante una base de datos relacional (SQLite) y la garantía de la capacidad de funcionar en modo sin conexión.

1.2 Presentación del tema

En el sector agrícola la falta de tecnologías modernas, para el monitoreo ambiental en tiempo real, la fragmentación de almacenamiento y limitada conectividad, son aspectos que afectan a la calidad y eficiencia en la toma de decisiones agrónomas. La Arquitectura Edge Lakehouse sobre LoRaWAN para la gestión de datos ambientales meteorológicos es de suma relevancia, por su capacidad de captura y procesamiento de datos. Esta solución tecnológica mejora la sustentabilidad, maximizando la productividad y reduciendo costos operativos, lo cual es fundamental en la agricultura inteligente.

1.3 Ubicación y contextualización de la problemática

La prestigiosa Universidad Laica Eloy Alfaro de Manabí (Uleam) es una institución de educación superior pública, la cual se fundó en 1985, su matriz principal se encuentra en la ciudad de Manta, provincia de Manabí, Ecuador. Esta institución cuenta con extensiones, entre ellas la Extensión El Carmen, ubicada en el cantón El Carmen, dentro de la misma provincia. Las instalaciones del campus “Lastenia Vera” de esta extensión, se ubican en un área experimental recuperada del antiguo colegio “Lastenia Isabel Vera Pinargote”. Actualmente es un complejo moderno académico, con aulas de clase y áreas verdes dedicadas para el vivero de plantas de cacao.

Para la implementación del proyecto de estación meteorológica, se eligió el campus “Lastenia Vera” debido a sus condiciones climáticas excelentes y a su ubicación alejada del centro urbano de la ciudad, factor fundamental que favorece al crecimiento de cultivos de la región. En esta área rural, la limitada conectividad a internet impide o limita la automatización de procesos y sistemas para el monitoreo ambiental, lo que justifica la implementación de tecnologías adaptadas a estas condiciones que sea de gran apoyo a la toma de decisiones agronómicas.

1.4 Planteamiento del problema

1.4.1 Problematicación

En el vivero “Lastenia Vera” de la Uleam El Carmen los principales procesos que se realizan son: 1) selección de semillas óptimas, 2) germinación de la semillas de cacao, 3) preparación de sustrato y envasado con una mezcla de tierra y cascarilla de arroz en fundas de polietileno, 4) repique o trasplante de las semillas germinadas a las fundas con la mezcla de sustrato , 5) cuidados en vivero de riego, sombra, fertilización, control fitosanitario, aclareo y selección de plantas, 6) rustificación y trasplante de plantas a campo definitivo.

En la selección de semillas, germinación y preparación de sustrato comparten el problema de la dependencia de registros manuales y la falta de monitoreo en tiempo real para el acondicionamiento de las semillas, la incubación e hidratación. En el repique o trasplante presenta problemas durante la operación ya que queda expuesta a las fluctuaciones de la temperatura y viento sin una alerta temprana, sin un sistema que informe automáticamente las condiciones que se presentan durante el trasplante, reduce la supervivencia de las mismas.

En los cuidados en el vivero, riego, sombra, entre otros cuidados más, el problema de la falta de monitoreo de humedad del suelo y radiación solar en tiempo real impide que se optimice el riego y la fertilización, mientras que el control de plagas se basa en inspecciones visuales activas sin registros y análisis de las variables climáticas aumentan el riesgo de estos brotes para llegar finalmente a la última etapa debido a la falta del historial de los datos del ambiente y crecimiento de las plantas resulta en una difícil anticipación de adecuación del entorno lo que produce que las plantas mueran fácilmente.

Las siguientes interrogantes analizan la problemática: ¿Cómo mejorar el seguimiento de las plantas en el vivero “Lastenia Vera”? ¿Cómo la digitalización de los registros facilita las etapas

de desarrollo de las semillas de cacao y la gestión de recursos? ¿Cómo la ayuda de alertas automáticas ayuda a prevenir que existan pérdidas en el vivero a causa del cambio de temperatura o la reproducción de enfermedades?

1.4.2 Génesis del problema

El Génesis de la problemática surgió desde el impulso del proyecto de siembra de un millón de plantas de cacao, por parte de la Uleam el 05 de junio de 2024, liderado por el Dr. Marcos Zambrano, rector de la Uleam. El cual tiene como objetivo reforestar zonas desérticas y promover la expansión del cultivo de cacao. El origen del problema se remonta a la administración de datos ambientales, gestión de recursos para el cuidado de las semillas para el desarrollo de las semillas, ya que la cantidad es muy grande para abarcar con procesos operativos manuales.

El problema no estaba presente antes de estos procesos operativos en el campus, se originó con la creación del campus “Lastenia Vera” y la disposición de las áreas verdes para la implementación del proyecto de un millón de plantas de cacao, ya que la administración y proceso de recolección de datos ambientales se hacen de manera manual, lo que dificulta la eficiencia y optimización de recurso para el cuidado de las plántulas.

1.4.3 Estado actual del problema

Actualmente, las mediciones se realizan de manera manual y se registran en bitácoras, lo que genera errores, omisiones y falta de continuidad en los datos. Todo esto impiden un monitoreo constante y preciso de las condiciones microclimáticas. Lo cual dificulta la gestión del vivero limitando la capacidad de mantener un ambiente estable y optimizado para el crecimiento de las plántulas.

Las causas de esta problemática, se vinculan con dependencia de la observación visual y la experiencia empírica que limita el respaldo cuantitativo que se base en evidencia científica y parámetros que sean medibles. La ausencia de alertas tempranas preventivas que retrasan las respuestas ante condiciones adversas climáticas. Por último, el registro fragmentado de la información climática que elimina la posibilidad de realizar análisis históricos que ayuden en la toma de decisiones agronómicas.

Como resultado, se presentan los siguientes efectos, tales como los retrasos en los procesos productivos que presentan retrasos en los cronogramas afectando la planificación. El uso ineficiente de los recursos hídricos y nutricionales, resulta en un incremento de los costos operativos sin que se garantice las condiciones óptimas necesarias para el desarrollo de las plántulas. La toma de decisiones tardías ante los cambios climáticos genera estrés abiótico en las plántulas, lo cual incrementa la tasa de mortalidad de las mismas, también afecta en la calidad fitosanitaria. Estos efectos disminuyen la productividad generada del vivero, generando pérdidas significativas en la producción.

1.5 Diagrama causa – efecto del problema

Ilustración 1: Diagrama causa - efecto del problema



1.6 Objetivos

1.6.1 Objetivo general

Implementar arquitectura Edge Lakehouse sobre LoRaWAN para la gestión de datos ambientales meteorológicos en el Vivero Lastenia Uleam El Carmen.

1.6.2 Objetivos específicos

- Determinar las restricciones presentes en la administración y almacenamiento de los datos meteorológicos ambientales en el Vivero Lastenia, mostrando los impactos que tiene la falta de una arquitectura de datos automatizada y unificada.
- Fundamentar de manera técnica y teórica el uso y el origen de la tecnología LoRaWAN y la arquitectura Edge Lakehouse, analizando el cómo se aplican estas tecnologías en sistemas de monitoreo ambiental en entornos controlados como los viveros.
- Definir los requerimientos y las necesidades tecnológicas de información para el análisis, a través de los procedimientos que se lleva a cabo en el manejo y registro de las variables ambientales que se presentan en el Vivero Lastenia Vera.
- Desarrollar y establecer un sistema que use tecnología LoRaWAN y se base en la arquitectura Edge Lakehouse, con el objetivo de centralizar el almacenamiento de datos y organizar los datos ambientales del vivero.
- Verificar mediante pruebas que el sistema sea funcional y apropiado para garantizar la exactitud y la fiabilidad de la transferencia de datos, el almacenamiento íntegro y la capacidad de respuesta.

1.7 Justificación

La implementación de una Arquitectura EdgeLakehouse sobre LoRaWAN proporcionará un monitoreo ambiental automatizado que, en tiempo real, recopilará las variables ambientales, evitando errores como el registro manual y optimizando la gestión de los recursos hídricos y nutricionales. Este sistema permitirá generar bases de datos históricas para el análisis descriptivo y prescriptivo, reduciendo la mortalidad de las plántulas mediante alertas y mejorando el tiempo de la producción del vivero. Además, la información se podrá acceder a la información de forma estructurada desde dispositivos móviles, facilitando la toma de decisiones agronómicas.

Este proyecto beneficiará directamente a comunidad de universitarios de la Uleam El Carmen, incluyendo a los estudiantes investigadores y docentes ya que podrán utilizar esta plataforma para proyecto e investigaciones aplicadas. Los productores de cacao de la región se verán beneficiados al acceder a plantaciones con mayor control para la calidad y uniformidad de sus producciones, como también instituciones agrícolas que repliquen este modelo, extendiendo los beneficios a toda la comunidad fortaleciendo la productividad en esta área.

Este sistema transformará las metodologías de enseñanza e investigación dentro del área de la agricultura de precisión, permitiendo así el desarrollo en el análisis de datos y tecnologías emergentes. Para los productores, proporcionará plantaciones más viables, reduciendo las pérdidas económicas y mejorando la productividad. En el área científica proporcionará conocimiento sobre la implementación de tecnologías IoT y Edge Computing en entornos agrícolas, posicionando a la Uleam como centro de referencia de innovación tecnológica agropecuaria, facilitando futuros proyectos investigativos y desarrollo tecnológico.

1.8 Impactos esperados

1.8.1 Impacto tecnológico

Este sistema tecnológico integrado con IoT, Edge Computing y LoRaWAN, revolucionará en el sector agrícola, ya que facilitará la transferencia de conocimiento tecnológico desde el ámbito académico hacia el ámbito productivo. La implementación de este sistema capacitará tanto a las estudiantes, investigadores y productores en el manejo de sensores inteligentes para el análisis de datos en tiempo real, esto permitirá migrar las prácticas tradicionales de la agricultura a sistemas automatizados. Promoviendo así una cultura de innovación que impulsará en este contexto tecnológico agrícolas sostenibles.

1.8.2 Impacto social

Beneficiará a los productores de plantaciones de cacao, no solamente incrementará su productividad y rentabilidad económica, también mejorará las condiciones socioeconómicas de sus familias en lugares rurales. Los estudiantes y docentes de la Uleam podrán acceder a una formación tecnológica avanzada. Las comunidades se beneficiarán de la economía generado por la mejora en la producción de cacao, creando empleo para comercialización y

procesamiento. Este proyecto contribuirá a la sostenibilidad ambiental al optimizar los recursos, promoviendo prácticas responsables ante el cambio climático en las producciones.

1.8.3 Impacto ecológico

Contribuirá significativamente a la conservación ambiental mediante la optimización de recursos naturales como el agua y nutricionales, reduciendo el desperdicio de agua en este sector evitando la sobre fertilización que afecta los suelos y las fuentes de agua de la zona. El monitoreo preciso permitirá aplicar tratamientos fitosanitarios únicamente cuando sea necesario, disminuyendo el riesgo de enfermedades a las plantaciones disminuyendo el uso de agroquímicos y minimizando el impacto ecológico que genera su uso. Esto contribuirá a una producción sostenible y eficiente mitigando el impacto ambiental de la región.

CAPÍTULO II

2 MARCO TEÓRICO

2.1 Antecedentes históricos

El desarrollo del Edge Computing es una respuesta directa a las limitaciones de retraso que imponen las arquitecturas centralizadas en la nube. Con la rápida proliferación de los dispositivos y sensores IoT, los datos se generan a una velocidad y a una escala sin precedentes, lo que hace que la transmisión continua de datos a centros de datos remotos sea inviable para aplicaciones sensibles al tiempo (Xu y Shi, 2025).

La arquitectura Lakehouse se creó con el objetivo de resolver los paradigmas históricos de almacenamiento de Big Data: el Data Warehouse y el Data Lake. Los Data Warehouses ofrecían una estructura y transmisión costosa como ACID, estaban limitados a manejar datos estructurados. Los Data Lakes unió dos mundos como el almacenamiento de bajo costo del Data Lake y añadiendo también una capa de gestión de metadatos para incluir la confiabilidad y el gobierno de un Data Warehouses. Como resultado dio una plataforma unificada para el análisis y Machine Learning sobre el mismo conjunto de datos (Thalpati, 2024).

A mediados del siglo XX surgió la gestión medioambiental moderna como resultado de una concientización sobre las limitaciones de recursos naturales de la Tierra. Como ejemplo se tiene el informe que planteo un punto de no retorno esto gracias a las predicciones del Club de Roma en 1972 donde los escenarios futuros se basan en el consumo de los recursos. Dichas predicciones generaron una urgencia en la necesidad de tener los datos del medio ambiente reales lo cual sería de gran ayuda a las organizaciones y gobiernos para planificar el impacto ambiental que tienen sus acciones (Santana Moncayo y Aguilera Peña, 2017).

Según D'Angelo Gagnetten et al. (2022), la gestión de datos medioambientales se logra con ayuda de herramientas tecnológicas especiales en los sistemas de gestión, como la gran importancia para las organizaciones que tienen los sistemas de información geográfica (SIG). La cual no solo permite guardar grandes cantidades de información, sino también permite el análisis de distribución espacial lo cual la confiere en uno de los indicadores más claros y reales.

2.2 Antecedentes de investigaciones relacionadas al tema presentado

Trabajo 1: Sistema de información para el monitoreo de la contaminación ambiental de la ciudad de Puyo de acuerdo con las normativas del ministerio del ambiente.

Este proyecto fue desarrollado en la ciudad de Puyo, Ecuador por Becerra Arévalo y Romero Ortiz (2021), donde se implementó un sistema de monitoreo de contaminación atmosférica mediante sensores Arduino MEGA, software web (Apache, PHP, HTML5, MySQL) y una plataforma en línea que permitió a los empleados del Ministerio de Medio Ambiente visualizar datos de calidad del aire en tiempo real, cumpliendo con normativas vigentes y optimizando la gestión ambiental municipal.

Trabajo 2: Creación de prototipo de IoT para el monitoreo ambiental y radiación solar utilizando hardware accesible y alimentación de energía solar.

La investigación de García Álvarez y López Lozano (2024), se desarrolla dentro del ámbito académico ecuatoriano, donde se enfoca en el diseño de un prototipo IoT. Con el objetivo de que este trabajo sea realizado con componentes de bajo costo (económico), para hacer posible un monitoreo ambiental de manera remota y en tiempo real de las variables ambientales. El principal aporte fue la integración funcional de una placa Raspberry Pi 4 con módulos BME280, lo que permitió realizar una comunicación telemétrica hacia las plataformas en la nube apoyándose en servicios orientado al Internet de las cosas, en este caso ThinkSpeak.

Trabajo 3: Arquitectura inteligente edge computing para entornos IoT.

La siguiente investigación doctoral de Sittón Candanedo (2023), la cual fue defendida en la Universidad de Salamanca (España), en la que se planteó alternativas orientadas a administrar la carga masiva de información proveniente de las redes de dispositivos de internet de las cosas. El objetivo de la investigación fue principalmente diseñar una Global Edge Computing Architecture la cual sea capaz de implementar la capacidad de Edge Computing dentro de un ecosistema IoT. Esta investigación dio como resultado las bases de conceptos y de operación de este modelo planteado, como también mediante entornos reales se demostró la eficiencia para la optimización de los procesos directamente en los límites de la red.

Los trabajos de tesis se relacionan con el trabajo al abordar la gestión de datos ambientales mediante IoT y también arquitecturas distribuidas, en las cuales se destaca el uso de Edge Computing para el monitoreo ambiental. La diferencia con esta investigación se basa en la integración de una arquitectura Edge Lakehouse, el cual combina el procesamiento local en tiempo real del Edge con escalabilidad de gobernanza del Lakehouse. Esto permitirá una gestión de datos ambientales meteorológicos eficiente y encaminado a la analítica avanzada, superando limitaciones de enfoque en la nube o servidores tradicionales.

2.3 Definiciones conceptuales

2.3.1 Arquitectura Edge Lakehouse

Esta arquitectura es la combinación perfecta en la flexibilidad que tiene y la escalabilidad debido al almacenamiento Lakehouse con las grandes ventajas del Edge Computing. Dentro de esta arquitectura, los datos se procesan parcialmente cerca de la fuente (por ejemplo, sensores IoT) y luego se aprovechan las capacidades del almacenamiento lakehouse para el almacenamiento o la gestión de datos. Esto permite un análisis de datos, un aprendizaje automático y una gestión de datos eficientes, al tiempo que se reduce la latencia y se minimiza el consumo de ancho de banda. (Hayat et al., 2024, pág. 24)

2.3.1.1 Fundamentos de Edge Computing y sus Modelos de Despliegue

El procesamiento en el lugar del origen es fundamental en la computación el borde, la cual disminuye la latencia y optimiza sistemas no centralizados. Integra TI, computación en la nube y telecomunicaciones, lo cual la convierte en una arquitectura donde cada dispositivo es independiente, donde se filtran y transforman los datos antes de enviar a los servidores centrales, con el objetivo de generar respuestas inmediatas, disminuir el ancho de banda y realizar un procesamiento local, convirtiendo al Edge Computing en un elemento esencial en la implementación de Internet de las cosas. (Lea, 2020, págs. 310-313)

Los modelos de implementación de Edge Computing se clasifican según el lugar donde se procesan los datos. Una arquitectura distribuida jerárquica puede gestionar enormes flujos de datos y coordinar tareas de procesamiento que debe ocurrir localmente. Este diseño segmentado es crucial para resolver problemas de rendimiento y mantener la resiliencia operativa en sistemas distribuidos. (Menon, 2022, págs. 40-42)

2.3.1.2 Tecnologías de Conectividad IoT de Bajo Consumo LoRaWAN

Los sistemas de monitoreo ambiental tienen una importancia fundamental para la tecnología de largo alcance de baja potencias (LPWAN), como LoRaWAN. En estos sistemas los dispositivos generalmente requieren de alimentación autónoma por batería en lugares apartados. Esta red está optimizada para enviar de manera periódica paquetes pequeños de datos, lo que es fundamental para extender la duración de los sensores y ofrecer una cobertura eficiente en extensas zonas geográficas, como los viveros. (Silva Torres y Coello Hurtado, 2020, págs. 30-32)

La conexión de sensores medioambientales requiere protocolos que minimicen el consumo de energía y garanticen una comunicación fiable. La monitorización en tiempo real de las variables medioambientales pone de relieve la necesidad de sensores altamente eficientes y de bajo consumo. El uso de sistemas de hardware como Arduino junto con sensores permite la recopilación precisa de datos medioambientales in situ. (Oña Salazar y Vaca Mármol, 2015, págs. 45-47)

2.3.1.3 Conceptos de Componentes y Capas de la Arquitectura Lakehouse

La arquitectura Lakehouse surgió como un paradigma moderno destinado a combinar las características funcionales de los lagos de datos y los almacenes de datos. Su objetivo principal es crear una plataforma unificada que admita la escalabilidad y la flexibilidad en el almacenamiento de datos sin procesar, al tiempo que incorpora atributos de calidad de los datos, la aplicación de esquemas y las características de procesamiento transaccional (ACID) de un almacén de datos. (Menon, 2022, págs. 23-24)

Se compone de diferentes capas importantes como la de almacenamiento de bajo costo y escalable y otra capa de gestión de grupos de datos (metadatos). El formato que utiliza la última capa es en forma de tabla abierta (como por ejemplo Delta Lake o Apache Hudi) la cual introduce una capa de abstracción de datos donde admite archivos de datos no estructurados provenientes directos de definiciones y operaciones ACID. (Thalpati, 2024, págs. 25-27)

2.3.1.4 Integración de Edge Computing con Lakehouse: El Patrón Edge Lakehouse

El patrón Edge Lakehouse se define como una extensión natural de la arquitectura que aporta capacidades de almacenamiento y procesamiento de datos a los puntos de recopilación de datos (por ejemplo, puertas de enlace). Este modelo es fundamental en los sistemas de supervisión, ya que permite recopilar y preprocesar localmente los datos LoRaWAN antes de la sincronización central. (Lea, 2020, págs. 308-309)

El Gateway con la implementación de la arquitectura Edge Lakehouse, actúa como almacenamiento pequeño distribuido. Lo cual nos asegura la confiabilidad de la transmisión y la caridad de lo datos recolectados aun en situaciones inestables en que la red de internet se caiga. Por consecuencia los datos se envían al almacenamiento centralizado una vez haya sido verificado y organizado, así optimizando la eficacia del ancho de banda y asegurando que los datos sean íntegros. (Silva Torres y Coello Hurtado, 2020, págs. 80-82)

2.3.1.5 Procesamiento de Datos en el Borde para Streaming Ambiental y Detección de Anomalías

El procesamiento de datos de flujo en el borde es esencial para la gestión eficaz de la información medioambiental de alto rendimiento. Realizar cálculos en el borde reduce la necesidad de enviar todo el volumen de los datos sin procesar a la nube. En su lugar, las puertas de enlace ejecutan algoritmos sencillos antes de transmitir, filtrar, agregar o comprimir los datos. (Lea, 2020, págs. 350-352)

Una ventaja directa del procesamiento local es la posibilidad de identificar anomalías en los límites desde el inicio. Esto, por ejemplo, comprende la habilidad de identificar picos altos de temperatura que exceden los límites anticipados. Esta facultad posibilita que se tomen medidas de corrección o alertas en tiempo real, sin importar los retrasos en la conexión a la red del centro de datos, lo cual optimiza la capacidad de respuesta del sistema. (Menon, 2022, págs. 100-102)

2.3.1.6 Gobernanza de Datos y Calidad de la Información en Entornos Lakehouse

En la arquitectura Lakehouse, la gobernanza de datos es de suma importancia. Esto se debe a que debe garantizar que tanto los datos estructurados como los no estructurados cumplan con las políticas de seguridad y cumplimiento. Este proceso tiene como objetivo crear metadatos, trazabilidad de datos y mecanismos de auditoría para garantizar la trazabilidad y la fiabilidad a lo largo de toda la cadena de procesamiento de la información meteorológica, desde los sensores hasta el análisis final. (Thalpati, 2024, págs. 34-36)

Lakehouse garantiza la calidad de los datos gracias a su arquitectura multinivel. Los datos sin procesar procedentes del periférico se someten a una transformación gradual en cada nivel, con reglas de validación, limpieza de datos y estandarización de esquemas aplicadas a lo largo de todo el proceso. Esta estrategia garantiza que los usuarios de análisis empresarial y aprendizaje automático solo tengan acceso a conjuntos de datos de alta calidad. (Oña Salazar y Vaca Mármol, 2015, págs. 78-80)

2.3.1.7 Desafíos de Seguridad y Latencia en las Arquitecturas Distribuidas Edge-Cloud

Existen grandes desafíos de seguridad en la computación en la nube esto debido a la gran superficie de ataque que tiene, donde se necesita una protección de muchas capas como medida de seguridad que abarque todo el espacio dando una protección física de los dispositivos hasta internamente en la transmisión y almacenamiento de datos cifrados. Donde la autenticación de los puntos de conexión a la red y la gestión de identidad de dispositivos es de suma importancia. (Lea, 2020, págs. 400-403)

En las redes LPWAN como LoRaWAN el factor de latencia es crítico debido a que el ancho de banda y los ciclos de trabajo ponen restricciones de tiempo, generando que la distinción de las funciones importantes que deben realizarse en el borde se convierta en un reto, como también las que pueden ser transmitidas a la nube como las funciones de alta latencia en el almacenamiento de datos históricos. (Silva Torres y Coello Hurtado, 2020, págs. 55-57)

2.3.2 Gestión de datos ambientales.

El proceso de recopilar, guardar, verificar, procesar y analizar la información de las condiciones del clima se define como gestión de datos ambientales, el cual tiene como objetivo obtener información real y confiable lo que ocasiona tomar decisiones informadas, cumpliendo con normas y elaborar informes sobre la sostenibilidad como también organizar protección y analizar posibles riesgos ((EPA), 2021).

2.3.2.1 Modelo de Datos para Sensores y Series Temporales Ambientales

El modelo de datos desempeña un papel extremadamente importante en la recopilación de información estructurada procedente de sensores meteorológicos, en la que el «tiempo» constituye la unidad de medida principal. Los datos deben organizarse según una estructura específica, de modo que incluyan un identificador único del sensor, la ubicación geográfica y, lo más importante, la marca de tiempo de la medición. Un modelo de datos sólido garantiza una indexación y una búsqueda eficientes de los datos de series temporales. (Mudelsee, 2020, págs. 30-32)

Los datos medioambientales se consideran, en esencia, una serie temporal, y cada registro está formado por un par de puntos de datos asociados a variables específicas (tiempo, valor numérico). Este modelo debe permitir la incorporación continua de datos y garantizar la optimización de las consultas en distintos intervalos de fechas. Esta estructura es de vital importancia para analizar tendencias pasadas e identificar patrones estacionales. (Shen y North, 2023, págs. 50-52)

2.3.2.2 Ciclo de Vida de los Datos Ambientales

El ciclo de vida de los datos medioambientales comienza con la recopilación de datos mediante sensores instalados en dispositivos periféricos. En esta fase inicial, pueden producirse errores de transmisión y fallos en los equipos. A continuación, los datos brutos pasan a la fase de procesamiento, donde se limpian, verifican y calibran de acuerdo con los protocolos de monitorización establecidos. (Burt, 2024, págs. 140-142)

La etapa final consiste en un procedimiento de almacenamiento, durante el cual los datos procesados se almacenan en un esquema al que se puede acceder para su posterior tratamiento. En esta etapa, es necesario garantizar la trazabilidad de las fuentes de datos. Es decir, para cada valor de datos, debe ser posible rastrear su recorrido desde la fuente hasta el usuario final. Esta gestión del ciclo de vida de los datos constituye la base para gestionar la información geoespacial y medioambiental, así como para llevar a cabo auditorías. (Dormann, 2020, págs. 110-112)

2.3.2.3 Estrategias de Ingesta y Transformación (ETL/ELT) de Datos Heterogéneos

La recopilación de datos en el borde requiere una estrategia que admita el procesamiento de streaming y adapte la arquitectura de datos al volumen de información y la velocidad de transmisión. La transformación de datos sin procesar (ETL/ELT) es un paso importante para convertir los datos a un formato coherente y analizable, lo que incluye la conversión de unidades, la agregación de tiempo y la estandarización de esquemas. (Shen y North, 2023, págs. 70-73)

A la hora de elegir una estrategia de conversión entre el procesamiento in situ y la precarga, es necesario tener en cuenta de forma exhaustiva la complejidad del proceso de limpieza y la necesidad de conservar el historial completo de los registros. La estrategia elegida debe ser lo suficientemente flexible como para tener en cuenta la heterogeneidad inherente a los datos multisensor, al tiempo que permite el análisis estadístico de los datos de series temporales. (Mudelsee, 2020, págs. 70-73)

2.3.2.4 Calidad, Limpieza y Validación de Datos de Monitoreo Ambiental

Se determina la calidad de los datos ambientales por la precisión, integridad y coherencia que tengan. Estos deben de reflejar de manera legible fácilmente el valor de la variable que se mide, la cual no debe contener errores. Su verificación inicia confirmando que los valores medidos se encuentran dentro de los parámetros establecidos que se esperan técnicamente (como las variables de temperatura y humedad). (Burt, 2024, págs. 200-202)

Para mitigar o limpiar errores en los datos, la limpieza de estos tiene que ser el objetivo principal. Todo comienza con la identificación de valores anormales, la gestión de los datos

que faltan (haciendo una interpolación o imputación) y corrigiendo errores que existan del formato. Son de gran importancia estos métodos ya que garantizan que los datos almacenados sean adecuados para el análisis estadístico de los datos ambientales. (Dormann, 2020, págs. 140-142)

2.3.2.5 Técnicas de Almacenamiento y Optimización de Consultas para Big Data Ambiental

La gestión eficaz de datos geospaciales y temporales a gran escala desempeña un papel fundamental a la hora de optimizar el almacenamiento y la recuperación de grandes cantidades de datos medioambientales. Este sistema debería utilizar métodos de segmentación e indexación basados en coordenadas geográficas y marcas de tiempo, lo que reducirá considerablemente los tiempos de búsqueda. (Dormann, 2020, págs. 160-163)

Para la optimización es de suma importancia que exista un preprocesamiento de los datos, ya que permiten un análisis más profundo sobre los mismos. Incluyendo el debido uso de las estructuras de datos, ya que permiten una recolección de los datos y un cálculo estadístico. Fundamentalmente la eficiencia de las consultas ayuda a la detección temprana de fenómenos ambientales extremos y como también al desarrollo de los modelos predictivos. (Mudelsee, 2020, págs. 180-182)

2.3.2.6 Visualización y Reporting de Indicadores Ambientales Clave KPIs

Los principios de maximización de la comprensión y prioridad a la presentación de datos son clave en la visualización de datos. Ya que la presentación de los datos mediante gráficos que destaquen la información más importante, como también valores de riesgos, límites de alerta y tendencias a largo de manera clara para el usuario es parte importante de la gestión del medioambiente. (Tufte, 2001, págs. 10-12)

Los informes sobre los indicadores ambientales clave (KPIs) (por ejemplo, la temperatura media diaria y la humedad mínima) deben presentarse en forma de panel de control visual. Estos informes deben reflejar de manera eficaz el estado operativo del vivero y el impacto de las condiciones meteorológicas utilizando métodos de visualización adecuados (mapas, gráficos de líneas). (Burt, 2024, págs. 300-302)

2.3.2.7 Reglamento y Requisitos necesarios para la Gobernanza de la Información Ambiental según estándares Locales e Internacionales

Es necesario en el ámbito de la gestión medioambiental establecer mecanismos que controlen el acceso, almacenamiento de datos y políticas de seguridad que garanticen la fiabilidad de los datos tras los procesos de su ciclo de vida, Estos sistemas abarcan factores fundamentales que deben garantizar la trazabilidad, la reproducibilidad e integridad de los datos, con el fin de mantener la calidad y fiabilidad de las pruebas. Llevando una gestión rigurosa permite auditar los datos ambientales sin comprometer su fiabilidad, sino también garantizar los datos sean reutilizables para nuevas investigaciones. (Shen y North, 2023, págs. 300-302)

De conformidad con el marco normativo sobre datos medioambientales, la arquitectura del sistema debe garantizar una documentación adecuada, que incluya metadatos detallados (licencias de uso, trazabilidad de los datos, formatos, etc.). Estos requisitos están ampliamente adoptados en las normas internacionales que regulan el intercambio de datos geográficos y medioambientales. El cumplimiento de estos requisitos permite a otras instituciones científicas y organismos reguladores hacer un uso eficaz de esta información. (Dormann, 2020, págs. 250-252)

2.3.3 Metodología de desarrollo Prototipos

Los modelos prototipo permiten crear rápidamente una versión funcional inicial del sistema (aunque sea incompleta) para que los usuarios la evalúen. Este enfoque es de naturaleza recursiva e iterativa, y da prioridad a los comentarios iniciales sobre las características clave del sistema. Su valor reside en reducir la incertidumbre en proyectos cuyos requisitos no están claros en la fase inicial. (Pressman y Maxim, 2021, pág. 45)

La recopilación rápida de las necesidades iniciales son el paso principal en el ciclo de vida del prototipo, después los pasos de diseño y creación del mismo. La fase de evaluación donde se dan los comentarios del diseño por parte del usuario, donde también se analizan las deficiencias y sugerencias para añadirlas es parte esencial de este modelo, ya que este continua hasta que finalmente el prototipo se convierta en un sistema aceptable o descartable. (Sommerville, 2016, pág. 82)

Edge Lakehouse como prototipo en el sistema de monitoreo ambiental automatizado da muchas ventajas, ya que permite realizar pruebas inmediatas de la interfaz donde el usuario la controla y la representación gráfica de los datos que se obtienen así facilitando a los encargados de campo comprender las condiciones del ambiente que son clave para prevenir errores que afecten económicamente y alertar con anticipación. (IEEE, 2018, pág. 10)

El principal problema de este modelo es que existe una tendencia a convertir directamente los prototipos, que presentan un bajo grado de precisión, en el producto final sin el apoyo suficiente de los ingenieros de calidad. Si el ciclo de iteración finaliza prematuramente, pueden quedar en el sistema defectos de diseño, vulnerabilidades arquitectónicas o problemas relacionados con el mantenimiento a largo plazo. Por lo tanto, es extremadamente importante elaborar un plan claro para desmontar y rediseñar el prototipo. (Larman, 2002, pág. 15)

2.4 Conclusiones del marco teórico

Como solución más óptima para la gestión de datos ambientales no centralizados (como ejemplo en el vivero Lastenia Vera de la Uleam El Carmen) se tiene a la arquitectura Edge Lakehouse con tecnología LoRaWAN, ya que esta al combinar la computación en el borde con la conectividad de largo alcance que tiene LoRaWAN, permite que los datos se recopilen de manera eficiente y sostenible como también ofrece escalabilidad y confianza de los datos antes de que se guarden en un solo lugar central.

Para obtener un mejor control de calidad, confiabilidad y trazabilidad de la información recolectada para la gestión de datos ambientales se deben seguir métodos de preprocesamiento, como la verificación de volumen de datos y reemplazo de valores perdidos, ya que son de suma importancia en el procesamiento de datos temporales. Se puede garantizar precisión de los indicadores de rendimiento del medio ambiente gracias a un control del ciclo de vida de los datos lo cual genera un apoyo importante en el análisis de toma de decisiones agronómicas.

Al integrar la arquitectura Edge Lakehouse en LoRaWAN, se gestiona los datos meteorológicos ambientales, eliminando los retrasos operativos y las inconsistencias en el análisis histórico. La configuración de la pasarela como una microarquitectura lakehouse proporciona consistencia de transacción para datos históricos grandes, al tiempo que cumple

con las necesidades críticas para la toma de decisiones en el borde (detección de anomalías), gracias a esta combinación permite un seguimiento real inmediato y confiable.

En el desarrollo óptimo para este proyecto se tiene la creación de prototipos como metodología, ya que esta permite que una verificación seguida de las necesidades de los usuarios al principio, como también la visualización interactiva en la etapa inicial es fundamental en su creación para que sea fácil de entender para el encargado del vivero lo que ocasiona que se reduzcan los riesgos de que los usuarios no aprueben el resultado final y también minimiza el costo de posteriores correcciones.

CAPÍTULO III

3 MARCO INVESTIGATIVO

3.1 Introducción

En este capítulo se describe la metodología aplicada durante el desarrollo de la investigación, la cual permitió sustentar técnicamente la propuesta planteada. Su propósito fue recolectar información confiable sobre la forma en que se gestionan actualmente los datos ambientales en el Vivero Lastenia de la ULEAM Extensión El Carmen, con el fin de identificar las limitaciones existentes y orientar el diseño de la solución.

La investigación se abordó desde un enfoque descriptivo, no experimental y aplicado, dado que se buscó caracterizar la situación tal como ocurre en el vivero, sin intervenir en las variables observadas. Para el tratamiento de la información se emplearon los métodos inductivo-deductivo, de análisis-síntesis y de observación.

La recolección de datos se realizó a partir de fuentes primarias, mediante una encuesta dirigida a los estudiantes de la carrera de Agropecuaria vinculados al vivero y una entrevista aplicada al coordinador del área, definiendo previamente la población, la muestra y la estructura de cada instrumento.

Finalmente, se presentan el análisis e interpretación de los resultados obtenidos, cuyos hallazgos constituyeron la base para definir los requerimientos del sistema y justificar la propuesta desarrollada en el capítulo siguiente.

3.2 Tipos de investigación

3.2.1 Investigación Descriptiva

La investigación descriptiva tiene como objetivo identificar los atributos, características y rasgos comunes de individuos, grupos, comunidades o fenómenos objeto de análisis. Su finalidad principal es medir o evaluar diversos aspectos, dimensiones y componentes del fenómeno estudiado, más que establecer relaciones causales entre ellos. Esta metodología

pretende ofrecer una descripción detallada de cómo se manifiesta y se presenta un fenómeno concreto en su contexto. (Villanueva Couoh, 2022, pág. 45)

Este tipo de investigación se utilizó para describir la situación actual de la gestión de datos ambientales en el vivero de Lastenia. El estudio explicó los métodos de trabajo manuales existentes, evaluó cuantitativamente el nivel de retraso de los datos y realizó un análisis detallado de la variabilidad de los datos meteorológicos recopilados por los sensores. Este trabajo estableció un punto de partida y justificó la necesidad de una nueva arquitectura.

3.2.2 Investigación No Experimental

La característica que define la investigación no experimental es la observación y el análisis de fenómenos que se producen en su entorno natural, sin manipular deliberadamente las variables o las condiciones del estudio. En concreto, las encuestas transversales son una metodología en la que se recopilan datos en un momento específico para describir variables y analizar su prevalencia e interrelaciones. Este enfoque permite comprender la situación actual sin intervenir en ella. (Hernández Sampieri y Mendoza Torres, 2018, pág. 80)

En este proyecto, la investigación no experimental es clave en las etapas de recopilación de información y diagnóstico. Observó que los procesos manuales de recopilación de datos existentes y, tras implementar la nueva arquitectura Edge Lakehouse en un entorno real (viveros de cacao), evaluó su eficacia operativa, sin modificar las condiciones de cultivo del cacao a lo largo de este proceso.

3.2.3 Investigación Aplicada

La investigación aplicada tiene como objetivo resolver problemas prácticos y relevantes en situaciones específicas utilizando los conocimientos adquiridos en la investigación fundamental. Su objetivo no es solo crear teorías, sino también desarrollar herramientas, métodos o soluciones que puedan aplicarse directamente para mejorar los procesos o productos existentes. Este tipo de investigación está orientada a la práctica y tiene como objetivo lograr resultados reales. (Maldonado Pinto, 2018, pág. 35)

La investigación aplicada se utilizó en el desarrollo y la implementación de la arquitectura Edge Lakehouse basada en LoRaWAN. El objetivo es crear una solución técnica funcional y específica (sistema de monitorización automatizado) para resolver los problemas relacionados con la gestión manual de datos en el vivero Lastenia, aumentar significativamente la eficiencia y optimizar la toma de decisiones agronómicas.

3.2.4 Métodos de investigación

3.2.4.1 Método Inductivo-Deductivo

El método inductivo-deductivo es una combinación de dos procesos de pensamiento: la inducción y la deducción. La inducción permite derivar principios universales a partir de la observación de casos específicos, mientras que la deducción utiliza estos principios universales para explicar casos específicos. Este enfoque integrado es crucial para poner a prueba hipótesis y extraer conclusiones teóricas. (Hernández Sampieri y Mendoza Torres, 2018, pág. 110)

En este proyecto la inducción examinó problemas específicos que surgen del vivero (errores en la gestión del personal, grandes retrasos) y utilizó el razonamiento inductivo para establecer los principios que subyacen a los requisitos de automatización. A continuación, utilizando la deducción, se demostró que la arquitectura Edge Lakehouse (una solución integrada) es una solución técnica específica capaz de resolver estos problemas locales y específicos en los viveros.

3.2.4.2 Método de Análisis-Síntesis

El método analítico-sintético implica dos procesos intelectuales complementarios. El análisis consiste en descomponer un todo complejo en sus partes constituyentes, estudiar estos elementos por separado y comprender sus interrelaciones. La síntesis se centra entonces en reconstruir los elementos estudiados e integrarlos en nuevas estructuras o conceptos para explicar el todo. (Maldonado Pinto, 2018, pág. 42)

Esta metodología es directamente aplicable al diseño de la arquitectura del sistema. En la fase de análisis, el problema se dividió en módulos (sensor, LoRaWAN, Edge Computing, arquitectura Lakehouse). La fase de síntesis llevó a cabo en la fase de diseño de la solución,

creando una arquitectura Lakehouse periférica funcional mediante la integración de todos los componentes (hardware, comunicaciones, base de datos, software).

3.2.4.3 Método de Observación

Procesos y técnicas de interpretación de la realidad, descripción y comprensión de los eventos del entorno que se rodea es parte fundamental del método de observación ya que esta debe de realizarse de manera cuidadosa siguiendo un plan que permita obtener los datos sin alterarlos. En las ciencias aplicadas y en las ciencias sociales este método es muy importante ya que mediante esta se obtiene la información principal inicial. (Villanueva Couoh, 2022)

El método de observación se utilizó durante las fases de diagnóstico y evaluación. En primer lugar, se observaron los procesos manuales actuales de la guardería para identificar los pasos y requisitos ineficientes. A continuación, este método se utilizó para supervisar el estado operativo del sistema Edge Lakehouse ya implantado, verificó la precisión de los sensores y comprobó la capacidad de generar alertas en tiempo real.

3.3 Fuentes de información de datos

3.3.1 Fuentes primarias

3.3.1.1 Encuestas

Una encuesta es un método de recopilación de datos mediante la distribución de cuestionarios estandarizados a una muestra representativa de la población. Su objetivo principal es obtener información cuantitativa y descriptiva sobre las características, opiniones o comportamientos de un grupo específico en un momento determinado. Las encuestas permiten realizar análisis estadísticos posteriores, ya que los datos se recopilan utilizando un método uniforme. (Zacarías y Supo, 2020, pág. 105)

La encuesta se realizó al estudiante de la carrera de Agropecuaria, los cuales realizaban actividades de frecuentes en el campus “Lastenia Vera” de la extensión Uleam El Carmen. Estos estudiantes tenían un acercamiento temprano en el ámbito del cuidado de las plántulas de cacao en un entorno con variables ambientales. El objetivo de la encuesta fue obtener una comprensión profunda de los problemas relacionados con la gestión de datos y formular los

requisitos del sistema. Se abordaron cuestiones clave como la frecuencia y los métodos de recopilación de datos meteorológicos, la concienciación sobre los retrasos en la recepción de información y la necesidad de alertas tempranas para la toma de decisiones agrícolas.

3.3.1.2 Entrevista

La entrevista mediante el dialogo de persona a persona o mediante otros medios tecnológicos es un método de recopilación de datos que tiene como objetivo obtener información específica, totalmente diferente a las encuestas, ya que permite profundizar de mejor manera las opiniones, experiencias o conocimientos a través de preguntas abiertas lo que como consecuencia genera respuestas más detalladas y contextualizadas. (Villanueva Couoh, 2022, pág. 105)

Se realizó una entrevista al coordinador Mg. Pedro Nivelá, responsable del vivero de “Lastenia Vera” de la Uleam extensión El Carmen, basándose en su experiencia directa en la gestión de cultivos y los requisitos de cultivo. La entrevista se centró en cuestiones técnicas, entre las que se incluyen las siguientes: resistencia de las plántulas a los riesgos climáticos, umbrales críticos de temperatura y humedad para las plántulas de cacao, y las elevadas exigencias en los sistemas de gestión de datos.

3.4 Estrategia operacional para la recolección de datos

3.4.1 Población

Una población (o universo) es un conjunto completo de individuos, objetos o fenómenos que comparten una o más características comunes sobre las que se busca información. Cuando la población es demasiado grande, se utilizan métodos de muestreo. Es decir, se selecciona un subgrupo representativo de la población para su estudio y los resultados se generalizan a la población en general. (Tamayo y Tamayo, 2014, pág. 115)

La población de estudio para este proyecto, estuvo conformada por los estudiantes de la carrera de Agropecuaria que directamente estaban involucrados en la gestión de datos, cuidado y mantenimiento de las plantaciones de cacao en el vivero del campus “Lastenia Vera” de la Uleam extensión El Carmen. Dada la estructura especializada del personal, este estudio consideró una población total de 88 estudiantes que gestionaban directamente las actividades,

junto con el coordinador del área del vivero, para garantizar una comprensión más completa de la problemática.

3.4.2 Muestra

Una muestra es un subconjunto finito representativo extraído de una población que se utiliza con fines de investigación para extraer conclusiones sobre la población. Los métodos de muestreo se dividen normalmente en muestreo probabilístico (en el que todos los elementos tienen las mismas posibilidades de ser seleccionados) y muestreo no probabilístico. La elección del tipo de muestreo viene determinada por la naturaleza y el tamaño de la población. (Pérez Juste et al., 2012)

Para determinar el tamaño de la muestra teniendo en cuenta a la población dirigida, se aplicó la siguiente fórmula considerando que el nivel de confianza sea del 95% y el margen de error de 5%. Sustituyendo los valores con la población $N=88$, el resultado es:

$$n = \frac{N * Z^2 * p * q}{e^2 * (N - 1) + Z^2 * p * q}$$

$$n = \frac{88 * 1.96^2 * 0.5 * 0.5}{0.05^2 * (88 - 1) + 1.96^2 * 0.5 * 0.5}$$

$$n = \frac{88 * 3.8416 * 0.5 * 0.5}{0.0025 * 87 + 3.8416 * 0.5 * 0.5}$$

$$n = \frac{338.0608 * 0.25}{0.2175 + 0.9604}$$

$$n = \frac{84.5152}{1.1779}$$

$$n = 71.75$$

Al tratarse de personas se redondea el resultado, estableciendo como muestra mínima requerida de $n=72$. Sin embargo, durante la fase de recolección de datos se logró la participación de 80

estudiantes, superando la muestra establecida. Lo cual representa de mejor manera a la población total, validando y fortaleciendo los resultados obtenidos de la encuesta.

3.4.3 Análisis de las herramientas de recolección de datos a utilizar

3.4.3.1 Encuesta

La encuesta planteada se conforma de 30 preguntas, dirigida para los estudiantes de la carrera de agropecuaria de la Uleam extensión El Carmen. Con el objetivo de hacer un diagnóstico de las variables ambientales como también validar la problemática sobre el seguimiento manual en el vivero Lastenia Vera. Dentro de los temas en los que se conforma esta encuesta se encontraron los siguientes que incluyen los registros físicos, el impacto que tiene el cambio climático en las plántulas y la utilidad que tiene el implementar tecnologías de Internet de las cosas y alertas automáticas para el cuidado de las plántulas del vivero.

3.4.3.2 Entrevista

La entrevista planteada está conformada por 13 interrogantes, la cual está dirigida al responsable coordinador del Vivero Lastenia a cargo del Ing. Pedro Nivelá. Con el objetivo de profundizar sobre los niveles y rango óptimos para que las plántulas sobrevivan, identificar deficiencias en los procedimientos que siguen de manera manual los estudiantes y validar la falta de un historial de datos en el control fitosanitario. Esta entrevista incluye temas como el estrés hídrico, la verificación de los datos y consistencia, como la verificación de la consistencia y las estrategias de comunicación al presentarse una anomalía en el vivero.

3.4.3.3 Estructura de los instrumentos de recolección de datos aplicados

3.4.3.3.1 Estructura de entrevista

A La estructura de la entrevista aplicada se puede revisar en el ANEXO D

3.4.3.3.2 Estructura de encuesta

A La estructura de la encuesta aplicada se puede revisar en el ANEXO E

3.4.4 Plan de recolección de datos

Tabla 1: Recolección de Datos

ACTIVIDAD	RESPONSABLE	FECHA DE EJECUCIÓN
Entrevista al Coordinador encargado del Vivero Lastenia, al Mg, Pedro Nivelá.	Anderson Steven Plua Chamorro Dereck Elian Castro Segovia David Ricardo Gerrero Salvatierra Kevin David Rodríguez Ordoñez Anthony Javier Zambrano Alcivar Bryan David López Loor	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	Anderson Steven Plua Chamorro Dereck Elian Castro Segovia David Ricardo Gerrero Salvatierra Kevin David Rodríguez Ordoñez Anthony Javier Zambrano Alcivar Bryan David López Loor	28/01/2026 – 21/04/2026

3.5 Análisis y presentación de resultados

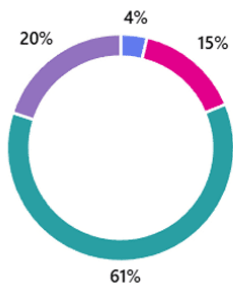
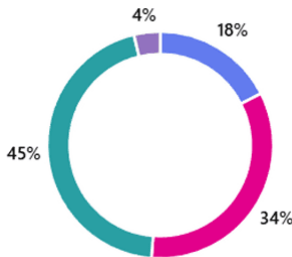
3.5.1 Presentación y descripción de los resultados obtenidos

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

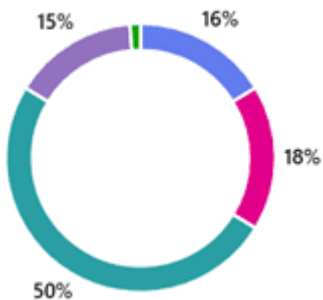
El Carmen

Tabla 2: Análisis de encuesta dirigida a estudiantes

PREGUNTA	GRÁFICO	INTERPRETACIÓN
Pregunta 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 16 4 46 14	De acuerdo con la mayoría de encuestados se observa que el acceso a información en tiempo real es fundamental para la gestión del vivero. La ausencia de estos datos inmediatos puede ocasionar problemas como un diagnóstico fitosanitario tardío y tomar decisiones incorrectas por la falta de información oportuna sobre el estado de las plantas.
Pregunta 11: ¿Cómo valora la utilidad de recibir alertas automáticas (ej. notificaciones de calor extremo) para la	● Inútil ● Poco útil ● Útil ● Muy útil 3 12 49 16	De acuerdo con la mayoría de encuestados se observa que la recepción de alertas automáticas es considerada una herramienta de gran utilidad para la prevención

PREGUNTA	GRÁFICO	INTERPRETACIÓN												
prevención de daños en los cultivos?	 <table> <tr> <th>Respuesta</th> <th>Cantidad</th> </tr> <tr> <td>Muy de acuerdo</td> <td>4</td> </tr> <tr> <td>De acuerdo</td> <td>15</td> </tr> <tr> <td>Neutral</td> <td>36</td> </tr> <tr> <td>En desacuerdo</td> <td>20</td> </tr> <tr> <td>Muy en desacuerdo</td> <td>0</td> </tr> </table>	Respuesta	Cantidad	Muy de acuerdo	4	De acuerdo	15	Neutral	36	En desacuerdo	20	Muy en desacuerdo	0	de daños. La carencia de estas notificaciones inmediatas ante eventos de calor extremo puede ocasionar problemas como estrés térmico severo en las plántulas y tomar decisiones incorrectas por la falta de información preventiva oportuna.
Respuesta	Cantidad													
Muy de acuerdo	4													
De acuerdo	15													
Neutral	36													
En desacuerdo	20													
Muy en desacuerdo	0													
Pregunta 12: ¿Cree usted que el registro manual de datos de riego y clima aumenta el riesgo de cometer errores en la información recolectada?	<table> <tr> <th>Respuesta</th> <th>Cantidad</th> </tr> <tr> <td>Muy de acuerdo</td> <td>14</td> </tr> <tr> <td>De acuerdo</td> <td>27</td> </tr> <tr> <td>Neutral</td> <td>36</td> </tr> <tr> <td>En desacuerdo</td> <td>3</td> </tr> <tr> <td>Muy en desacuerdo</td> <td>0</td> </tr> </table> 	Respuesta	Cantidad	Muy de acuerdo	14	De acuerdo	27	Neutral	36	En desacuerdo	3	Muy en desacuerdo	0	De acuerdo con la mayoría de encuestados se observa que el registro manual de datos aumenta significativamente el riesgo de cometer errores en la información recolectada. La persistencia de esta metodología tradicional puede ocasionar problemas como la falta de integridad en los reportes y tomar decisiones incorrectas por la falta de información confiable sobre las condiciones climáticas del vivero.
Respuesta	Cantidad													
Muy de acuerdo	14													
De acuerdo	27													
Neutral	36													
En desacuerdo	3													
Muy en desacuerdo	0													

PREGUNTA	GRÁFICO	INTERPRETACIÓN
Pregunta 18: La falta de un repositorio centralizado de datos genera confusión e inconsistencia en los informes finales míos o de mi grupo.	Muy de acuerdo14 De acuerdo36 Neutral29 En desacuerdo1 Muy en desacuerdo0	De acuerdo con la mayoría de encuestados se observa que la falta de un repositorio centralizado de datos genera confusión e inconsistencia en los informes finales. La carencia de una plataforma unificada para el almacenamiento de información puede ocasionar problemas como la fragmentación de los registros y tomar decisiones incorrectas por la falta de información coherente y centralizada sobre las variables del vivero.
Pregunta 25: ¿Cómo considera la falta de alertas automáticas para el cuidado de las plántulas cuando no hay personal presente?	Muy adecuada13 Adecuada14 Neutral40 Poco adecuada12 Nada adecuada1	De acuerdo con la mayoría de encuestados se observa que la falta de alertas automáticas es considerada una condición desfavorable para el cuidado de las plántulas ante la ausencia de personal. La carencia de este sistema de monitoreo

PREGUNTA	GRÁFICO	INTERPRETACIÓN
	 A donut chart with four segments. The segments are colored purple (15%), blue (16%), pink (18%), and teal (50%). There is a small green segment at the top of the purple segment.	autónomo puede ocasionar problemas como la falta de respuesta ante riesgos climáticos imprevistos y tomar decisiones incorrectas por la falta de información inmediata sobre el estado del vivero.

3.5.1.2 Análisis de entrevista dirigida a Coordinador de Vivero Lastenia Vera

Tabla 3: Análisis de entrevista dirigida a Coordinador

PREGUNTA	RESPUESTA	INTERPRETACIÓN
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?	“Claro, como vemos existe muchas fluctuaciones en el clima, estas fluctuaciones producen estrés en las plantas de cacao, entonces si es necesario tener un monitoreo constante con equipos para saber cuándo sube la temperatura para activar mecanismos para contrarrestar y no afecte al metabolismo de la planta”	De acuerdo con la respuesta se dice que existen factores que afectan a las plantas de cacao y que es necesario un monitoreo para contrarrestar las percusiones de estas afectaciones que se dan.

PREGUNTA	RESPUESTA	INTERPRETACIÓN
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?	“Justamente tenemos una mini estación meteorológica en nuestra carrera de agropecuaria, en base a esos datos se hace una proyección y un análisis para el tema ambiental”	Actualmente se puede verificar que el proceso para registrar las variables ambientales se realiza mediante una mini estación que está limitada
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?	“Es importante las fluctuaciones de las lluvias ya que permite tener mayor o menor presencia de enfermedades, entonces si sirve como análisis estadístico para tener un bosquejo para tener presencia de qué ambiente vamos a tener debido al cambio climática, muchas veces no es tan constante los datos históricos entonces siempre tenemos que estar actualizados en la información”	Aquí se hace relevante la importancia de la presencia de los datos históricos para prever posibles riesgos y afectaciones que puede tener el vivero a causa de las fluctuaciones, no es constante los datos históricos por ende buscan estar actualizados de alguna forma
10. ¿Cuáles son las estrategias de comunicación que se utilizan para resolver	“Existe el apoyo de diferentes personas como especialistas tanto en la parte	De acuerdo con esta respuesta se puede entender que existe apoyo manual

PREGUNTA	RESPUESTA	INTERPRETACIÓN
anomalías críticas entre los estudiantes y la dirección técnica cuando no hay presencia física en el sector Lastenia Vera?	ambiental, en la parte objetiva, Fito sanidad, nutrición, esto ha permitido poder afrontar en conjunto con los estudiantes las novedades que se nos han presentado”	humano, de especialistas y otras personas que forman parte de esa área, mas no una alternativa para ejecutar acciones pertinentes ante anomalías, que se realice cuando no hay presencia física.
11. ¿De qué manera se verifica hoy que los datos recolectados por los estudiantes sean consistentes y no presenten duplicidades antes de ser analizados?	“Hay varias estrategias, primero en el manejo de la información, se les explica a los estudiantes que existe la ética profesional, entonces tenemos que ser íntegros en el manejo de la información, no por afectar o favorecer vamos a alterar datos”	Podemos tener en claro con esta respuesta que no hay una verificación objetiva y consistente, que sea fiable a duplicados, ya que solo se basan en la ética profesional más no en un manejo que sea real consistente de información.

3.5.2 Informe final del análisis de los datos

En cuanto al proceso de monitoreo y reporte en tiempo real, los encuestados en las preguntas 8 y 25 señalaron que la falta de alertas dificulta la reacción ante los cambios climáticos. Esto coincide con lo expresado por el docente entrevistado en la pregunta 11, quien afirmó que no existen estrategias de comunicación efectivas para atender las anomalías que surgen cuando no hay personal presente. Ambos hallazgos evidencian una brecha en la supervisión autónoma del vivero y confirman de manera directa una de las causas del problema: la ausencia de alertas preventivas ante condiciones adversas, que impide la toma de decisiones oportunas.

Respecto a la integridad y la calidad de la información recolectada, los resultados de las preguntas 12 y 18 de la encuesta indican que el registro manual incrementa el riesgo de errores y genera confusión en los reportes. En la misma línea, el coordinador manifestó en la entrevista que no existen mecanismos para verificar la consistencia de los datos ni para evitar su duplicidad. Estos resultados sustentan otra de las causas del problema: el registro fragmentado y desactualizado de los datos climáticos, un proceso que, al depender de métodos físicos y sin gobernanza, compromete la fiabilidad de la información para los análisis posteriores.

Finalmente, en relación con la utilidad del análisis para la prevención de enfermedades, las respuestas de la pregunta 11 reflejan que las notificaciones para evitar posibles daños son de alto valor para los encuestados. El coordinador, en la pregunta 10, confirmó la ausencia de datos históricos que permitan actuar de forma anticipada en el control de plagas. Esta situación se relaciona directamente con la causa del problema referida a la medición manual y esporádica de los parámetros, ya que la falta de un registro continuo impide conservar el histórico necesario para anticiparse a las enfermedades y respalda que la gestión de datos actual es deficiente.

CAPÍTULO IV

4 MARCO PROPOSITIVO

4.1 Introducción

En este capítulo se presenta una explicación detallada de la evolución técnica y metodológica de la solución sugerida para el vivero Lastenia Vera. Utilizando la metodología ágil Scrum, examinaremos los procedimientos de diseño, implementación y planificación de un sistema para gestionar datos ambientales que se basa en el enfoque Edge Lakehouse. Los requerimientos del sistema, el modelado UML (diagrama de transición de estados, diagramas de secuencia y casos de uso) y los reportes acerca del avance de los sprints se exponen en las secciones subsiguientes. Asimismo, se explica cómo se desarrolla una aplicación móvil fundamentada en una base de datos relacional, la cual está diseñada para asegurar un desempeño autónomo y la capacidad de revisar los datos medioambientales.

4.2 Descripción de la propuesta

La implementación de la arquitectura Edge Lakehouse sugerida, que se fundamenta en LoRaWAN, tiene como objetivo posibilitar el manejo automatizado de los datos meteorológicos en el vivero Lastenia Vera. Este ecosistema incorpora sensores de Internet de las cosas (IoT) que se encargan de la recolección de datos. El procesamiento de estos datos tiene lugar en el borde utilizando Edge Computing.

Sin embargo, es importante destacar que el alcance de este trabajo de titulación y su aporte técnico específico se restringen a bases de datos móviles y alojamiento de datos (Módulo 3). La seguridad del servidor central y la configuración de hardware de LoRaWAN son parte de otros módulos de este proyecto integrador; no obstante, el diseño, la puesta en marcha y la administración de una base de datos local relacional (SQLite) en dispositivos móviles es lo que se abordará principalmente en esta propuesta. El procesamiento de consultas y el almacenamiento local de datos son gestionados por este componente, que es el núcleo central del sistema. Este último proporciona un almacenamiento de datos organizado y constante, además de posibilitar la operación sin conexión, lo cual favorece la toma de decisiones en el sector agrícola.

La coordinación efectiva mientras el sistema esta funcionando de debe asegurar estableciendo de manera clara las responsabilidades de cada parte, permitiendo que lo administradores supervisen mediante el sistema de manera técnica y examinen el estado de conexión de la red. Por consiguiente, los usuarios en este caso los agricultores y estudiantes, tiene la posibilidad de consultar los registros digitales y examinar los datos ambientales históricos mediante el empleo de la aplicación móvil en el campo, facilitando la toma de decisiones oportunas. Terminando así con un chatbot que mediate IA examina los datos históricos ambientales ayuda con sugerencias para el cuidado de los cultivos. Los roles distribuidos aseguran que cada uno obtenga datos exactos, ayudando a mejorar el control de la salud de las plantas e incrementar la eficacia operativa en el vivero Lastenia Vera.

4.3 Determinación de recursos

4.3.1 Humanos

Tabla 4: Determinación recursos humanos

Recurso	Cargo	Función
Anderson Plua	Desarrollador	Diseñar la base de datos móvil y la arquitectura de alojamiento local para la gestión del monitoreo ambiental.
Ing. Rómulo Arévalo	Supervisor	Encargado del seguimiento del correcto desarrollo del proyecto de titulación.
David Guerrero	Desarrollador	Diseña la base de datos en la nube y sincronización segura como punto de colección central de los datos.
Mg. Pedro Nivelá	Coordinador	Dispone de la información necesaria de campo para aplicar al sistema de monitoreo.

4.3.2 Tecnológicos

Tabla 5: Determinación de recursos tecnológicos

Recurso	Característica	Descripción
Laptop	Marca DELL, procesador AMD Ryzen 7 5700U with Radeon Graphics, 16 GB de RAM,	Utilizada para el diseño y desarrollo del sistema de

Recurso	Característica	Descripción
		transmisión y aplicación con base de datos.
Visual Studio Code	Versión 1.115 Release date: April 8, 2026	IDE utilizado para el desarrollo y emulación de la aplicación móvil.
PlatformIO	PlatformIO (extensión de Visual Studio Code) — con el framework Arduino para ESP32	Entorno utilizado para la configuración y desarrollo del comportamiento del ESP32 de los nodos sensores y el Gateway.
ESP32	ESP32 de 38 pines, WIFI + BT SoC Inside – JSM2 802.11/b/g/n	ESP32 microcomputador utilizado para definir el comportamiento y definición de los datos,
Modulo Lora RYLR998	Módulo con antena integrada, de 4 pines, REYAX RYLR998 25/23	Módulo para transmitir los datos recolectados de larga distancia sin conexión a internet.
Sensor DHT22	Sensor de temperatura con 3 pines	Sensor específicamente utilizado para la captura de datos temperatura ambiental.

4.3.3 Económicos

Tabla 6: Determinación de recursos económicos

Cantidad	Recurso	Precio unitario	Subtotal
1	Laptop	\$650	\$650
1	Teléfono celular	\$150	\$150
1 año	Alojamiento	\$3.33	\$39.96
2	ESP32 (38 pines)	\$10.65	\$21.30
2	Módulos Lora RYLR998	\$47.60	\$95.20
4 meses	Internet	\$25	\$100
288 horas	Horas de desarrollo	\$5.00	\$1.440
1	Sensor DHT22	\$4.50	\$4.50
TOTAL:			\$ 2,500.96

4.4 Desarrollo del sistema de trabajo bajo la metodología Scrum

Se utilizó la metodología Scrum porque sus características se adaptan perfectamente a la creación de prototipos y permiten un desarrollo iterativo e incremental, al tiempo que integran el hardware y el software. Al trabajar en sprints, permite probar y optimizar la conectividad de los sensores y la transmisión de datos, adaptándolos a las necesidades reales del vivero Lastenia Vera. La flexibilidad de este método nos ayuda a resolver problemas y a adaptarnos al trabajo específico de los agrónomos.

Esta metodología permite una gestión eficaz del tiempo y los recursos y garantiza la implementación por fases de productos funcionales. Scrum divide la arquitectura Edge Lakehouse en módulos manejables, como la recopilación y el análisis de datos, y contribuye a garantizar el control de calidad en cada etapa. Esto minimiza los riesgos y garantiza que las soluciones y los productos finales aborden de forma eficaz los retos que plantea la gestión de datos medioambientales.

4.4.1 Descripción del producto

4.4.1.1 Propósito del producto

Crear un prototipo operativo que se base en una arquitectura concreta. Esta estructura posibilita la alerta anticipada a través de la incorporación y el uso de varios sensores del Internet de las cosas (IoT), la recolección de datos medioambientales y su procesamiento en el borde. La arquitectura Edge Lakehouse, que se basa en la tecnología LoRaWAN, constituye el fundamento de esta estructura arquitectónica. Este prototipo tiene como propósito mejorar los procesos de gestión y las actividades operativas en el vivero Lastenia Vera de la Uleam extensión El Carmen. Está creado para almacenar información centralizada, en un formato estructurado, dentro de un almacén de datos. Esto facilita el análisis de datos históricos. Los términos que se emplean en la arquitectura (como Lakehouse, LoRaWAN y Edge) son utilizados conforme a las funciones que cada nivel del sistema lleva a cabo. Para conseguir información acerca de las formas específicas de implementación, revise la sección "Adaptación tecnológica del proyecto".

4.4.1.2 Funcionalidades Clave

4.4.1.2.1 Funcionalidades del módulo (base de datos móvil):

- Recepción de datos por Bluetooth Clásico
- Alojamiento local en SQLite
- Cola de retransmisión (store-and-forward)
- Consulta del histórico sin conexión
- Aviso por umbral
- Verificación de conectividad real

4.4.1.2.2 Funcionalidades del sistema integrado (otros módulos):

- Captura ambiental en la estación
- Transmisión por LoRa
- Cifrado y descifrado
- Almacenamiento centralizado en el servidor
- Generación de reportes históricos

4.4.1.3 Usuarios objetivos

Tabla 7: Tipos de usuarios objetivos

Tipo de usuario	Principales funcionalidades
Administrador	Posee privilegios de alto nivel y acceso a un dashboard (panel de control) web exclusivo para computadora. Desde esta interfaz de escritorio, supervisa de manera global los datos ambientales monitoreados, gestiona los accesos y evalúa el estado general del sistema integrado.
Usuario (Estudiantes y Agricultores)	Son los protagonistas y usuarios directos de la aplicación móvil en el terreno. Emplean sus dispositivos móviles para acceder a la bitácora digital y observar el historial de las variables medioambientales de manera organizada y sin conexión a Internet (gracias a SQLite). Para tomar decisiones agronómicas, como la protección de las plántulas o el riego, tienen acceso a esta información en el lugar.
ChatBot IA	Del Módulo 6, entidad automatizada del sistema. Con el objetivo de ofrecer respuestas o sugerencias automatizadas a los usuarios sobre el cultivo del cacao, procesa las consultas históricas y utiliza los datos ambientales organizados en las bases de datos.

4.4.1.4 Criterios de éxito del Producto

- **Precisión y confiabilidad de Datos:** El sistema debe garantizar la captura y transmisión íntegra de variables ambientales (temperatura, humedad, radiación y viento) con una tasa de error de transferencia mínima para asegurar la confiabilidad del almacenamiento.
- **Reducción de Latencia en Alertas:** La arquitectura debe generar alertas preventivas en tiempo real ante condiciones climáticas adversas, eliminando los retrasos operativos de la gestión manual y permitiendo una reacción inmediata del personal.
- **Integridad y no duplicidad del almacenamiento:** la base de datos local debe conservar los datos de forma estructurada, sin pérdida ni registros duplicados, aplicando control de unicidad sobre la información recibida.
- **Disponibilidad sin conexión:** la aplicación debe conservar y permitir la consulta de la información almacenada localmente, garantizando el acceso a los datos incluso ante la ausencia de conexión a internet.
- **Optimización de recursos:** El sistema debe evidenciar una mejora en la toma de decisiones agronómicas que resulte en la reducción de la tasa de mortalidad de las plántulas de cacao y un uso más eficiente de los recursos hídricos.

4.4.2 Requerimientos del sistema

4.4.2.1 Historias de Usuario

4.4.2.1.1 Historia de Usuario 1: Monitoreo en tiempo real

Tabla 8: HU1 Monitoreo en tiempo real

HU1: MONITOREO EN TIEMPO REAL			
Id de historia	HU01	Título:	Visualización de variables climáticas en tiempo real
Prioridad:	Alta	Usuario:	Usuario (Estudiantes y Agricultores)
Riesgo de desarrollo:	Medio	Tamaño de la tarea:	Mediano
Como:	Usuario en campo		

HU1: MONITOREO EN TIEMPO REAL	
Quiero:	Visualizar los datos más recientes de temperatura y humedad obtenidos del sistema.
Para:	Conocer el estado climático actual del vivero para la toma de decisiones.
Criterios de aceptación:	El sistema debe mostrar en pantalla los valores climáticos más actualizados extraídos de la base de datos local del dispositivo.
Dependencias:	Conexión a la API REST del servidor y almacenamiento móvil (SQLite) operativo.

4.4.2.1.2 Historia de Usuario 2: Alertas de Humedad Crítica

Tabla 9: HU2 Alertas de humedad crítica

HU2: ALERTAS DE HUMEDAD CRÍTICA			
Id de historia	HU02	Título:	Generación de alertas preventivas automáticas
Prioridad:	Alta	Usuario:	Usuario (Estudiante/Agricultor)
Riesgo de desarrollo:	Alto	Tamaño de la tarea:	Grande
Como:	Usuario en campo		
Quiero:	Recibir una notificación en la aplicación móvil cuando la humedad o temperatura superen los umbrales críticos.		
Para:	Reaccionar preventivamente en el campo y evitar la proliferación de enfermedades fitosanitarias.		
Criterios de aceptación:	La alerta debe dispararse localmente en el móvil al comparar los datos limpios descargados de la API contra los umbrales estáticos definidos.		
Dependencias:	Lógica de evaluación de umbrales configurada en el cliente móvil (Flutter).		

4.4.2.1.3 Historia de Usuario 3: Reportes de datos históricos

Tabla 10: HU3 Reporte de datos históricos

HU3: REPORTES DE DATOS HISTÓRICOS			
Id de historia	HU03	Título:	Consulta de historial climático
Prioridad:	Media	Usuario:	Usuario (Estudiante/Agricultor)
Riesgo de desarrollo:	Bajo	Tamaño de la tarea:	Mediano
Como:	Usuario del sistema		

HU3: REPORTE DE DATOS HISTÓRICOS	
Quiero:	Consultar el historial de variables climáticas registradas sin necesidad de contar con conexión a internet.
Para:	Analizar las tendencias ambientales del vivero directamente en el sitio de cultivo.
Criterios de aceptación:	Los datos mostrados deben ser extraídos directamente de la base de datos relacional local garantizando una navegación fluida offline.
Dependencias:	Base de datos local (SQLite) con registros históricos persistidos.

4.4.2.1.4 Historia de Usuario 4: Gestión de Dispositivos IoT

Tabla 11: HU4 Gestión de Dispositivos IoT

HU4: GESTIÓN DE DISPOSITIVOS IOT			
Id de historia	HU04	Título:	Verificación de estado de sincronización local
Prioridad:	Media	Usuario:	Administrador
Riesgo de desarrollo:	Bajo	Tamaño de la tarea:	Mediano
Como:	Administrador del sistema		
Quiero:	Verificar el estado de la conexión y revisar la fecha de la última sincronización de la base de datos local con la API del servidor central.		
Para:	Garantizar que el dispositivo móvil cuente con la información ambiental más actualizada, detectando posibles fallos de red.		
Criterios de aceptación:	La interfaz debe mostrar claramente la fecha y hora de la última descarga exitosa de datos meteorológicos estructurados.		
Dependencias:	Servicio de consumo HTTP/JSON implementado en la aplicación móvil.		

4.4.2.1.5 Historia de Usuario 5: Bitácora Digital de actividades

Tabla 12: HU5 Bitácora digital de actividades

HU5: BITACORA DIGITAL DE ACTIVIDADES			
Id de historia	HU05	Título:	Registro automatizado de variables meteorológicas
Prioridad:	Baja	Usuario:	Usuario (Estudiante/Agricultor)

HU5: BITACORA DIGITAL DE ACTIVIDADES			
Riesgo de desarrollo:	Bajo	Tamaño de la tarea:	Pequeño
Como:	Usuario en campo		
Quiero:	Acceder a una bitácora digital que muestre automáticamente los datos climáticos organizados y capturados por el sistema.		
Para:	Sustituir completamente la recolección manual de datos en papel, evitando errores humanos y la pérdida de información física.		
Criterios de aceptación:	La bitácora debe mostrar un historial en formato de lista que incluya: fecha, hora exacta y el valor de la variable climática, impidiendo el ingreso o edición manual de estos registros.		
Dependencias:	Estructura de tablas relacionales (SQLite) implementada y operativa en el móvil.		

4.4.2.2 Requerimientos funcionales

- RF01 (Recolección de datos); el sistema debe permitir recibir continuamente las lecturas de las variables ambientales como la temperatura con ayuda de los sensores IoT instalados en el campo.
- RF02 (Procesamiento Edge): los datos que se recopilen deben ser preprocesados en el borde, determinando los límites de los valores críticos antes de enviarse al repositorio central.
- RF03 (Generación de Alertas): si las variables ambientales exceden los valores límites establecidos, el sistema debe de generar una alerta y enviarla de manera automática al teléfono del usuario.
- RF04 (Transmisión LoRaWAN): el sistema debe de ser capaz de enviar paquetes desde los nodos sensores hacia el Gateway utilizando el protocolo de transmisión de bajo consumo y largo alcance LoRaWAN.
- RF05 (Almacenamiento Móvil): Para asegurar la persistencia y disponibilidad de los datos ambientales sin conexión antes de sincronizar, el sistema debe contar con un almacenamiento de datos estructurado en una base de datos local SQLite, la cual esta en el dispositivo celular.
- RF07 (Gestión de Nodos): El administrador debe tener la capacidad para añadir, modificar o borrar los sensores LoRaWAN que se han asignado al vivero Lastenia.

4.4.2.3 Requerimientos no funcionales

- RNF01 (Tiempo de respuesta del aviso): El tiempo transcurrido entre el momento en que una lectura se registra en la base de datos local y la generación del aviso visual al usuario, cuando dicha lectura supera el umbral configurado, no debe exceder los 2 segundos.
- RNF02 (Tolerancia a Fallos): Aunque exista pérdida de conexión la aplicación móvil deberá de seguir funcionando, operando y almacenando datos de manera local, conservándolos para cuando vuelva la conexión estos datos puedan ser reenviados al servidor.
- RNF03 (Eficiencia de energía): Se deben de configurar los nodos sensores para que funcionen con bajo consumo de energía lo que asegura la autonomía dentro de campo.
- RNF04 (Integridad de los Datos): Para prevenir duplicidad de datos y la fiabilidad de los datos almacenados el sistema debe de poner en acción procedimientos que garanticen la integridad de los datos.
- RNF05 (Escalabilidad): Ante el aumento continuo de datos histórico la base de datos debe ser de capaz de gestionar sin comprometer el tiempo de consulta de los mismos.

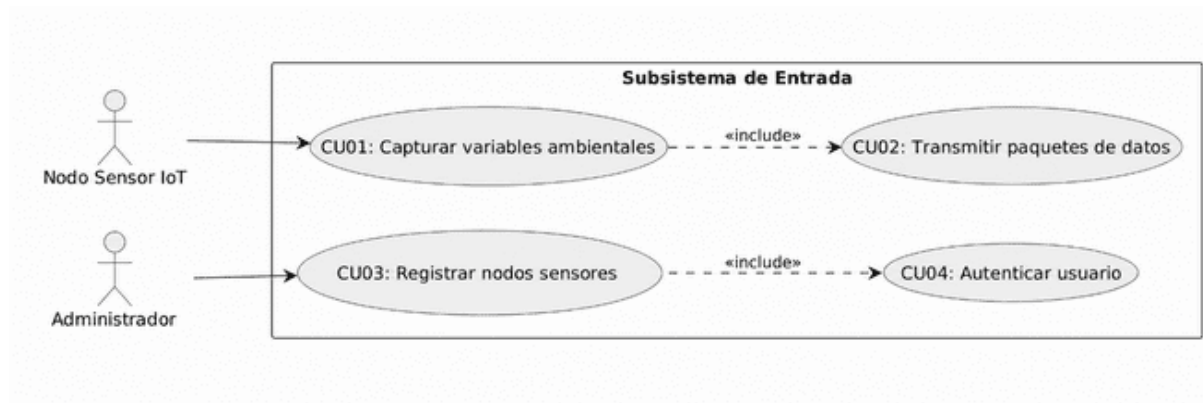
4.4.3 Diseño del Sistema

4.4.3.1 Casos de Uso

4.4.3.1.1 Caso de uso: Entrada de Datos (Captura y Registro)

Este caso de uso modela cómo ingresa la información a la plataforma. Por un lado, representa la lectura y transmisión automática de las variables ambientales (temperatura y humedad) capturadas por los nodos sensores a través del enlace LoRa. Por otro lado, incluye el registro y configuración de nuevos nodos sensores por parte del Administrador Técnico.

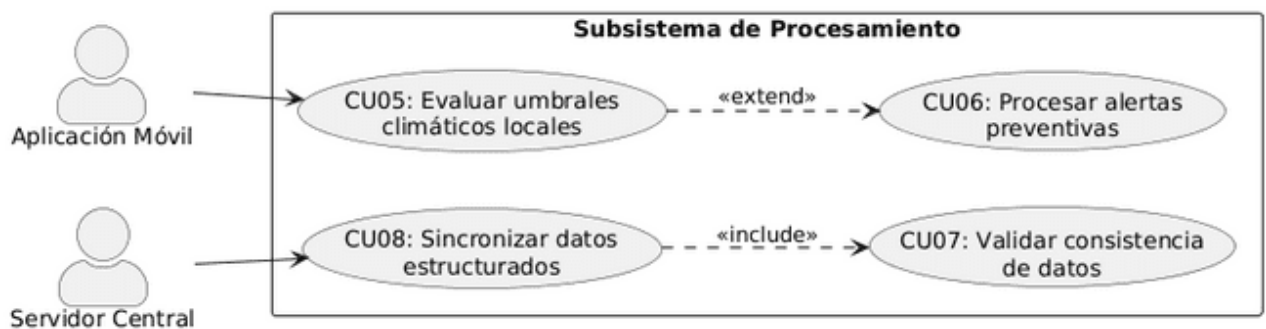
Ilustración 2: Diagrama Caso de Uso (Entrada)



4.4.3.1.2 Caso de uso: Procesamiento (Edge Computing y Sincronización)

El siguiente diagrama representa las operaciones internas y automatización que tiene el sistema. Dando importancia en cómo se evalúan los datos ambientales sobre lecturas almacenadas en la base de datos local que mediante la aplicación móvil determine las alertas de prevención al usuario. Como también validando la consistencia de los datos en el servidor, la cual se encarga de evitar duplicidad antes del almacenamiento final.

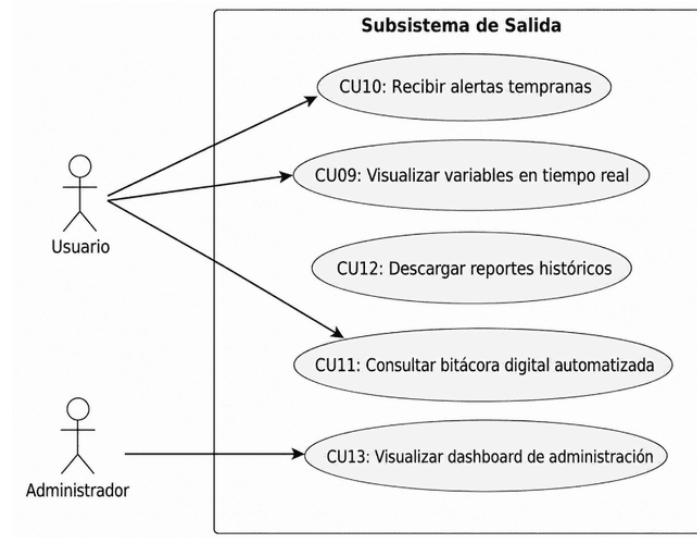
Ilustración 3: Diagrama Caso de Uso (Proceso)



4.4.3.1.3 Caso de uso: Salida de Información (Visualización y Reportes)

Se detalla el consumo de la información procesada por el sistema en este caso de uso, destacando el acceso del usuario a la aplicación móvil para la visualización de las variables ambientales, recibiendo alertas por umbral y consultas mediante la bitácora digital, como también la generación de reportes históricos. Por otro lado, mediante un dashboard el administrador supervisa el estado general del sistema y los datos que provienen del servidor central.

Ilustración 4: Diagrama Caso de Uso (Salida)

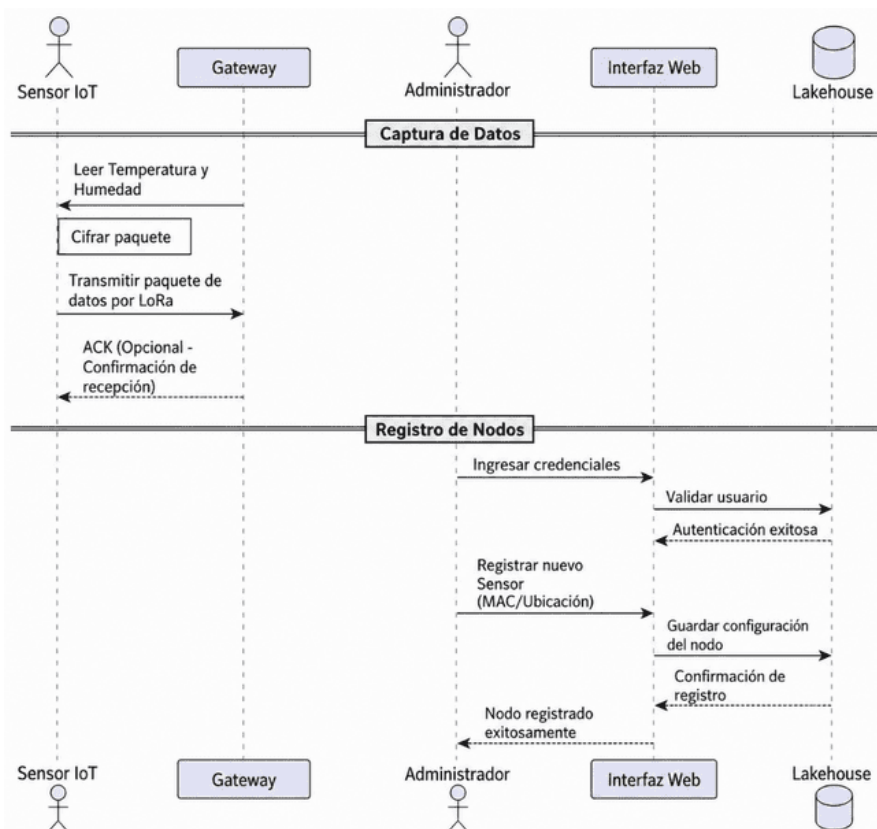


4.4.3.2 Diagramas de Secuencia

4.4.3.2.1 Diagrama de secuencia: Entrada de datos (Captura y Registro)

Este diagrama ilustra la línea de tiempo de dos flujos de entrada. Primero, la comunicación máquina a máquina (M2M), en la cual el nodo sensor captura las variables climáticas, cifra el paquete en el propio dispositivo y lo transmite hacia el Gateway vía LoRa. Segundo, la interacción humana, en la que el Administrador Técnico se autentica en el sistema y registra un nuevo nodo sensor.

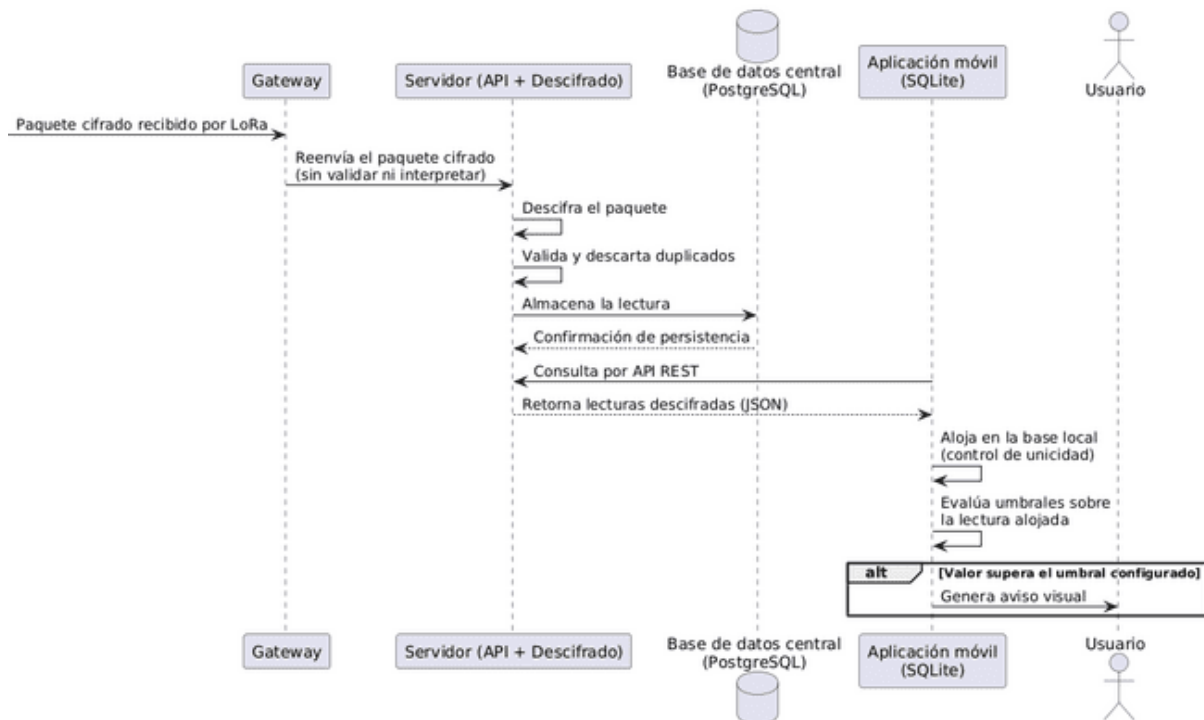
Ilustración 5: Diagrama de Secuencia Entada de datos (Captura y Registro)



4.4.3.2 Diagrama de secuencia: Procesamiento (Edge Computing y Sincronización)

Este diagrama ilustra el trayecto que sigue el paquete desde que se recibe hasta que se consulta. Explica de qué manera el Gateway recibe el paquete cifrado y lo envía nuevamente al servidor sin verificar ni analizar su contenido. El paquete es validado, se eliminan las copias antes de guardar la lectura en la base de datos central y se descifra en el servidor. Luego, la aplicación móvil utiliza la API REST para consultar la información descifrada, la almacena en su base de datos local siguiendo el control de unicidad y analiza los límites configurados para emitir la notificación correspondiente al usuario.

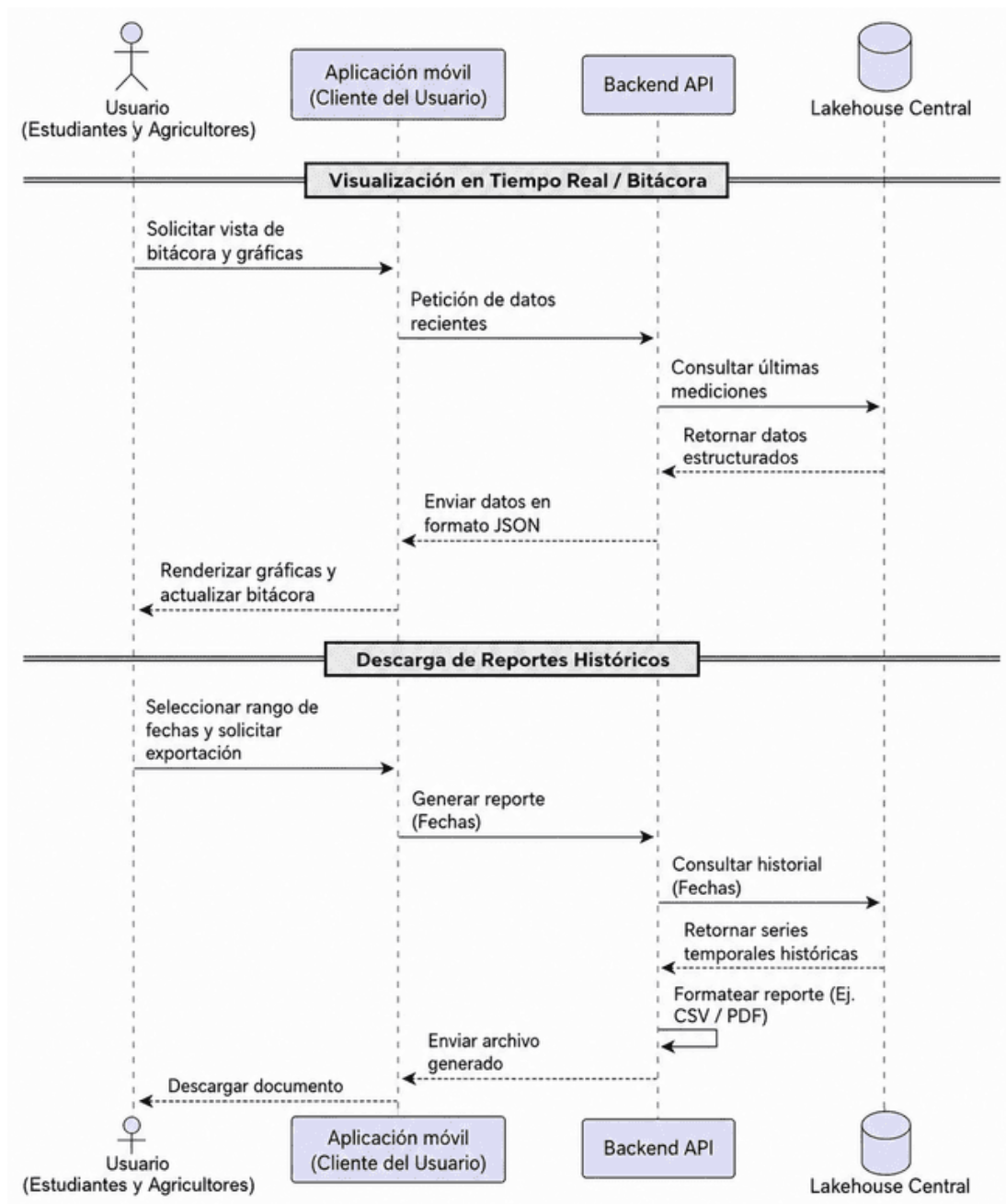
Ilustración 6: Diagrama de Secuencia Procesamiento (Edge Computing y Sincronización)



4.4.3.2.3 Diagrama de secuencia: Salida de información (Visualización y Reportes)

Este diagrama representa la interacción de los usuarios (estudiantes y agricultores) con la plataforma para consumir la información. Muestra el proceso de solicitud desde la aplicación móvil, la consulta que realiza la API al repositorio central y el retorno de los datos estructurados, ya sea para su visualización en la bitácora o para la generación y descarga de un reporte histórico.

Ilustración 7: Diagrama de Secuencia Salida (Visualización y Reportes)



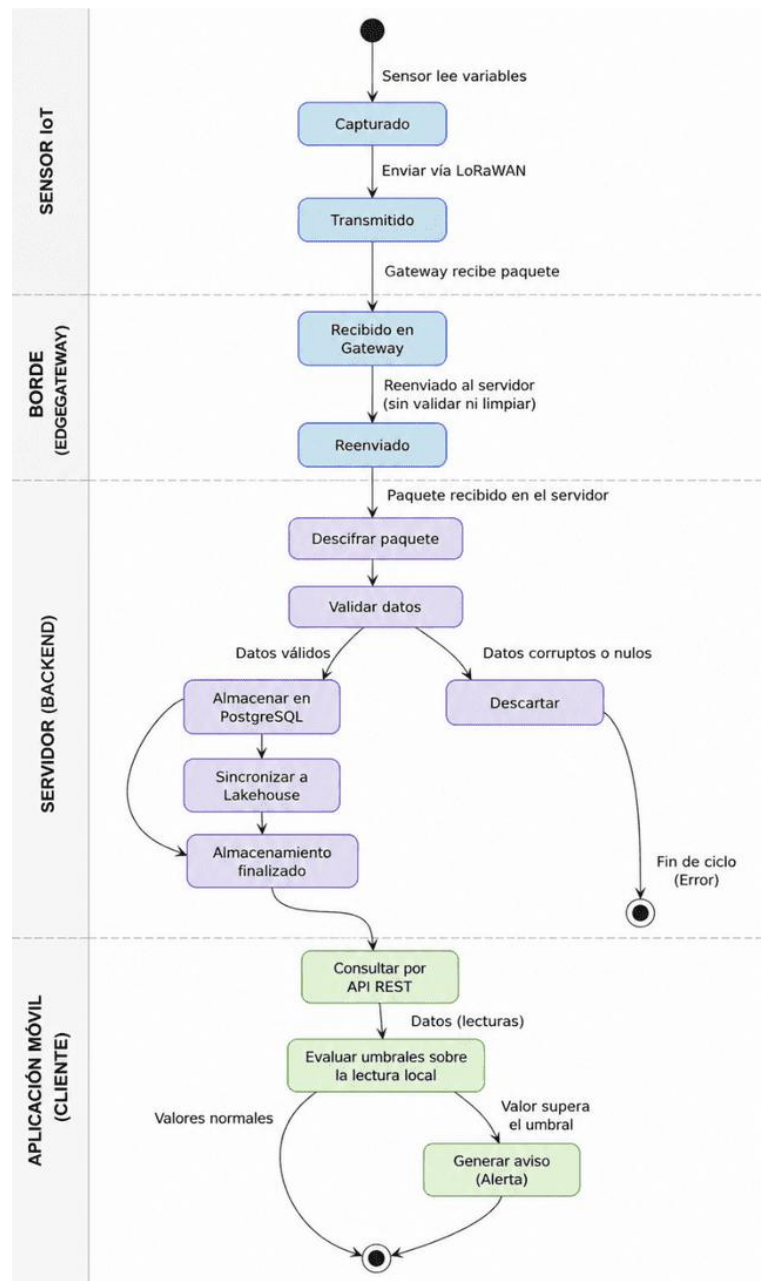
4.4.3.3 Diagramas de Estado

4.4.3.3.1 Diagrama de estado: Paquete de datos Ambientales

Este diagrama ilustra el ciclo de vida de un paquete de información climática desde que es capturado por el sensor hasta que queda disponible para su consulta. El recorrido se organiza en cuatro etapas: la captura y cifrado en el nodo sensor, la retransmisión en el Gateway sin interpretación del contenido, el descifrado y la validación en el servidor —donde el paquete

puede ser descartado si la información resulta corrupta o nula—, y finalmente la consulta desde la aplicación móvil, que aloja la lectura y evalúa los umbrales configurados.

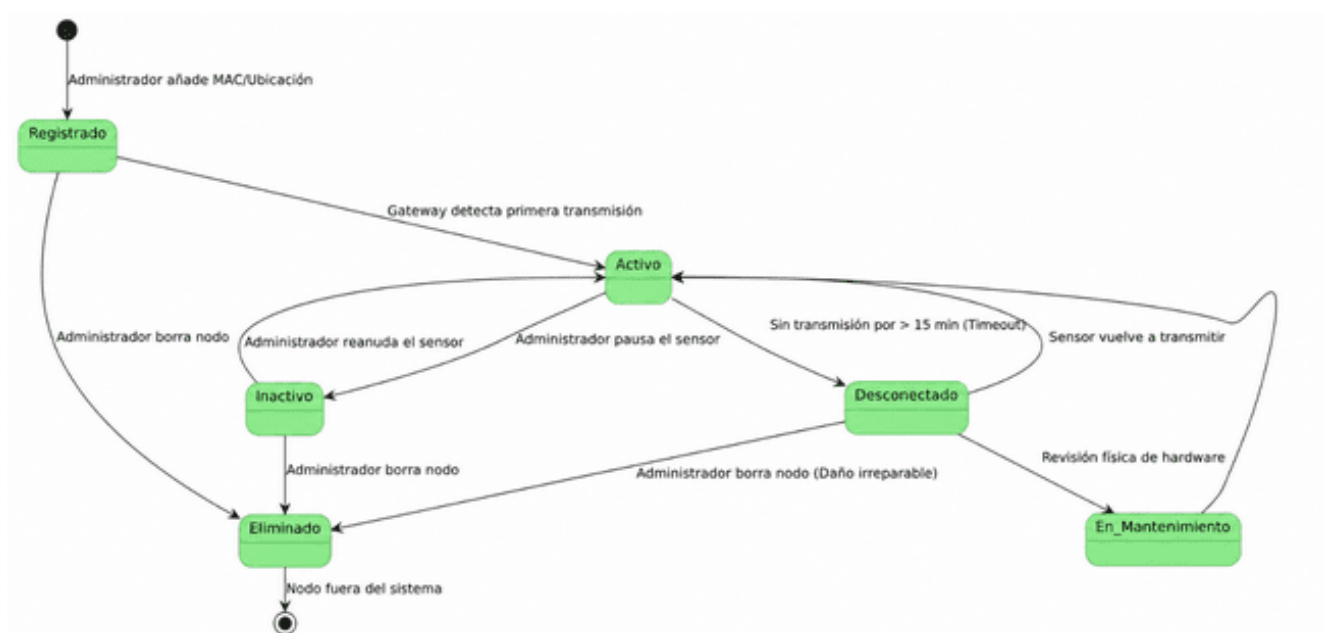
Ilustración 8: Diagrama de Estado Paquete de datos Ambientales



4.4.3.3.2 Diagrama de estado: Nodo Sensor IoT

Este diagrama modela los estados por los que pasa un dispositivo físico (el sensor ESP32/LoRa) dentro del sistema de administración, desde que es registrado hasta que se da de baja o presenta un fallo de conexión.

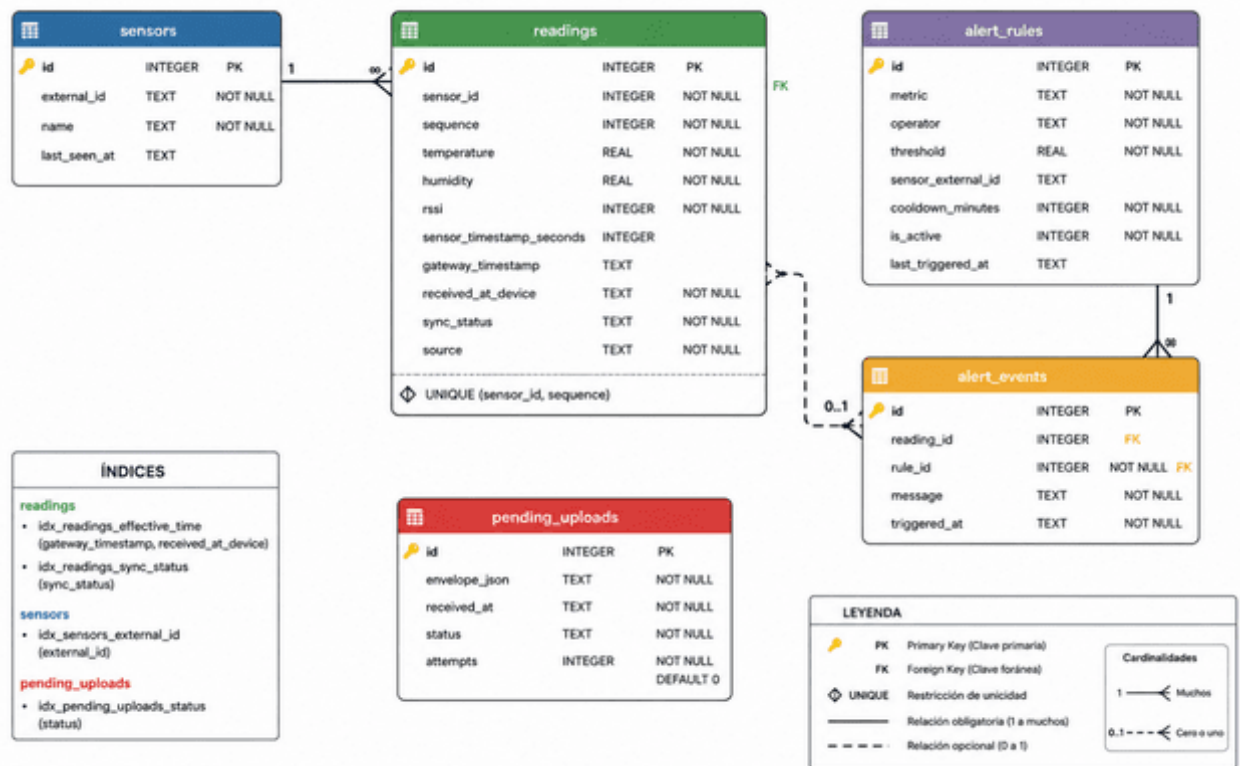
Ilustración 9: Diagrama de Estado Nodo Sensor IoT



4.4.3.4 Diagrama de Base de Datos

El siguiente diagrama presenta la estructura de la base de datos local implementada en el dispositivo móvil. El esquema se compone de dos grupos de tablas con propósitos diferenciados. Por un lado, la tabla `pending_uploads` conserva el paquete cifrado recibido desde el Gateway junto a su estado de envío y su número de intentos, lo que permite retener la información cuando no existe conexión a internet. Por otro lado, las tablas `sensors` y `readings` almacenan las lecturas ya descifradas que la aplicación obtiene del servidor, relacionadas mediante una clave foránea y protegidas por una restricción de unicidad sobre el sensor y el número de secuencia, la cual impide el registro de datos duplicados.

Ilustración 10: Diagrama de base de datos SQLite



4.4.4 Arquitectura del sistema

4.4.4.1 Arquitectura general

El sistema lleva una estructura de desarrollo basada en el Modelo de Arquitectura Distribuida en Capas implementando Cliente/Servidor, ya que esta permite separar la lógica mediante capas que conllevan la captura, procesamiento, almacenamiento y presentación de los datos ambientales meteorológicos, garantizando la disponibilidad de los mismos, baja latencia, resiliencia de conectividad en zonas rurales.

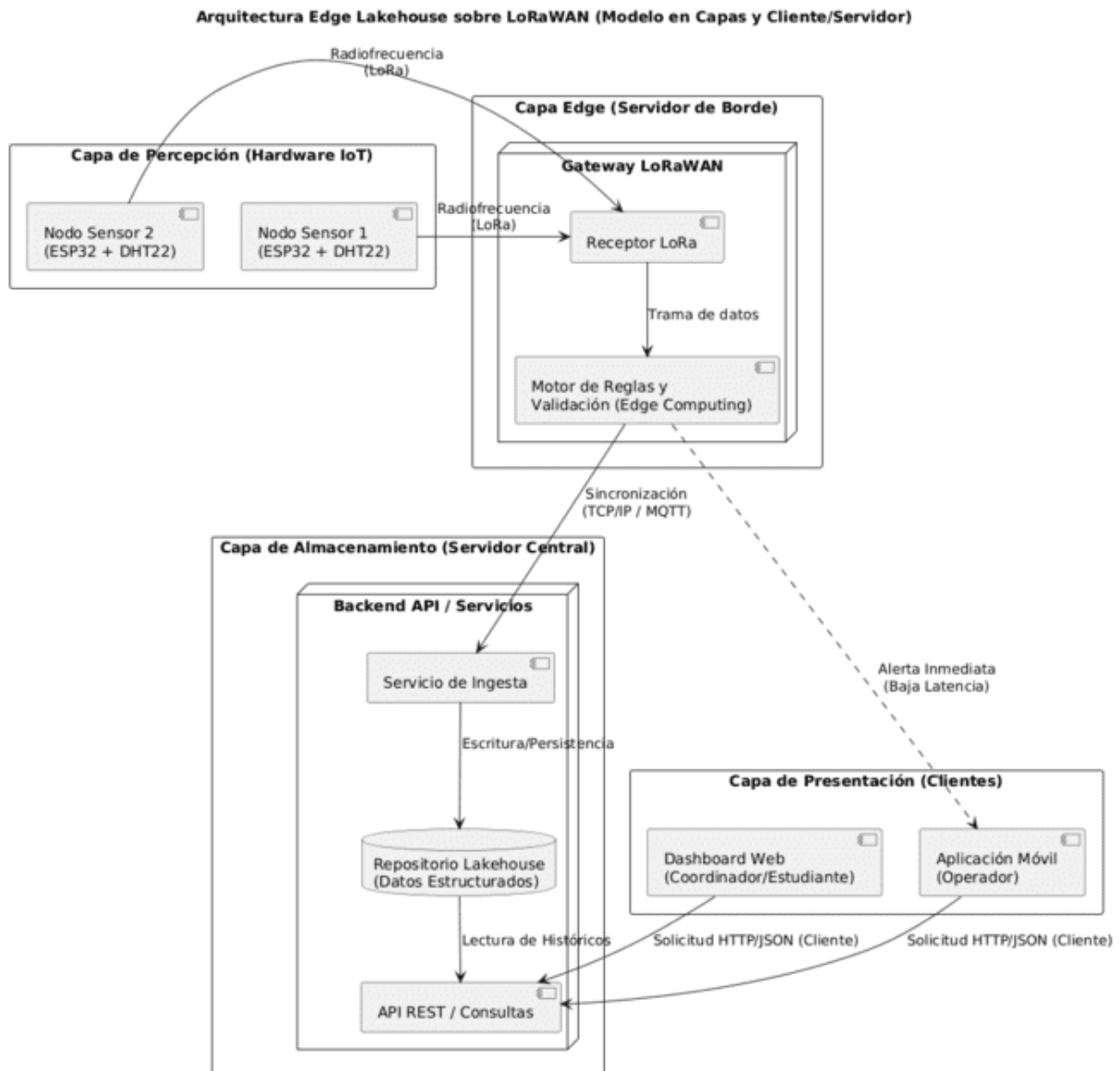
Se divide en las siguientes capas:

- Capa de Percepción (Dispositivos IoT): Se compone por nodos sensores desplegados en el vivero. Estos sensores actúan como emisores unidireccionales que capturan la

temperatura y humedad, transmitiendo los paquetes de datos a través de radiofrecuencia (LoRa) sin requerir conexión a internet.

- Capa de Retransmisión (Gateway): Representada por el Gateway ESP32, que capta las tramas que la estación envía vía LoRa y las reenvía al servidor y a un dispositivo móvil. El Gateway no descompone ni altera el contenido de los paquetes porque estos son cifrados; su trabajo es funcionar como un puente de retransmisión entre la red de sensores y las capas más altas del sistema.
- Capa de Almacenamiento y Servicios (Servidor Backend / Lakehouse): Cumple la función de servidor central en el sistema. Implementa la base de datos Lakehouse, la cual reúne los datos estructurados y ofrece una API RESTful que presenta la información del presente y del pasado de forma segura.
- Capa de Presentación (Cliente Móvil / Web): Corresponde a las interfaces gráficas utilizadas por los coordinadores y estudiantes (aplicación móvil o dashboard). Actúan como clientes ligeros que consumen los servicios del servidor central para renderizar gráficas, reportes históricos y la bitácora digital.

Ilustración 11: Arquitectura General del Sistema (Arquitectura Edge Lakehouse sobre LoRaWAN)



4.4.4.2 Tecnologías utilizadas

Tabla 13: Tecnologías utilizadas

Categoría	Herramienta / Tecnología	Descripción y uso dentro del proyecto
Entornos de Desarrollo (IDE)	Visual Studio Code + PlatformIO	Entorno empleado para programar, compilar e instalar el firmware en el microcontrolador ESP32 de los nodos sensores y el Gateway, utilizando la extensión PlatformIO con Visual Studio Code.
Entornos de Desarrollo (IDE)	Visual Studio Code	IDE primordial empleado para la programación y emulación del avance de la aplicación móvil cliente.
Lenguajes de Programación	C++ (Arduino Framework)	Lenguaje del microcontrolador ESP32, utilizado para la lógica de captura, el manejo de bytes y la transmisión por radiofrecuencia (LoRa).
Lenguajes de Programación	Dart	Lenguaje de programación orientado a objetos que se emplea para diseñar la aplicación móvil en términos lógicos y estructurales.
Frameworks Móviles	Flutter	Paquete de desarrollo (SDK) usado para el diseño de la interfaz del cliente móvil, la cual se encarga de mostrar la bitácora digital y las alertas.
Base de Datos Móvil	SQLite	Base de datos relacional ligero que se incorpora en la aplicación móvil de manera nativa para asegurar el almacenamiento de las variables ambientales permitiendo las consultas aun sin conexión a internet.

Categoría	Herramienta / Tecnología	Descripción y uso dentro del proyecto
Librerías de Comunicación	HardwareSerial.h/Bluetooth Clásico (SPP)	Se emplean bibliotecas en el microcontrolador (ESP32 Gateway) para establecer la comunicación a través de radiofrecuencia (LoRa), así como la retransmisión local sin hilos hacia el dispositivo celular por medio de Bluetooth clásico (SPP).

4.4.4.2.1 Adaptación tecnológica al entorno del proyecto

El título de este proyecto sugiere una estructura Edge Lakehouse basada en LoRaWAN. Estos conceptos definen la perspectiva conceptual y las tareas que la arquitectura debe llevar a cabo; su implementación se ajustó a las condiciones, los recursos y el ámbito específico de un entorno académico universitario. Cada idea del título se refiere a una aplicación específica que mantiene su función, pero se adapta a la escala del proyecto: LoRaWAN se manifiesta en una conexión LoRa punto a punto a través de módulos Reyax y comandos AT, que tiene la misma función de transmitir información a larga distancia; Lakehouse, en un almacenamiento centralizado construido con PostgreSQL en el servidor; y Edge, en el preprocesamiento y empaquetado realizados por los sensores ESP32. Por lo tanto, los términos del título deben ser interpretados como la descripción funcional de cada capa y no como la obligación de utilizar cada tecnología en su versión estándar al pie de la letra.

4.4.5 Roles Scrum

Tabla 14: Roles Scrum

Rol Scrum	Participante(s)	Función dentro del Proyecto (Enfoque en el Módulo)
Product Owner	Representante del Vivero Lastenia Vera	Actúa como la voz del cliente final.

Rol Scrum	Participante(s)	Función dentro del Proyecto (Enfoque en el Módulo)
	• Director del Proyecto Integrador	
Scrum Master	• Tutor Académico de Titulación	• Garantiza el cumplimiento de la metodología ágil y los estándares de ingeniería de software. • Remueve los impedimentos metodológicos o arquitectónicos que bloqueen el avance del desarrollo.
Scrum Team	Anderson Steven Plua Chamorro	• Responsable autogestionado de construir el incremento de software planificado. • Diseña e implementa la lógica de la arquitectura base de datos móvil local en el cliente móvil. • Desarrolla la interfaz gráfica en Flutter, la ingesta de datos mediante Bluetooth Clásico (SPP), el almacenamiento relacional offline en SQLite y el servicio de cola de reenvío hacia el servidor (store-and-forward).

4.4.6 Definición de Hecho (DoD)

4.4.6.1 Criterios Generales

Los criterios generales de la Definición de Hecho (DoD) constituyen el conjunto de requisitos transversales de calidad técnica y metodológica que deben verificarse de forma obligatoria en cualquier componente o incremento de software desarrollado antes de que este sea marcado como finalizado.

Una funcionalidad tiene que satisfacer las siguiente lista de requerimientos para ser entregada:

- **Compilación sin errores:** La compilación del código creado tiene que ser sin errores, ni fallos en la ejecución, ni advertencias críticas que pongan en peligro la estabilidad del sistema.
- **Concordancia con Historias de Usuario:** Para asegurar el comportamiento esperado del usuario final se debe establecer que la funcionalidad implementada este establecida en su respectiva historia de usuario, satisfaciendo sus requisitos específicos y alcanzando el éxito de aceptación.
- **Compatibilidad e integración:** Se debe garantizar que el componente desarrollado sea capaz de acoplarse a los demás módulos del sistema global de manera correcta y transparente, como también que la comunicación e intercambios entre los mismos no genere conflictos, que afecten el rendimiento del ecosistema de software
- **Consistencia y calidad visual:** Un diseño uniforme y consistente debe de ser características fundamentales de la interfaz gráfica que se expone al usuario, como también debe de estar libre de errores, renderizados, desbordamientos que afecten a la experiencia y usabilidad de los dispositivos visuales.

4.4.6.2 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 almacenamiento, procesamiento en el borde y visualización móvil de este módulo.

Para que una función específica del sistema sea considerada apta y terminada, debe cumplir obligatoriamente con las siguientes validaciones de software:

- **Persistencia cronológica en el Lakehouse Móvil:** El almacenamiento local en SQLite debe indexar obligatoriamente cada registro ambiental junto con su identificador único, impidiendo la inserción de registros duplicados o corruptos.
- **Evaluación y emisión de alertas:** Verificación de alertas: los datos que se almacenen en la base de datos local una vez que se hayan leído, estos deben de validar según los umbrales establecidos, y cuando estos valores sobrepasen los límites establecidos de humedad o temperatura se dispara una alerta avisando al usuario.
- **Persistencia y consulta sin conexión en la app:** Aun si el dispositivo móvil no tenga conexión a internet la actualización y registro se deben hacer localmente mediante base

de datos local SQLite, permitiendo el acceso continuo a indicadores clave y a la bitácora digital que necesite el usuario

- Limitación de vistas según el Rol Scrum: El módulo de presentación visual debe delimitar el acceso a la información dependiendo del usuario autenticado. Por ejemplo, los alumnos solo podrán ver los informes históricos de investigación, pero el coordinador técnico tendrá un acceso exclusivo a la configuración de los límites críticos de las alertas fitosanitarias.
- Trazabilidad y auditoria en el registro de la bitácora: Los cambios de estado o sucesos operativos del sistema tienen que ser anotados en la bitácora digital automatizada de manera cronológica, garantizando que el linaje de los datos climáticos sea totalmente rastreable desde su comienzo en el nodo sensor hasta su almacenaje final.

4.4.7 Product Backlog

Tabla 15: Product Backlog

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Configurar la comunicación UART y transmisión de datos entre el microcontrolador ESP32 y el módulo LoRa.	HU01	Alta	Por hacer
2	Configurar el servicio de transmisión inalámbrica local vía Bluetooth en el Gateway para el envío de datos hacia el dispositivo móvil.	HU01	Alta	Por hacer
3	Diseñar la interfaz gráfica principal (Dashboard) en la aplicación móvil para visualizar temperatura y humedad en tiempo real.	HU01	Alta	Por hacer
4	Programar el motor de reglas en el servidor de borde (Funcionalidad del macro-proyecto provista por el Módulo de Backend).	HU02	Alta	Por hacer
5	Integrar un servicio de notificaciones push en la aplicación móvil (Flutter) para la recepción de alertas tempranas.	HU02	Alta	Por hacer

Nro.	Tarea	Historia de usuario	Prioridad	Estado
6	Desarrollar la API REST (Backend) con endpoints (Funcionalidad del macro-proyecto provista por el Módulo de Backend).	HU03	Media	Por hacer
7	Crear el módulo de interfaz para que el administrador pueda registrar, editar o dar de baja direcciones MAC de nuevos nodos sensores.	HU04	Media	Por hacer
8	Implementar la base de datos móvil relacional (SQLite) para garantizar el almacenamiento offline (Desarrollo directo del presente módulo).	HU05	Media	Por hacer
9	Diseñar la vista de lista cronológica (Bitácora Digital) en la aplicación móvil que consuma los datos sincronizados de SQLite.	HU05	Baja	Por hacer
10	Implementar el mecanismo de sincronización y cola de retransmisión local (Store-and-Forward) en SQLite para tolerar cortes de red.	HU03	Alta	Por hacer
11	Ejecutar pruebas de campo in situ evaluando el rendimiento de la aplicación móvil (batería, memoria y alertas) y generar el empaquetado final (.APK).	Transversa 1	Alta	Por hacer

4.4.8 Sprint 1: Implementación de la Capa de Percepción y Enlace LoRaWAN

Para que la base de datos móvil pueda ingestar y almacenar información veraz, es necesario describir el origen del flujo de datos. Durante este primer Sprint, el equipo de hardware del proyecto integrador ejecutó la configuración del microcontrolador ESP32 y los sensores como el DHT22, el enlace se implementó como LoRa punto a punto. El rol del presente módulo (Base de datos móvil) en esta fase fue estrictamente consultivo, asegurando que la trama de

datos generada por la capa física estuviera correctamente formateada para su futura inserción en las tablas relacionales de SQLite

4.4.8.1 Planificación del Sprint

Tabla 16: Planificación del Sprint 1.

SPRINT 1: Configuración de Hardware IoT y Comunicación en entre Edge			
Duración:	2 semanas (06/01/2026 – 12/01/2026)	Prioridad:	Alta
Objetivo:	Configurar los microcontroladores ESP32 junto con los sensores y módulos LoRa para capturar, transmitir los datos climáticos de forma inalámbrica hasta el Gateway Edge, garantizando la recepción íntegra de los paquetes.		
Historias de usuario:	HU01: Visualización de variables climáticas en tiempo real		
Tareas de desarrollo:	• Instalar Visual Studio Code junto con PlatformIO y librerías. • Montaje los módulos LoRa RYLR998 y sensores. • Codificación en lenguaje C++ para la recolección y empaquetado de datos. • Configuración de recibimiento de tramas LoRa en el Gateway • Pruebas integrales de comunicación inalámbrica y rectificación de errores.		
Plan de entrega:	• Establecimiento del entorno y las bibliotecas. • Instalación de PlatformIO en Visual Studio Code para definir el dispositivo ESP32 junto con las librerías que necesiten (DHT sensor y HardwareSerial) • Tiempo de desarrollo: 4 horas • Fecha de Inicio y Fecha Final: 06/01/2026 - 06/01/2026. • Montaje y configuración de hardware (LoRa y sensores). • Conectar el módulo LoRa y el sensor DHT22 a los pines UART pertinentes del ESP32. • Tiempo de desarrollo: 6 horas • Fecha de Inicio y Fecha Final: 07/01/2026 - 07/01/2026. • Programación de la transmisión LoRa. • Crear un código en C++ que lea la humedad y la temperatura y lo transmita al módulo LoRa a través del puerto serie utilizando comandos AT • Tiempo de desarrollo: 12 horas • Fecha de Inicio y Fecha Final: 08/01/2026 - 18/01/2026. • Configuración del Gateway Edge receptor. • Configurar el nodo receptor para que reciba los paquetes LoRa entrantes y compruebe la integridad de la información. • Tiempo de desarrollo: 10 horas. • Fecha de Inicio y Fecha Final: 08/01/2026 - 08/01/2026. • Pruebas de comunicación y modificaciones finales. • Validar la estabilidad entre el ESP32 y el Gateway, solucionar problemas de pérdida de paquetes y comprobar el envío constante de datos que simula las condiciones del vivero.		

SPRINT 1: Configuración de Hardware IoT y Comunicación en entre Edge	
	• Tiempo de desarrollo: 16 horas • Fecha de Inicio y Fecha Final: 09/01/2026 - 12/01/2026.

4.4.8.2 Sprint Backlog

Tabla 17: Sprint Backlog 1.

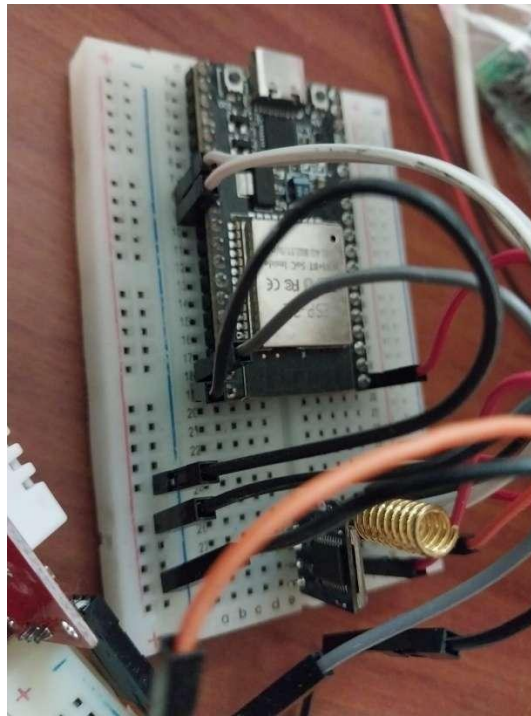
Historia de usuario.	Tarea	Duración estimada
HU01	Instalación del entorno Visual Studio Code + PlatformIO y configuración de librerías (DHT, HardwareSerial)	4 horas
HU01	Ensamblaje físico de sensores DHT22 y módulos LoRa RYLR998 al ESP32	6 horas
HU01	Programación en C++ para lectura, filtrado estadístico y empaquetado de datos	12 horas
HU01	Configuración del Gateway Edge para recepción de tramas LoRa y servicio Bluetooth	10 horas y media
HU01	Pruebas de comunicación inalámbrica LoRa, validación de integridad y corrección de errores	16 horas

4.4.8.3 Desarrollo del Sprint

4.4.8.3.1 Funcionalidad de captura y transmisión de datos vía LoRaWAN

En esta funcionalidad se desarrolló el módulo de percepción y comunicación inalámbrica del sistema. Consiste en la programación del microcontrolador ESP32 para que interactúe con el sensor DHT22, capture las lecturas precisas de temperatura atmosférica y humedad, y estructure estos datos en un payload cifrado. Posteriormente, el microcontrolador se comunica mediante el puerto serial (UART) con el módulo de radio RYLR998 para transmitir este paquete de datos a larga distancia hacia el Gateway (Edge) utilizando el protocolo LoRaWAN, garantizando que la información llegue de forma íntegra y segura sin depender de una red Wi-Fi local.

Ilustración 12: Ensamblaje físico sensor



- Explicación de la fotografía: En la imagen superior se observa el ensamblaje físico del nodo sensor. Se detalla la conexión de los pines de datos del sensor DHT22 al pin analógico/digital del ESP32, así como la conexión de los pines TX/RX del módulo LoRa RYLR998 a los puertos UART del microcontrolador. Este ensamblaje constituye la capa física encargada de recopilar las variables climáticas del Vivero.

Ilustración 13: Monitor Serial de PlatformIO en Visual Studio Code

```
PS C:\Users\Usuario\Desktop\Sincronizacion de datos\PlatformIO\Projects\Wodo_Meteorologico_Lastenia> C:\Users\Usuario\.platformio\penv\Scripts\platformio.exe device monitor --port COM11 --baud 115200
>>> BAUDRATE DETECTADO: 115200 <<<
[LoRa] +OK
[LoRa] +OK
[LoRa] +OK
[LoRa] Error modulo: +ERR=4
[LoRa] ADVERTENCIA: PARAMETER fallo – TX seguira en SF12 (lento).
[LoRa] +OK
[LoRa] +CRFOP=5
[LoRa] CRFOP=5 dñm. Si AT+SEND da +READY, soldar 470uF en VCC/GND del Reyax.
Listo. Ciclo continuo cada 15 minutos.
Medicion – sequence pendiente #231 | millis=2324
Buffer insuficiente para tendencia.
[Estado] Tendencia: sin_datos | Anomalia: no | sequence: 231
[Payload cifrado] vRV5sxmPLxetUuLDRiNDUxQDewpu9n2ubmPZZ-yrt71QrhZmYZC8BvLaSHNdk9jvW456ZCpQFBBFT7JVCigc0IX78S6vZFEIefvT7MbUQ
[Payload] 106 bytes
[LoRa] +OK
[LoRa] Enviando AT+SEND (30000 ms timeout)...
[LoRa] +OK
[LoRa] Transmission confirmada.
Ciclo completado. Proxima medicion en 15 min.
Medicion – sequence pendiente #232 | millis=38508
Buffer insuficiente para tendencia.
[Estado] Tendencia: estable | Anomalia: no | sequence: 232
[Payload cifrado] ewpSIDUH99VsxQA_0McYIHta2cv_uLLSo7BgcbY7gE9fIyowfeGT7vpque16a9AzZo5FmZ4d1GzU6vRCqz0UESaoLZAYssFssBYLn5XA
[Payload] 106 bytes
[LoRa] +OK
[LoRa] Enviando AT+SEND (30000 ms timeout)...
[LoRa] +OK
[LoRa] Transmission confirmada.
Ciclo completado. Proxima medicion en 15 min.
Medicion – sequence pendiente #233 | millis=74691
[Tendencia] Temperatura estable.
[Tendencia] Humedad estable.
[Estado] Tendencia: estable | Anomalia: no | sequence: 233
[Payload cifrado] BRjT7e5ujqmsqRij88KP7V_Igj3sM_cIQlf3MA8RjufGwRRIBom8YIzRG2Adk8IBmRLKBwj2AB4CBt5i-VE8XAL-kIf7Zyx0-LxrAQ
[Payload] 106 bytes
[LoRa] +OK
```

Mediante la interfaz del Monitor Serial de PlatformIO (integrado en Visual Studio Code), es posible auditar la ejecución del firmware en tiempo real. La rutina operativa del dispositivo inicia al fijar la tasa de baudios, seguida de manera inmediata por el arranque del transceptor LoRa y la toma de la lectura ambiental, la cual queda indexada bajo un número de secuencia único. En la salida de la consola, cada evento de registro va acompañado de la etiqueta "Payload cifrado" junto con la respectiva secuencia de bytes encriptada. Acto seguido, el visor muestra la ejecución del comando AT+SEND para validar la emisión de la trama por radiofrecuencia; esta comprobación permite constatar empíricamente que el nodo sensor captura, ofusca y envía las mediciones sin comprometer la fiabilidad del ciclo operativo.

- **Desarrollo del Código Fuente:**

El fragmento a continuación describe el algoritmo de preprocesamiento programado en C++ que opera dentro del ESP32. La meta de esta rutina es consolidar las variables físicas medidas en un único paquete de datos para luego aplicar un cifrado local antes de emitir cualquier señal. Al incorporar la barrera de seguridad directamente en el microcontrolador, se descarta el riesgo de exposición de los registros al momento de abandonar el dispositivo, protegiendo así el enlace de red desde el propio punto de origen.

Ilustración 14: Función preparePayload

```
// Construye el texto plano, lo cifra y devuelve solo el blob base64 para LoRa
static String preparePayload(const SensorData& data, uint32_t sequence) {
    static const char* DEVICE_LORA_ID = "esp32-01";

    char plain[128];
    snprintf(plain, sizeof(plain),
             "{\\"sequence\\":%lu,\\"temperature\\":%.1f,\\"humidity\\":%.1f}",
             static_cast<unsigned long>(sequence),
             data.temperature,
             data.humidity);

    return securityManager.encryptPayloadOnly(String(plain), DEVICE_LORA_ID);
}
```

La función **preparePayload**, ejecutada en el nodo sensor, cumple dos tareas fundamentales antes de entregar el dato a la capa de transmisión:

- **Estructuración del dato:** La función recibe las lecturas de humedad y temperatura recogidas por el sensor y las dispone junto a un número secuencial en una cadena de texto con formato estructurado. La identificación de cada lectura se logra a través del número de secuencia, el cual es fundamental para detectar posteriormente duplicados en las capas más altas del sistema.
- **Cifrado en el origen:** cuando el mensaje se ha estructurado, la función utiliza `encryptPayloadOnly` para invocar al componente de seguridad del sistema. Este cifra el contenido y devuelve la información protegida, lista para ser transmitida a través de LoRa. Así, el dato deja el nodo sensor ya cifrado y no hay ningún elemento intermedio, ni la aplicación móvil ni el gateway, que tenga la capacidad de descifrarlo. El servidor, que tiene la llave, es el único responsable de su descifrado. Este procedimiento es el que sostiene la integridad y confidencialidad de la información durante todo el sistema.

4.4.8.4 Sprint Review

Durante este primer Sprint se completó con éxito la funcionalidad de captura y transmisión del dato en el nodo sensor. El sistema integrado permite que el microcontrolador ESP32 lea las variables de temperatura y humedad a través del sensor DHT22, estructure la información junto a un número de secuencia y la cifre en el propio dispositivo antes de transmitirla. Finalmente, el sistema transmite este payload ya cifrado hacia el Gateway utilizando el módulo de radio

LoRa RYLR998 mediante comandos AT por el puerto UART, estableciendo el enlace fundamental sobre el cual operan las capas superiores del sistema.

Ilustración 15: Respuesta del Monitor Serial de PlatformIO en Visual Studio Code

```
[Payload cifrado] vRV5xomPLXeUuLDRiNOuXpOewpu9n2ubmPZZ-yrt71QRhZnYZC8BvLaShnDk9jvAH56ZcMpOFBBfT7JVCigc8NIX7856vZFEIefvT7MbbuQ
[Payload] 106 bytes
[LoRa] +OK
[LoRa] Enviando AT+SEND (30000 ms timeout)...
[LoRa] +OK
[LoRa] Transmision confirmada.
Ciclo completado. Proxima medicion en 15 min.
Medicion - sequence pendiente #232 | millis=38508
Buffer insuficiente para tendencia.
[Estado] Tendencia: estable | Anomalia: no | sequence: 232
[Payload cifrado] ewpSIDUNh997VsxQA_QMcYIhta2cv_uLLSo7BgcbY7gE9flyowfeGT7vpquel6a9AzZoSFmZ4d1GzU6vRCqz0UESaoLZAYssFssBYLn5XA
[Payload] 106 bytes
[LoRa] +OK
[LoRa] Enviando AT+SEND (30000 ms timeout)...
[LoRa] +OK
[LoRa] Transmision confirmada.
Ciclo completado. Proxima medicion en 15 min.
Medicion - sequence pendiente #233 | millis=74691
[Tendencia] Temperatura estable.
[Tendencia] Humedad estable.
[Estado] Tendencia: estable | Anomalia: no | sequence: 233
[Payload cifrado] BRjT7e5ujqmsqRij88KP7V_IgJ3sM_cIQljfJMA8RjufGaRRibOmBYIzRG2Adk81bMRlKBwj2AB4C8t5i-vE8XAL-klfJzYx0-LxraDQ
[Payload] 106 bytes
[LoRa] +OK
[LoRa] Enviando AT+SEND (30000 ms timeout)...
[LoRa] +OK
[LoRa] Transmision confirmada.
Ciclo completado. Proxima medicion en 15 min.
Medicion - sequence pendiente #234 | millis=110874
[Tendencia] Temperatura estable.
[Tendencia] Humedad estable.
[Estado] Tendencia: estable | Anomalia: no | sequence: 234
[Payload cifrado] jRVk-OHAH03V0Si3k9CrzTuyk-yKDQ-ftso352of2tfID3406W0iz-clIOFrH6h8Fi2x-113ANDwCz_ejEyQgXhdk24j0rnAwAofMv1-Z0w
[Payload] 106 bytes
[LoRa] +OK
```

En la captura superior se evidencia la ejecución en tiempo real del firmware. Se observa en el log del sistema la correcta lectura de las variables climáticas, el proceso de cifrado del dato y la confirmación de la trama enviada vía LoRa. Como se aprecia en las líneas etiquetadas como "Payload cifrado", la información abandona el nodo ya protegido y resulta ilegible para cualquier componente intermedio, lo que demuestra que la capa de percepción física está operativa y comunicándose de forma segura.

- **Retroalimentación recibida**

Durante la reunión de revisión (Sprint Review) con el coordinador técnico y el equipo de proyecto, se evaluó el desempeño del prototipo en envío continuo y se detectaron las siguientes observaciones:

1. **Saturación del canal LoRa:** A lo largo de las pruebas, se observó que el módulo transmisor denegaba comandos de envío de manera intermitente al emitir la alerta

"Ocupado" (+ERR=12). Esta falla ocasionaba la pérdida inmediata de los paquetes climáticos, lo cual perjudicaba la constancia del flujo de datos dentro de la plataforma.

2. Rastreo de paquetes: Se descubrió una brecha importante ligada a los cortes de alimentación eléctrica. Si el microcontrolador experimentaba un reinicio forzado, el contador de secuencias borraba su progreso y regresaba a cero. Este comportamiento no solo desorganizaba el orden cronológico del historial guardado, sino que dejaba a la infraestructura de sensores vulnerable ante eventuales ataques de repetición (Replay Attack).

- **Mejoras realizadas**

Con el fin de resolver los comentarios planteados en la revisión, se llevaron a cabo las siguientes modificaciones directas en el código fuente del ESP32:

1. Aplicación lógica de reintentos (Tolerancia a fallos): La función `sendPayloadWithRetry` fue creada para analizar la respuesta del módulo LoRa. Si se identifica el error de saturación (+ERR=12), el sistema ahora implementa un periodo automático de espera de 400 milisegundos y vuelve a intentar la transmisión, disminuyendo considerablemente la pérdida de información climática.
2. Persistencia del número de secuencia: Para corregir esta vulnerabilidad, se programó la rutina `nextSequence()` apoyándose en la librería nativa `Preferences.h`. Dicha integración traslada el registro del contador de paquetes directamente hacia la memoria no volátil (NVS) del ESP32. Bajo este esquema operativo, el hardware recupera la numeración correlativa exacta luego de sufrir cualquier corte de energía, blindando la red contra vectores de ataque por repetición y garantizando un orden cronológico inalterable en el historial.

4.4.8.5 Pruebas del Sprint

4.4.8.5.1 Pruebas de Caja Negra (Validación de captura y estructuración del Payload)

Tabla 18: Pruebas de Caja Negra - Sprint 1

Nombre del campo	Tipo de campo	Valor permitido	Observación
Temperatura (temperatureDeciC)	Entero corto (int16_t)	Valores numéricos que representen entre -400 y 800 (Equivalente a -40.0°C a 80.0°C).	Permite comprimir el valor flotante multiplicándolo por 10. Rechaza valores nulos (isnan).
Humedad (humidityDeciPct)	Entero positivo (uint16_t)	Valores numéricos entre 0 y 1000 (Equivalente a 0.0% a 100.0%).	Se captura correctamente del DHT22 y se comprime. Ignora el envío si el sensor falla.
Número de Secuencia (seq)	Entero largo (uint32_t)	Números enteros positivos (> 0) en incremento constante.	Funciona de manera adecuada. Después de que el ESP32 se reinicia, el valor se restaura desde la memoria (NVS).
Trama a transmitir (payloadHex)	Cadena de texto (String)	Paquete encriptado y codificado en formato base64, preparado para emisión (Ej: arreglo ofuscado de 106 bytes).	El empaquetado y la encriptación de las lecturas se realizan dentro del propio nodo antes de iniciar el envío. De este modo, se garantiza que ningún equipo intermediario logre interpretar la información.
Comando de envío LoRa (AT+SEND)	Cadena de texto (String)	Formato estricto: AT+SEND=[Address],[Length],[Payload]\r\n	El código del sistema comprueba primero el tamaño de la cadena de datos antes de enviar la instrucción al módulo LoRa utilizando el puerto UART.

4.4.8.5.2 Pruebas de Caja Blanca (Lógica de tolerancia a fallos y persistencia NVS)

Tabla 19: Pruebas de Caja Blanca - Sprint 1

Método	Acción esperada	Acción obtenida	Observación
Condicional de lectura del sensor (isnan)	Detener la ejecución del bloque principal en caso de que el DHT22 registre datos nulos (NaN) ante una pérdida física de conexión.	El microcontrolador forzaba el procesamiento de las lecturas vacías, lo que desestabilizaba por completo la construcción del paquete final.	La comprobación inicial del sensor resultaba ineficaz. Para solucionarlo, se insertó la sentencia return; junto a un delay(2000), logrando interrumpir el ciclo y dar tiempo a que el componente se estabilice.
Bucle de reintentos LoRa (sendPayloadWithRetry)	Identificar la respuesta "+ERR=12" correspondiente a canal ocupado, aplicar un retardo de 400ms y lanzar nuevamente el ciclo	La subrutina finalizaba prematuramente al retornar false durante la primera evaluación, impidiendo cualquier intento de	El fallo residía en la parametrización de la condición attempt < 2. Al ajustar este límite matemático, el algoritmo ahora verifica el estado de la red y repite la transmisión adecuadamente.
Método de persistencia NVS (nextSequence)	Extraer el último índice guardado en la memoria permanente, incrementar su valor en una unidad y guardar la actualización mediante el método putULong.	Al someter la placa ESP32 a un reinicio manual, la numeración de los envíos no se interrumpió y avanzó al dígito consecutivo exacto (por ejemplo, del 685 al 686).	Resultado exitoso. Queda demostrado a nivel de software que el dispositivo soporta cortes repentinos de energía sin perder el rastreo cronológico.

4.4.8.6 Incremento y Entregables del Sprint

Al finalizar este primer Sprint enfocado en la capa de percepción y comunicación en el borde, se obtuvo un prototipo físico y lógico completamente funcional. A continuación, se detallan los procesos y funciones que se completaron satisfactoriamente de acuerdo con la planificación inicial:

- Se completó el ensamblaje físico del hardware: Integración exitosa del microcontrolador ESP32 con el sensor de temperatura/humedad DHT22 y el módulo de radiofrecuencia LoRa RYLR998.
- Se desarrolló la lógica para recolectar los datos: Programación en C++ para leer adecuadamente las variables climáticas y empaquetarlas de manera optimizada en binario dentro del microcontrolador.
- Se desarrolló la transmisión inalámbrica tolerante a fallos: Para reducir la saturación del canal, se establece la comunicación LoRaWAN a través de comandos AT (UART) e incorporando una función de reintentos automáticos.
- Se configuró el Gateway Edge receptor: El nodo receptor fue habilitado para recibir y comprobar las tramas de datos que los sensores del vivero envían a través de largas distancias.

4.4.8.7 Estado actual del Product Backlog

Tabla 20: Estado actual del Product Backlog - Sprint 1

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Configurar la comunicación UART y transmisión de datos entre el microcontrolador ESP32 y el módulo LoRa.	HU01	Alta	Completada
2	Configurar el servicio de transmisión inalámbrica local vía Bluetooth en el Gateway para el envío de datos hacia el dispositivo móvil.	HU01	Alta	Completada
3	Diseñar la interfaz gráfica principal (Dashboard) en la aplicación móvil para	HU01	Alta	Por hacer

Nro.	Tarea	Historia de usuario	Prioridad	Estado
	visualizar temperatura y humedad en tiempo real.			
4	Programar el motor de reglas en el servidor de borde (Funcionalidad del macro-proyecto provista por el Módulo de Backend).	HU02	Alta	Por hacer
5	Integrar un servicio de notificaciones push en la aplicación móvil (Flutter) para la recepción de alertas tempranas.	HU02	Alta	Por hacer
6	Desarrollar la API REST (Backend) con endpoints (Funcionalidad del macro-proyecto provista por el Módulo de Backend).	HU03	Media	Por hacer
7	Crear el módulo de interfaz para que el administrador pueda registrar, editar o dar de baja direcciones MAC de nuevos nodos sensores.	HU04	Media	Por hacer
8	Implementar la base de datos móvil relacional (SQLite) para garantizar el almacenamiento offline (Desarrollo directo del presente módulo).	HU05	Media	Por hacer
9	Diseñar la vista de lista cronológica (Bitácora Digital) en la aplicación móvil que consuma los datos sincronizados de SQLite.	HU05	Baja	Por hacer
10	Implementar el mecanismo de sincronización y cola de retransmisión local (Store-and-Forward) en SQLite para tolerar cortes de red.	HU03	Alta	Por hacer
11	Ejecutar pruebas de campo in situ evaluando el rendimiento de la aplicación móvil (batería, memoria y alertas) y generar el empaquetado final (.APK).	Transversa 1	Alta	Por hacer

4.4.9 Sprint 2: Desarrollo de la Aplicación Móvil y Base de Datos Local (SQLite)

El uso de SQLite en el dispositivo móvil no solo sirve como almacenamiento temporal, sino también como un nodo esencial en la estructura de Distributed Edge Computing. El sistema logra una funcionalidad Offline-First cuando el almacén de datos (el Warehouse) se transfiere directamente al equipo del agricultor. Esto implica que la verificación de la bitácora y el análisis de datos climáticos históricos se llevan a cabo a través del procesamiento en el borde (directamente en el procesador del teléfono móvil), lo cual elimina cualquier latencia de la red y asegura que los datos ambientales puedan ser gestionados en lugares con conectividad nula, como es el caso del Vivero Lastenia.

4.4.9.1 Planificación del Sprint

Tabla 21: Planificación del Sprint 2.

SPRINT 2: Implementación del Cliente Móvil (Flutter) y Almacenamiento Offline (SQLite)			
Duración:	2 semanas (13/01/2026 – 27/01/2026)	Prioridad:	Alta
Objetivo:	Desarrollar la aplicación móvil cliente integrando un motor de base de datos relacional local (SQLite) que garantice el almacenamiento estructurado y la visualización en tiempo real e histórica (Bitácora) de las variables climáticas sin depender de conexión a internet.		
Historias de usuario:	HU01: Visualización de variables climáticas en tiempo real. HU05: Registro automatizado de métricas meteorológicas		
Tareas de desarrollo:	- Preparación del ecosistema de desarrollo en Flutter, incluyendo la vinculación de paquetes clave como sqflite y path_provider. - Construcción nativa del repositorio local empleando SQLite (correspondiente a la Tarea 9). - Elaboración del panel de control o Dashboard principal (Tarea 3), a la par de la maquetación del historial cronológico denominado Bitácora Digital (Tarea 10).		
Plan de entrega:	Estructuración del motor SQLite: - Construcción de esquemas, tablas y rutinas de persistencia (operaciones CRUD) destinadas a guardar la telemetría en el dispositivo. - Tiempo de desarrollo: 14 horas - Fecha de Inicio y Fecha Final: 13/01/2026 - 15/01/2026. Maquetación del Dashboard: - Creación de las pantallas en Flutter encargadas de proyectar en vivo los niveles de humedad y temperatura.		

SPRINT 2: Implementación del Cliente Móvil (Flutter) y Almacenamiento Offline (SQLite)	
	• Tiempo de desarrollo: 12 horas • Fecha de Inicio y Fecha Final: 16/01/2026 - 19/01/2026. • Desarrollo de la Bitácora Digital: • Programación del módulo que despliega el historial cronológico recuperando la información aislada (offline) de la base de datos local. • Tiempo de desarrollo: 10 horas • Fecha de Inicio y Fecha Final: 20/01/2026 - 23/01/2026. • Auditoría visual y de persistencia: • Certificar el correcto guardado de métricas sin conexión a la red y a pulir defectos en la interfaz. • Tiempo de desarrollo: 8 horas. • Fecha de Inicio y Fecha Final: 24/01/2026 - 27/01/2026.

4.4.9.2 Sprint Backlog

Tabla 22: Sprint Backlog 2

Historia de usuario.	Tarea	Duración estimada
HU05	Integración del motor relacional nativo (SQLite) con el propósito de asegurar la retención de las métricas en entornos desconectados.	14 horas
HU01	Maquetación y desarrollo del panel de control frontal (Dashboard) destinado a proyectar las lecturas de humedad y temperatura.	12 horas
HU05	Programación del módulo visual para la Bitácora Digital, configurado para recuperar y renderizar la información persistida internamente en SQLite.	10 horas
HU01/ HU05	Ciclo de auditoría enfocado en validar el comportamiento local del sistema y aplicar parches de resolución sobre la interfaz (UI).	8 horas

4.4.9.3 Desarrollo del Sprint

4.4.9.3.1 Funcionalidad de base de datos móvil (SQLite) e Interfaz en Flutter

El esquema emplea la tabla `pending_uploads` para retener temporalmente la trama cifrada proveniente de la conexión Bluetooth en su estado original, asegurando su disponibilidad para el envío hacia el servidor. En contraste, la tabla `readings` se destina a guardar los valores ambientales (temperatura y humedad) en formato legible, los cuales se obtienen mediante la API REST una vez que el backend los ha descifrado y validado. Mediante esta estrategia, el aplicativo logra resguardar el dato crudo sin requerir su interpretación inmediata, garantizando que en la bitácora se refleje exclusivamente la información que ya completó el procesamiento en el servidor.

En esta funcionalidad se desarrolló la capa de presentación y almacenamiento en el dispositivo cliente. Utilizando el SDK de Flutter y el lenguaje Dart, se construyeron las interfaces gráficas correspondientes al Dashboard y la Bitácora Digital. Para garantizar el cumplimiento de la arquitectura local (Edge), se integró la librería `sqflite`, permitiendo que la aplicación intercepte los datos sin procesar y los organice bajo un esquema relacional dentro de un archivo local (`.db`), asegurando la operatividad total en zonas del vivero que carezcan de conexión a internet.

- **Desarrollo del Código Fuente:**

A continuación, se detalla el núcleo del gestor de la base de datos local y los métodos asíncronos para la creación de tablas, inserción y lectura de las series temporales capturadas en el dispositivo móvil:

Ilustración 16: Base de datos Local SQLite

```
Future<void> _createSchema(DatabaseExecutor db) async {  
  await db.execute('''  
    CREATE TABLE sensors (  
      id INTEGER PRIMARY KEY AUTOINCREMENT,  
      external_id TEXT NOT NULL UNIQUE,  
      name TEXT NOT NULL,  
      last_seen_at TEXT  
    )  
  ''');  
  
  await db.execute('''  
    CREATE TABLE readings (  
      id INTEGER PRIMARY KEY AUTOINCREMENT,  
      sensor_id INTEGER NOT NULL,  
      sequence INTEGER NOT NULL,  
      temperature REAL NOT NULL,  
      humidity REAL NOT NULL,  
      rssi INTEGER NOT NULL,  
      sensor_timestamp_seconds INTEGER,  
      gateway_timestamp TEXT,  
      received_at_device TEXT NOT NULL,  
      sync_status TEXT NOT NULL,  
      source TEXT NOT NULL,  
      FOREIGN KEY(sensor_id) REFERENCES sensors(id),  
      UNIQUE(sensor_id, sequence)  
    )  
  ''');
```

En la ejecución de la siguiente función se definen las entidades interrelacionadas: sensors, con la finalidad de identificar el nodo empleado en el campo mediante el campo de external_id y readings, las cuales están direccionadas a guardar la telemetría que el sensor haya procesado. Para que las entidades se vinculen se emplea una clave foránea (FOREIGN KEY), asegurando que desde el propio motor de la base de datos que cada medición corresponda de manera exacta al sensor dispositivo que la genere. Además de evitar que los registros se dupliquen se aplicó una restricción compuesta UNIQUE(sensor_id, sequence). Gracias estructura planteada la base de datos SQLite descarta automáticamente cualquier intento de ingresar información redundante, lo cual hace posible preservar la integridad de los datos locales para su posterior revisión de consulta de datos históricos.

4.4.9.3.2 Funcionalidad de Ingesta Inalámbrica y Alojamiento del Paquete Recibido

A fin de facultar a la base de datos móvil para retener la información sin requerir acceso a internet, la aplicación cliente (Flutter) integra un módulo de recepción mediante Bluetooth Clásico (SPP). Dicho elemento actúa como el nexo de vinculación directa con el ESP32. Una

vez que el hardware de frontera capta las señales meteorológicas transmitidas por LoRa, se encarga de transferir la telemetría hacia el smartphone a través de este enlace Bluetooth. En esta etapa del flujo, el cliente móvil no dispone de las llaves criptográficas, dado que la arquitectura reserva la tarea de descifrado estrictamente para el servidor central. A raíz de esto, la plataforma evita evaluar el contenido de los mensajes y restringe su labor a la conservación segura del paquete.

- **Desarrollo del Código Fuente (Lógica de recepción y alojamiento en Dart):**

Al separar esta fase de guardado de cualquier intento de descifrado anticipado, se salvaguardan los registros climáticos obtenidos, lo que garantiza su retención inalterada en el dispositivo incluso cuando se carece por completo de conexión a la red.

Ilustración 17: Recepción por Bluetooth y alojamiento del paquete en la base local

```
void processBluetoothData(String rawData) {
  try {
    // Ejemplo de rawData esperado: "SEQ=1;T=333;H=612"
    List<String> parts = rawData.split(';');

    if (parts.length == 3) {
      // 1. Extracción de valores
      String seqStr = parts[0].replaceAll('SEQ=', '');
      String tempStr = parts[1].replaceAll('T=', '');
      String humStr = parts[2].replaceAll('H=', '');

      // 2. Transformación (Revirtiendo la compresión del Edge)
      double realTemperature = int.parse(tempStr) / 10.0;
      double realHumidity = int.parse(humStr) / 10.0;

      // 3. Empaquetado para SQLite
      Map<String, dynamic> sensorReading = {
        'device_id': 'Nodo-01',
        'timestamp': DateTime.now().toIso8601String(),
        'temperature': realTemperature,
        'humidity': realHumidity
      };

      // 4. Carga (Load) a la base de datos local
      DatabaseHelper.instance.insertReading(sensorReading);
      print("Dato guardado offline correctamente: $realTemperature °C");
    }
  } catch (e) {
    print("Error al parsear la trama Bluetooth: $e");
  }
}
```

El bloque de código presentado detalla el mecanismo de recepción ejecutado en el terminal móvil. Una vez interceptadas las señales por Bluetooth, la aplicación ensambla los fragmentos binarios hasta reconstruir el paquete completo y lo guarda directamente en el almacenamiento del dispositivo. En esta etapa se evita a propósito cualquier proceso de descifrado, ya que la interpretación de los datos corresponde únicamente al servidor. Para proteger la estabilidad del

software, la lógica de programación de estructura dentro de un bloque try-catch, de esta manera las excepciones derivadas de tramas erróneas se interceptan y evitando que la aplicación sufra de crasheos (App Crashes).

4.4.9.4 Sprint Review

Durante la revisión del Sprint 2, se demostró ante el Product Owner (representante del Vivero) que la aplicación móvil es capaz de almacenar y mostrar el historial climático sin necesidad de Wi-Fi o datos móviles.

4.4.9.4.1 Retroalimentación recibida:

- Durante la extracción masiva de registros históricos se identificó que las peticiones directas que se hacían a SQLite reducían el rendimiento gráfico, provocando que existan breves pausas en la interfaz de usuario.
- Las marcas de tiempo se guardaban y mostraban en la pantalla conservando la codificación numérica nativa de Unix, lo cual hacía que esta información no tuviera sentido práctico para que los usuarios (agricultores y estudiantes) del vivero las pudieran interpretar en campo.

4.4.9.4.2 Mejoras realizadas:

- Para resolver las interrupciones gráficas, las operaciones que pesaban se trasladaron como proceso secundario apoyándose en los Isolates y el componente FutureBuilder de las cuales son herramientas nativas de Flutter. Despejando esta carga de trabajo la interfaz queda despejada, garantizando que la navegación sea fluida.
- Se importó el paquete intl de Dart para intervenir los datos exclusivamente en la capa de vistas. Su función es capturar el valor Unix crudo y transformarlo sobre la marcha a una nomenclatura convencional (dd/MM/yyyy HH:mm) fracciones de segundos antes de construir la interfaz gráfica.

4.4.9.5 Pruebas del Sprint

4.4.9.5.1 Pruebas de Caja Negra (Interfaz y Base de Datos)

Tabla 23: Pruebas de Caja Negra - Sprint 2

Nombre de la Prueba	Condición de Entrada / Ejecución	Resultado Esperado	Observación
Inserción de registro offline	Recepción de un paquete cifrado por Bluetooth estando el dispositivo sin conexión a internet.	El paquete debe almacenarse íntegro en la cola local (pending_uploads) con estado pendiente, sin requerir conexión ni interpretación de su contenido.	Se valida que el alojamiento del paquete en la cola local (pending_uploads) no requiere peticiones HTTP externas ni interpretación de su contenido.
Consulta de Bitácora	El usuario (Estudiante) navega a la pestaña de "Historial" en la app.	Se muestra una lista renderizada en orden descendente con los datos.	La prueba es exitosa; la vista consume los datos locales respetando el ordenamiento por fecha (ORDER BY timestamp DESC).

4.4.9.5.2 Pruebas de Caja Blanca (Lógica Interna)

Tabla 24: Pruebas de Caja Blanca - Sprint 2

Método Evaluado	Acción esperada	Acción obtenida	Observación
Inicialización de Base de Datos (_createSchema)	Validar si el archivo .db existe. Si no, crearlo ejecutando el script SQL y asignando el path interno.	Generó la ruta interna del dispositivo y estructuró la tabla sin errores sintácticos.	Ejecución correcta. Se usó path_provider para evitar restricciones del sistema operativo en Android/iOS.

Método Evaluado	Acción esperada	Acción obtenida	Observación
Función getReadingsBetween	Extraer toda la tabla de lecturas en forma de una lista de mapas (List<Map>).	La función intentaba renderizar antes de que la consulta terminara.	Se corrigió obligando el uso de los operadores async/await en el controlador de la interfaz (UI).

4.4.9.6 Incremento y Entregables del Sprint

- Configuración del entorno móvil: Creación exitosa del proyecto en Flutter y acoplamiento de las librerías nativas necesarias.
- Almacenamiento local (SQLite): Implementando la tabla readings en la memoria persistente del dispositivo móvil se logro que se cumpla con las exigencias de un almacenamiento relacional.
- Desarrollo de Vistas (UI): Renderizado de las pantallas funcionales principales (Dashboard para monitoreo actual y Bitácora para historial).
- Autonomía de consulta: confirmación técnica de que la aplicación puede ordenar y mostrar las variables ambientales previamente almacenadas con el acceso a internet deshabilitado, y conservar los paquetes recibidos para su envío posterior.

4.4.9.7 Estado actual del Product Backlog

Tabla 25: Estado actual del Product Backlog - Sprint 2

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Configurar la comunicación UART y transmisión de datos entre el microcontrolador ESP32 y el módulo LoRa.	HU01	Alta	Completada
2	Configurar el servicio de transmisión inalámbrica local vía Bluetooth en el Gateway para el envío de datos hacia el dispositivo móvil.	HU01	Alta	Completada

Nro.	Tarea	Historia de usuario	Prioridad	Estado
3	Diseñar la interfaz gráfica principal (Dashboard) en la aplicación móvil para visualizar temperatura y humedad en tiempo real.	HU01	Alta	Completada
4	Programar el motor de reglas en el servidor de borde (Edge) para evaluar constantemente si los datos superan los umbrales críticos.	HU02	Alta	Por hacer
5	Integrar un servicio de notificaciones push en la aplicación móvil (Flutter) para la recepción de alertas tempranas.	HU02	Alta	Por hacer
6	Desarrollar la API REST (Backend) con endpoints para la consulta y exportación de datos históricos por rangos de fechas.	HU03	Media	Por hacer
7	Crear el módulo de interfaz para que el administrador pueda registrar, editar o dar de baja direcciones MAC de nuevos nodos sensores.	HU04	Media	Por hacer
8	Implementar la base de datos móvil relacional (SQLite) para garantizar el almacenamiento offline en el dispositivo del usuario.	HU05	Media	Completada
9	Diseñar la vista de lista cronológica (Bitácora Digital) en la aplicación móvil que consuma los datos sincronizados de SQLite.	HU05	Baja	Completada
10	Implementar el mecanismo de sincronización y cola de retransmisión local (Store-and-Forward) en SQLite para tolerar cortes de red.	HU03	Alta	Por hacer
11	Ejecutar pruebas de campo in situ evaluando el rendimiento de la aplicación móvil	Transversal	Alta	Por hacer

Nro.	Tarea	Historia de usuario	Prioridad	Estado
	(batería, memoria y alertas) y generar el empaquetado final (.APK).			

4.4.10 Sprint 3: Implementación de Alertas Locales (Edge Analytics)

4.4.10.1 Planificación del Sprint

Tabla 26: Planificación del Sprint 3.

SPRINT 3: Generación de alertas, visualización histórica y gestión de nodos			
Duración:	2 semanas (28/01/2026 - 10/02/2026)	Prioridad:	Alta
Objetivo:	Implementar el motor de evaluación de umbrales para emitir alertas preventivas locales, configurar el consumo de la API REST para la actualización de la bitácora histórica y desarrollar la vista del administrador para verificar el estado de conexión de los nodos.		
Historias de usuario:	HU02: Generación de alertas preventivas automáticas. HU03: Consulta de historial climático. HU04: Verificación de estado de sincronización local.		
Tareas de desarrollo:	- Codificación del núcleo analítico en el borde (Edge Analytics) para la ponderación de umbrales y la detonación de avisos en la capa de Flutter. - Implementación del cliente HTTP orientado a consumir y mapear la estructura JSON hacia el repositorio nativo en SQLite. - Construcción del panel de administración, enfocado en auditar la salud operativa de la red mediante el análisis de las marcas de tiempo (timestamp)		
Plan de entrega:	Motor local de alertas. - Codificación del script responsable de cruzar los registros de SQLite con los límites ambientales predefinidos, articulando su respuesta con los componentes de Flutter. - Tiempo de desarrollo: 10 horas - Fecha de Inicio y Fecha Final: 28/01/2026 - 31/01/2026. Integración de la API REST. - Ajuste del cliente HTTP sobre Dart para interceptar los paquetes JSON y transformar los datos crudos al esquema de la base local - Tiempo de desarrollo: 14 horas - Fecha de Inicio y Fecha Final: 01/02/2026 - 05/02/2026 Módulo de diagnóstico administrativo.		

SPRINT 3: Generación de alertas, visualización histórica y gestión de nodos	
	• Maquetación de la vista técnica que audita la antigüedad de los timestamps, permitiendo medir la latencia del sistema. • Tiempo de desarrollo: 8 horas • Fecha de Inicio y Fecha Final: 06/02/2026 - 08/02/2026. • Auditoría de UI e integración. • Sesiones de prueba diseñadas para inyectar datos anómalos, certificar los disparadores de advertencias y someter la aplicación a estrés. • Tiempo de desarrollo: 12 horas. • Fecha de Inicio y Fecha Final: 09/02/2026 - 10/02/2026

4.4.10.2 Sprint Backlog

Tabla 27: Sprint Backlog 3.

Historia de usuario.	Tarea	Duración estimada
HU02	Configuración del motor local de alertas: Programar el algoritmo de evaluación condicional sobre los datos ingresados e integrar notificaciones visuales en la interfaz.	10 horas
HU03	Sincronización mediante API REST: Parametrización del servicio HTTP nativo en Dart para capturar y decodificar el esquema JSON, mapeando los atributos resultantes directamente hacia las tablas relacionales de SQLite.	14 horas
HU04	Panel de diagnóstico administrativo: Construcción de una pantalla de auditoría capaz de evaluar la antigüedad del último timestamp, permitiéndole al coordinador monitorear posibles caídas o interrupciones de red en los dispositivos de campo.	8 horas
HU02, HU03, HU04	Ciclo de evaluación integral: Sometimiento de las interfaces gráficas a escenarios de latencia forzada e inyección de mediciones anómalas, abarcando la depuración inmediata de los fallos identificados en la aplicación.	12 horas

4.4.10.3 Desarrollo del Sprint

4.4.10.3.1 Funcionalidad de Alertas Locales (Edge Analytics)

En esta funcionalidad se desarrolló la funcionalidad de aviso por umbral, que dota a la aplicación de la capacidad de comparar cada lectura ya alojada en la base de datos local frente a valores límite definidos para el cultivo. Consiste en un algoritmo en Dart que, inmediatamente después de que un nuevo registro se persiste en SQLite, compara las variables de temperatura y humedad contra umbrales agronómicos establecidos para el cacao. Si los valores exceden los límites tolerables, se despliega un elemento gráfico en la interfaz de Flutter con el fin de advertir al usuario.

Ilustración 18: Funcionalidad de alertas por umbral



Explicación de la evidencia: En la captura superior se evidencia la ejecución en tiempo real del motor de alertas en la interfaz gráfica. Se observa un cuadro de diálogo emergente (Dialog) advirtiéndole al usuario sobre un riesgo térmico inminente, basándose exclusivamente en el procesamiento de los datos estructurados en la memoria del dispositivo.

- **Desarrollo del Código Fuente:**

Aquí se presenta el bloque lógico esencial implementado en Flutter para la evaluación local y detonación de la interfaz de advertencia:

Ilustración 19: Función lógica aviso por umbral

```
/// Evalua umbrales locales solo para lecturas recientes de gateway.
Future<void> _evaluateAlertRules(Reading reading) async {
  final isRecentGatewayReading =
    reading.gatewayTimestamp != null &&
    DateTime.now().difference(reading.gatewayTimestamp!).abs() <
    const Duration(minutes: 3);
  if (!isRecentGatewayReading) {
    return;
  }

  for (final rule in alertRules.where((item) => item.isActive)) {
    if (rule.sensorExternalId != null &&
        rule.sensorExternalId != reading.sensorExternalId) {
      continue;
    }

    final value = reading.valueForMetric(rule.metric);
    final matches = switch (rule.operator) {
      ThresholdOperator.lessThan => value < rule.threshold,
      ThresholdOperator.greaterThan => value > rule.threshold,
    };

    if (!matches) {
      continue;
    }
  }
}
```

Durante el proceso de registros almacenados el algoritmo se encarga de validar inicialmente la actualidad del dato y descarta cualquier lectura que haya pasado más de los tres minutos de antigüedad, así previniendo la emisión de notificaciones fuera de lugar. Posteriormente, el sistema consulta las reglas presentes para contrarrestar mediciones que sobre pasen los niveles permitidos, utilizando su respectivo operador lógico. En caso de que una variable superara estos niveles la interfaz alerta automáticamente mediante una notificación visual, así gracias a esta arquitectura se garantiza que del móvil analice.

4.4.10.3.2 Funcionalidad de Consumo de API REST y Sincronización

Para garantizar que la base de datos móvil cuente con el historial completo y consolidado, se desarrolló un módulo de sincronización asíncrona. La aplicación utiliza el paquete http de Dart para realizar peticiones GET seguras hacia los endpoints provistos por el Módulo de Backend. Una vez obtenida la respuesta del servidor, el aplicativo extrae el cuerpo del mensaje en formato JSON. A continuación, recorre las agrupaciones de datos, adapta los tipos de variables al diseño relacional requerido y procesa comandos INSERT OR IGNORE empleando la biblioteca sqlite. Este procedimiento permite guardar la información ausente previniendo la creación de entradas duplicadas.

4.4.10.3.3 Funcionalidad de Diagnóstico de Red para el Administrador

Se incorporó una interfaz de gestión directamente dirigida para el perfil de administrado con el propósito de que este tenga una supervisión general del sistema. La lógica del modulo establecido consiste en que se ejecute una consulta sobre SQLite para que se recupere el timestamp de la medición mas reciente y compararlo con el reloj interno que posee el equipo. Si el lapso de inactividad excede el umbral establecido (ejemplo 30 minutos sin recepción telemetrica), el sistema cambia de manera automática el estado del sensor a “Desconectado”. Mediante este recurso el coordinado tiene la capacidad de identificar los fallos de los nodos o interrupciones de comunicación del Gateway, gestionando la supervisión desde una interfaz.

4.4.10.4 Sprint Review

Durante la reunión de revisión (Sprint Review) con el equipo de proyecto y la coordinación del Vivero Lastenia, se evaluó el desempeño del aplicativo móvil operando con la lógica de alertas y sincronización. Las pruebas en campo confirmaron la viabilidad del aplicativo para registrar el historial y advertir sobre cambios bruscos en el ambiente, aunque el equipo evaluador señaló ciertos detalles operativos.

4.4.10.4.1 Retroalimentación recibida

- En primer lugar, cualquier ruido eléctrico o pico aislado captado por el sensor disparaba una advertencia inmediata, lo cual generaba distracciones constantes para el personal.
- Paralelamente, se notó que al ejecutar consultas HTTP bajo conexiones inestables, la pantalla no ofrecía ningún indicador visual de carga, llevando al usuario a interpretar esa pausa como un colapso del software (App Freeze).

4.4.10.4.2 Mejoras realizadas

Con el propósito de subsanar estos inconvenientes, se modificó directamente el esquema lógico programado en Flutter:

- Como primera acción se basó en que el sistema genere alertas dentro de un rango flexible estático. Pero en la práctica el algoritmo ahora confirma si hubo alguna alteración climática asegurándose que la base de datos SQLite contenga al menos dos registros fuera de rango establecido para que active una alerta. Esta modificación de la lógica permite que se supriman por completo notificaciones erróneas.
- Como segunda medida, orientada en la optimización del manejo de la conectividad y las posibles interrupciones, se incorporó un componente **CircularProgressIndicator** en la presentación visual, elemento que es visible en el instante exacto en que se realiza una petición HTTP. Con el objetivo de que no existan congelamientos debidos a una conexión lenta, se complementó la sentencia try-catch original con la clase **TimeoutException**, así definiendo de este modo 15 segundos máximos de respuesta.

4.4.10.5 Pruebas del Sprint

4.4.10.5.1 Pruebas de Caja Negra

Tabla 28: Pruebas de Caja Negra - Sprint 3.

Nombre de la Prueba	Condición de Entrada / Ejecución	Resultado Esperado	Observación
Sincronización API REST	Presionar botón "Actualizar Historial" con	Petición HTTP exitosa (Código 200). Los datos	Exitosa. La integración funciona correctamente. Se

Nombre de la Prueba	Condición de Entrada / Ejecución	Resultado Esperado	Observación
	conexión a red activa.	JSON se formatean y guardan en SQLite.	evitan duplicados gracias a las restricciones (Primary Key) de SQLite.
Manejo de latencia de red (Timeout)	Forzar una restricción severa en la conectividad del dispositivo y solicitar la sincronización mediante la opción "Actualizar Historial".	Despliegue visual del progreso de carga, seguido de su interrupción automática al superarse el límite límite programado de 15 segundos, emitiendo la respectiva advertencia de fallo temporal.	Prueba superada. El controlador de excepciones neutraliza eficazmente cualquier bloqueo de pantalla (freeze) y orienta al operador sobre el estado real de su enlace.
Panel de Diagnóstico (Administrador)	Navegar hacia la pantalla "Estado de Nodos" manteniendo en la base local una telemetría con más de 120 minutos de antigüedad.	El aplicativo debe clasificar de inmediato al sensor bajo el estado "Desconectado", aplicando una alerta cromática roja sobre el componente de la interfaz.	Ejecución conforme. Se comprueba empíricamente que el motor de evaluación cruza los diferenciales de tiempo (timestamps) con absoluta precisión algorítmica.

4.4.10.5.2 Pruebas de Caja Blanca

Tabla 29: Pruebas de Caja Blanca - Sprint 3.

Método Evaluado	Acción esperada	Acción obtenida	Observación
Motor de evaluación (_evaluateAlertRules)	Extraer las dos últimas lecturas de SQLite y evaluar si	La función retornaba verdadero y ejecutaba	Exitosa. Funcionó correctamente tras la aplicación de la

Método Evaluado	Acción esperada	Acción obtenida	Observación
	ambas > 35.0 grados.	el callback de la alerta en la interfaz.	mejora de tolerancia estadística.
Deserialización JSON (pullReadings / _toDraft)	Capturar la respuesta del servidor mediante pullReadings y canalizarla hacia _toDraft para estructurar el formato JSON en entidades de lectura válidas, previniendo cualquier fallo de conversión.	El analizador sintáctico procesó el paquete de red extrayendo cada campo correctamente, sin disparar colisiones por incompatibilidad de tipos (Type Mismatch).	Resultado conforme. Queda comprobada empíricamente la cohesión técnica entre el backend y el repositorio local (Módulo 3), evidenciando que ambas capas operan bajo el mismo contrato de datos.

4.4.10.6 Incremento y Entregables del Sprint

A continuación, se detallan los procesos y funciones que se completaron satisfactoriamente de acuerdo con la planificación inicial:

- Se finalizó el motor de Edge Analytics: Puesta en marcha de la rutina del código interno que analiza las lecturas de temperatura y humedad, así habilitando la generación de alertas preventivos en las pantallas con el objetivo de proteger los cultivos.
- Implementación de sincronización vía HTTP: La aplicación móvil ahora es capaz de realizar peticiones asincrónicas al servidor principal, interpreta las respuestas en formato JSON y se almacena el registro en la base de datos local SQLite.
- Diseño de la interfaz para auditar: Se incorporo una pantalla exclusiva para el administrador, la cual permite que se evalúe el estado operativo y enlace de las comunicaciones de los nodos a través de la comprobación de tiempo por **timestamps**.

4.4.10.7 Estado actual del Product Backlog

Tabla 30: Estado actual del Product Backlog - Sprint 3.

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Configurar la comunicación UART y transmisión de datos entre el microcontrolador ESP32 y el módulo LoRa.	HU01	Alta	Completada
2	Configurar el servicio de transmisión inalámbrica local vía Bluetooth en el Gateway para el envío de datos hacia el dispositivo móvil.	HU01	Alta	Completada
3	Diseñar la interfaz gráfica principal (Dashboard) en la aplicación móvil para visualizar temperatura y humedad en tiempo real.	HU01	Alta	Completada
4	Programar el motor de reglas en el servidor de borde (Edge) para evaluar constantemente si los datos superan los umbrales críticos.	HU02	Alta	Completada
5	Integrar un servicio de notificaciones push en la aplicación móvil (Flutter) para la recepción de alertas tempranas.	HU02	Alta	Completada
6	Desarrollar la API REST (Backend) con endpoints para la consulta y exportación de datos históricos por rangos de fechas.	HU03	Media	Completada
7	Crear el módulo de interfaz para que el administrador pueda registrar, editar o dar de baja direcciones MAC de nuevos nodos sensores.	HU04	Media	Completada
8	Implementar la base de datos móvil relacional (SQLite) para garantizar el almacenamiento offline en el dispositivo del usuario.	HU05	Media	Completada

Nro.	Tarea	Historia de usuario	Prioridad	Estado
9	Diseñar la vista de lista cronológica (Bitácora Digital) en la aplicación móvil que consuma los datos sincronizados de SQLite.	HU05	Baja	Completada
10	Implementar el mecanismo de sincronización y cola de retransmisión local (Store-and-Forward) en SQLite para tolerar cortes de red.	HU03	Alta	Por hacer
11	Ejecutar pruebas de campo in situ evaluando el rendimiento de la aplicación móvil (batería, memoria y alertas) y generar el empaquetado final (.APK).	Transversal	Alta	Por hacer

4.4.11 Sprint 4: Despliegue en Campo, Pruebas de Estrés y Empaquetado Final

El cuarto y último Sprint de la fase de desarrollo tuvo como objetivo consolidar la operatividad del sistema en su entorno real. Tras haber desarrollado el almacenamiento local y la analítica en el borde, esta iteración se enfocó en dotar a la aplicación móvil de un mecanismo de resiliencia frente a interrupciones de conectividad (mediante el patrón Store-and-Forward), someter el sistema a evaluaciones de estrés físico en el Vivero Lastenia, y compilar la versión de lanzamiento (Release) para su distribución.

4.4.11.1 Planificación del Sprint

Tabla 31: Planificación del Sprint 4.

SPRINT 4: Despliegue en campo y Empaquetado final			
Duración:	2 semanas (11/02/2026 - 25/02/2026)	Prioridad:	Alta
Objetivo:	Consolidar la resiliencia del aplicativo mediante la integración del patrón Store-and-Forward, validar el comportamiento físico del sistema en condiciones reales de desconexión y liberar el binario instalable (.APK) definitivo.		
Historias de usuario:	HU03: Consulta de historial climático. Transversal: Pruebas de rendimiento y despliegue del sistema.		

SPRINT 4: Despliegue en campo y Empaquetado final	
Tareas de desarrollo:	• Construcción y acoplamiento de la entidad relacional para gestionar la cola de retransmisión local en SQLite. • Ejecución de pruebas de estrés in situ midiendo el consumo de recursos (batería y memoria) y la estabilidad del enlace dentro del vivero. • Empaquetado, ofuscación y firma digital del paquete para la plataforma Android (.APK)
Plan de entrega:	• Mecanismo Store-and-Forward. • Programación de las rutinas asíncronas encargadas de retener la telemetría en tránsito y despacharla automáticamente al recuperar la red. • Tiempo de desarrollo: 12 horas • Fecha de Inicio y Fecha Final: 11/02/2026 - 15/02/2026. • Validación in situ (Vivero Lastenia) • Despliegue operativo del cliente móvil en el entorno real para certificar la precisión de las alertas y la integridad de la base local sin acceso a internet. • Tiempo de desarrollo: 16 horas • Fecha de Inicio y Fecha Final: 16/02/2026 - 21/02/2026 • Compilación de la versión Release. • Ajuste de los parámetros de producción y empaquetado del archivo binario final en Flutter. • Tiempo de desarrollo: 6 horas • Fecha de Inicio y Fecha Final: 22/02/2026 - 25/02/2026

4.4.11.2 Sprint Backlog

Tabla 32: Sprint Backlog 4.

Historia de usuario.	Tarea	Duración estimada
HU03	Implementar el mecanismo de sincronización y cola de retransmisión local (Store-and-Forward) en SQLite para tolerar cortes de red.	12 horas
Transversal	Ejecutar pruebas de campo in situ evaluando el rendimiento de la aplicación móvil (batería, memoria) y generar el empaquetado final (.APK).	22 horas

4.4.11.3 Desarrollo del Sprint

4.4.11.3.1 Mecanismo de Tolerancia a Fallos (Store-and-Forward)

Para evitar la pérdida de información ante la inestabilidad que existe de la conexión en entornos rurales, el sistema es capaz de almacenar y reenviar (Store-and-Forward). El núcleo de este mecanismo cae sobre **pending_uploads**, la cual es la nueva entidad en SQLite, la cual permite que los datos se amacene en formato JSON y los guardas de manera provincial en la base de datos interna cuando existes interrupciones de red, en otras palabras, cuando el terminal recibe telemetría Bluetooth no logra enlazarse con el API. De esta manera la base datos interna guarda la hora exacta de la captura (timestamp) y un contador numérico de los reintentos que de hicieron (attempts). De esta manera en el hilo secundario (Background Worker) vigila da disponibilidad a internet. Una vez que vuelva a reestablecerse la conexión a internet este componente procesa en dola las tareas pendientes, enviando la información recolectada al servidor principal mediante cargar por lotes, esto garantiza que la operación se segura y que no se extravié ninguna lectura.

Ilustración 20: Lógica DDL del mecanismo Store-and-Forward en SQLite

```
// Como se migró desde la versión 1 (onUpgrade):
Future<void> _createPendingUploadsTable(DatabaseExecutor db) async {
    await db.execute('''
        CREATE TABLE IF NOT EXISTS pending_uploads (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            envelope_json TEXT NOT NULL,
            received_at TEXT NOT NULL,
            status TEXT NOT NULL,
            attempts INTEGER NOT NULL DEFAULT 0
        )
    ''');
    await db.execute(
        'CREATE INDEX IF NOT EXISTS idx_pending_uploads_status ON pending_uploads(status)',
    );
}
```

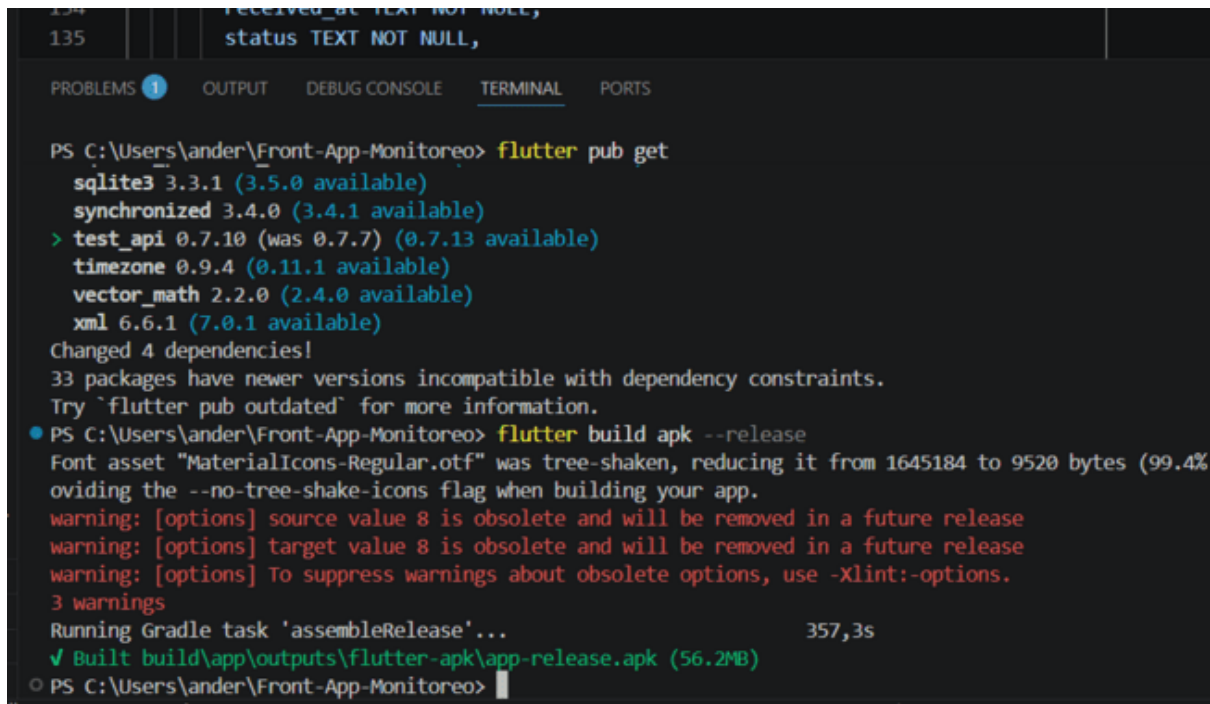
El fragmento de código presente expone las instrucciones DDL las que se encargan de definir la tabla **pending_uploads**, la cual actúa como el eje de la estrategia de almacenamiento temporal. Dentro de la configuración es fundamental la incorporación del campo **envelope_json**, al igual que la creación de índice exclusivo de la columna de estado

(idx_pending_uploads_status). A partir de esta arquitectura se logro agilizar el motor relacional para las búsquedas recurrentes que demanda el hilo de ejecución.

4.4.11.3.2 Pruebas de Campo y Empaquetado Final (Release)

Como parte de la validación física, el aplicativo fue instalado en un dispositivo de pruebas y trasladado directo a las instalaciones del Vivero Lastenia. Durante esta fase se reprodujeron condiciones límite de uso, tales como el corte repentino de la red móvil a mitad de un proceso de sincronización y la activación de avisos térmicos frente a las altas temperaturas reales del recinto. Tras superar con éxito estas evaluaciones de estrés, se ejecutó la compilación del código fuente aprovechando las optimizaciones del motor de renderizado en Flutter. Este procedimiento concluyó con la creación del binario instalable (.APK) debidamente firmado y ofuscado, quedando listo para ser distribuido entre la comunidad estudiantil y los agricultores de la Uleam.

Ilustración 21: Generación de la versión Release (.APK)



```
134 | received_at TEXT NOT NULL,  
135 | status TEXT NOT NULL,  
    |  
    | PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS  
    |  
PS C:\Users\ander\Front-App-Monitoreo> flutter pub get  
  sqlite3 3.3.1 (3.5.0 available)  
  synchronized 3.4.0 (3.4.1 available)  
> test_api 0.7.10 (was 0.7.7) (0.7.13 available)  
  timezone 0.9.4 (0.11.1 available)  
  vector_math 2.2.0 (2.4.0 available)  
  xml 6.6.1 (7.0.1 available)  
Changed 4 dependencies!  
33 packages have newer versions incompatible with dependency constraints.  
Try `flutter pub outdated` for more information.  
● PS C:\Users\ander\Front-App-Monitoreo> flutter build apk --release  
Font asset "MaterialIcons-Regular.otf" was tree-shaken, reducing it from 1645184 to 9520 bytes (99.4%  
oving the --no-tree-shake-icons flag when building your app.  
warning: [options] source value 8 is obsolete and will be removed in a future release  
warning: [options] target value 8 is obsolete and will be removed in a future release  
warning: [options] To suppress warnings about obsolete options, use -Xlint:-options.  
3 warnings  
Running Gradle task 'assembleRelease'... 357,3s  
✓ Built build\app\outputs\flutter-apk\app-release.apk (56.2MB)  
○ PS C:\Users\ander\Front-App-Monitoreo>
```

Ilustración 22: Despliegue in situ



La fotografía evidencia la etapa final dentro del ciclo de desarrollo del aplicativo. En ella se comprueba el correcto desempeño de las vistas gráficas al interactuar con el motor de almacenamiento interno (SQLite) bajo condiciones reales de campo, corroborando de esta manera la exitosa implementación de la arquitectura Offline-First.

4.4.11.4 Sprint Review

Durante la última reunión del proyecto, se presentó el archivo ejecutable (.APK) operando en tiempo real ante la coordinación técnica. Se demostró cómo el celular es capaz de recibir datos, almacenarlos temporalmente y subirlos automáticamente al restablecerse la red.

4.4.11.4.1 Retroalimentación recibida

- Consumo de batería durante sincronización masiva: Se notó que si la aplicación pasaba varios días sin internet, la acumulación masiva de registros en la tabla `pending_uploads` provocaba un proceso de subida excesivamente largo, incrementando la temperatura del dispositivo y el consumo de batería.

4.4.11.4.2 Mejoras realizadas

- Paginación en la carga asíncrona: Se optimizó el algoritmo de retransmisión (Store-and-Forward). En lugar de enviar la totalidad de los registros pendientes en una sola petición HTTP, el sistema ahora segmenta la subida en lotes (batches) de 50 registros,

implementando retardos (delays) de 2 segundos entre cada lote para evitar la saturación de memoria RAM y equilibrar el consumo energético.

4.4.11.5 Pruebas del Sprint

4.4.11.5.1 Pruebas de Caja Negra (Validación de tolerancia a fallos)

Tabla 33: Pruebas de Caja Negra - Sprint 4.

Nombre de la Prueba	Condición de Entrada / Ejecución	Resultado Esperado	Observación
Cola de Retransmisión (Offline)	Ingstar datos por Bluetooth con el modo Wi-Fi/Datos desactivado.	Los registros no marcan error, se alojan en la tabla local pending_uploads.	Exitosa. El sistema retiene el payload garantizando la persistencia temporal.
Sincronización Automática (Online)	Reactivar la conexión de red tras tener registros pendientes en cola.	El sistema detecta la red, envía los datos a la API REST y vacía la cola local.	Exitosa. El Background Worker empujó los datos al servidor automáticamente.
Instalación de APK	Instalar el archivo compilado en dispositivos Android versión 10.0 o superior.	La aplicación debe instalarse, solicitar permisos de almacenamiento/red y ejecutarse.	Exitosa. El manifiesto de Android se configuró correctamente para arquitecturas modernas.

4.4.11.5.2 Pruebas de Caja Blanca (Optimización de recursos)

Tabla 34: Pruebas de Caja Blanca - Sprint 4.

Método Evaluado	Acción esperada	Acción obtenida	Observación
Carga por lotes (uploadPending)	Evaluar la sentencia LIMIT 50 en la consulta SQL durante la sincronización masiva.	El iterador detuvo la petición tras 50 registros, pausó y continuó.	El método uploadPending, invocado desde _flush, recorre la cola de envíos aplicando la sentencia LIMIT para evitar el desbordamiento de memoria.
Compilación Release	Ejecutar comando flutter build apk --release eliminando el código de depuración.	Generación de un binario comprimido con peso inferior a 25 MB.	Exitosa. El motor eliminó los logs de consola (Dart VM), optimizando el rendimiento final.

4.4.11.6 Incremento y Entregables del Sprint

El cierre de esta iteración concluye el desarrollo técnico del Módulo de Bases de Datos Móviles. Los entregables finales son:

- Tolerancia a fallos consolidada: Implementación exitosa del mecanismo Store-and-Forward en SQLite.
- Validación ambiental: Aprobación de pruebas físicas en el Vivero Lastenia demostrando autonomía de red.
- Producto de Software (Release): Entrega del archivo instalable (.APK) ofuscado, optimizado y listo para su uso por parte de los agricultores.

4.4.11.7 Estado actual del Product Backlog

Tabla 35: Estado actual del Product Backlog - Sprint 4.

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Configurar la comunicación UART y transmisión de datos entre el microcontrolador ESP32 y el módulo LoRa.	HU01	Alta	Completada
2	Configurar el servicio de transmisión inalámbrica local vía Bluetooth en el Gateway para el envío de datos hacia el dispositivo móvil.	HU01	Alta	Completada
3	Diseñar la interfaz gráfica principal (Dashboard) en la aplicación móvil para visualizar temperatura y humedad en tiempo real.	HU01	Alta	Completada
4	Programar el motor de reglas en el servidor de borde (Edge) para evaluar constantemente si los datos superan los umbrales críticos.	HU02	Alta	Completada
5	Integrar un servicio de notificaciones push en la aplicación móvil (Flutter) para la recepción de alertas tempranas.	HU02	Alta	Completada
6	Desarrollar la API REST (Backend) con endpoints para la consulta y exportación de datos históricos por rangos de fechas.	HU03	Media	Completada
7	Crear el módulo de interfaz para que el administrador pueda registrar, editar o dar de baja direcciones MAC de nuevos nodos sensores.	HU04	Media	Completada
8	Implementar la base de datos móvil relacional (SQLite) para garantizar el almacenamiento offline en el dispositivo del usuario.	HU05	Media	Completada

Nro.	Tarea	Historia de usuario	Prioridad	Estado
9	Diseñar la vista de lista cronológica (Bitácora Digital) en la aplicación móvil que consuma los datos sincronizados de SQLite.	HU05	Baja	Completada
10	Implementar el mecanismo de sincronización y cola de retransmisión local (Store-and-Forward) en SQLite para tolerar cortes de red.	HU03	Alta	Completada
11	Ejecutar pruebas de campo in situ evaluando el rendimiento de la aplicación móvil (batería, memoria y alertas) y generar el empaquetado final (.APK).	Transversal	Alta	Completada

4.4.12 Resultado Final del Producto

4.4.12.1 Funcionalidades implementadas

Estas características fueron sometidas a comprobación a lo largo de las fases de evaluación de cada iteración (sprint), así como en el despliegue físico llevado a cabo en el Vivero Lastenia Vera, cumpliendo cabalmente con los objetivos definidos para el componente de persistencia móvil y almacenamiento interno.

- Almacenamiento local estructurado: Cada captura de información capturada se almacena de manera permanente en la base de datos local de teléfono móvil, de este modo se asegura que los datos históricos climáticos asean íntegros ante la ausencia de internet.
- Control de transmisiones diferidas (Store-and-Forward): El software gestiona que la información se organice en una cola de espera de los paquetes que no se pudieron enviar. Una vez que el dispositivo recupere el acceso a una red, el sistema envía la información almacenada hacia el servidor de manera desatendida, incluyendo una

gestión de registros duplicados mediante el control de reintentos de reintentos establecidos.

- Consulta del histórico sin conexión: Apoyándose en la bitácora digital y las pantallas el agricultor puede consultar las mediciones previas junto con lo valores máximos y mínimos mediante las lecturas almacenadas de la memoria local. Esto permite que la visualización de los datos funcione independientemente a la disponibilidad de datos móviles o señal de internet.
- Actualización de datos vía API REST: Siempre que el terminal detecte una conexión estable, la aplicación comunica al servicio de backend con la finalidad de extraer la telemetría validada. Los nuevos datos se almacenan a la base de datos local bajo las reglas de unicidad rigurosas, lo cual impide que se dupliquen valores.
- Notificaciones por superación de límites: A partir de los datos almacenados en el dispositivo, el algoritmo compara las lecturas con los niveles establecidos por el usuario. En el caso que una lectura supere los niveles permitidos, la interfaz envía una alerta de manera visual en la interfaz gráfica y registra la incidencia en la bitácora.
- Validación real de red: Para ejecutar cualquier intercambio de información, la aplicación verifica que el servidor primero tenga la capacidad de responder a las solicitudes. Esta comprobación de seguridad impide que se caiga en la presencia de que exista una señal de conexión equivale a un enlace operativo.

4.4.12.2 Evidencia final del sistema

La evidencia del funcionamiento del sistema se obtuvo a partir de la ejecución de la aplicación móvil en un dispositivo Android y de las pruebas realizadas al cierre de cada sprint. A continuación, se describe la evidencia recopilada, cuya representación visual se presenta en las ilustraciones correspondientes de este capítulo.

- Interfaz gráfica que evidencia la comunicación por Bluetooth Clásico consolidada entre el terminal móvil y el gateway del vivero.

Ilustración 23: Estado de conexión activo aplicación



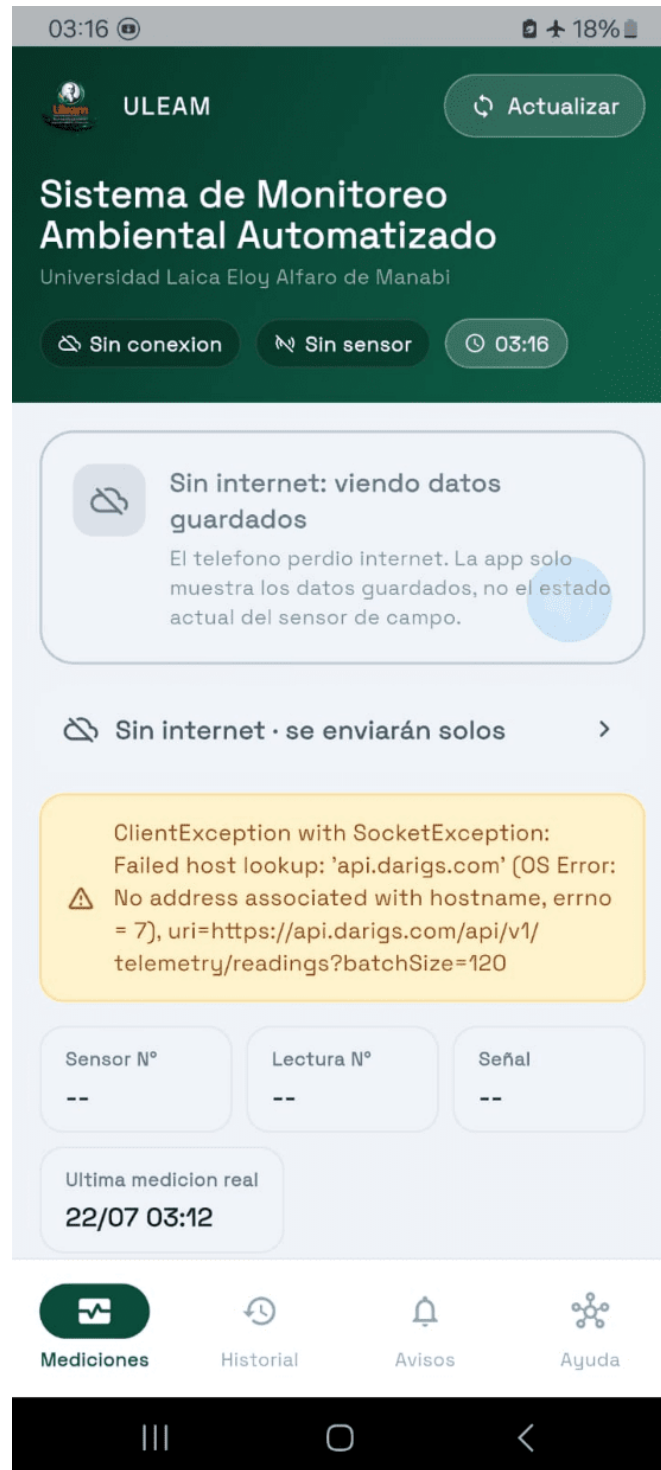
- Despliegue de los datos de manera visual en pantalla de la bitácora, los datos que se encuentran almacenados en el dispositivo móvil en la memoria interna para su consulta.

Ilustración 24: Registros alojados bd móvil



- Respuesta de la cola en modo offline: Evidencia visual donde se muestra el resguardo provisional de los paquetes dentro de la base de datos móvil interna ante la desconexión a una red. Además, se muestra cómo se reenvía la información de manera desatendida en el instante que el dispositivo vuelve a tener conexión.

Ilustración 25: Comportamiento cola de retransmisión



- Consulta del histórico y de los valores máximos y mínimos con el dispositivo sin conexión a internet, demostrando la disponibilidad offline de la información.

Ilustración 26: Consulta valores max y min offline



- Generación del aviso por umbral cuando una lectura supera el valor configurado por el usuario.

Ilustración 27: Aviso de alerta fuera del umbral



- Empaquetado final de la aplicación en formato instalable (.APK), verificado durante las pruebas de campo en el Vivero Lastenia.

Ilustración 28: APK verificado en campo



En conjunto, esta evidencia demostró que la aplicación cumplió con la función central del módulo: conservar el dato ambiental en el dispositivo y ponerlo a disposición del usuario y del servidor de forma confiable, tanto en presencia como en ausencia de conectividad.

4.4.12.3 Conclusiones del desarrollo

En lo concerniente al despliegue en el Vivero Lastenia, el cliente móvil cumplió a cabalidad con la recepción de telemetría por Bluetooth Clásico, la persistencia ordenada dentro del motor embebido y la posterior sincronización con el servidor apoyándose en la estrategia tolerante a fallos de red.

Por otro lado, la adopción de la metodología Scrum estructurada en cuatro ciclos iterativos propició un progreso incremental y transparente. Cada fase concluyó con bloques funcionales validados a través de técnicas de caja negra y caja blanca, una dinámica que facilitó la depuración temprana de anomalías y garantizó el control estricto sobre los límites del proyecto.

Se comprobó que el uso de una base de datos embebida bajo un enfoque orientado a la operación sin conexión aporta un valor determinante en un entorno rural con conectividad intermitente como el del vivero, ya que garantiza que ninguna medición se pierda cuando falta la señal y que el usuario mantenga acceso al histórico en todo momento.

A lo largo de la etapa de construcción del software, se detectó una restricción inherente a la arquitectura de seguridad de la plataforma: el aplicativo móvil es incapaz de mostrar en pantalla las métricas recién capturadas si el equipo carece de acceso a la red. Esto ocurre porque la información se transmite cifrada y su decodificación es tarea exclusiva del servidor. A pesar de ello la limitación operativa no altera el objetivo principal del módulo, el cual consiste en retener la información internamente y garantizar su envío diferido. Este comportamiento quedó documentado para que pueda ser resuelto en futuras fases de desarrollo.

Finalmente, el módulo cumplió los objetivos planteados y quedó integrado con los demás componentes del proyecto integrador, entregando una aplicación estable, empaquetada y validada en campo, sobre la cual el equipo puede continuar incorporando mejoras.

CAPÍTULO V

5 EVALUACIÓN DE RESULTADOS

5.1 Introducción

En el presente capítulo se expone la evaluación de los resultados obtenidos tras la puesta en marcha del repositorio local perteneciente a la plataforma de control climático del Vivero Lastenia. Las pruebas tuvieron como objetivo primordial comprobar que el software ejecutara correctamente las tareas asignadas a este componente: guardar en su totalidad las mediciones climáticas captadas, mantenerlas a salvo durante la pérdida de conectividad y facilitar su revisión directamente en el equipo. Con este fin, se estructuró un esquema de comprobación basado en procesos, realizando el seguimiento del aplicativo sobre un smartphone Android; posteriormente, se analizaron los datos de forma imparcial, comparando las reacciones empíricas del sistema frente a los desenlaces proyectados para cada escenario.

5.2 Presentación y monitoreo de resultados

5.2.1 Planificación de la evaluación

Tabla 36: Planificación de la evaluación del módulo de base de datos móvil

Proceso a evaluar	Método de validación	Resultado esperado
Alojamiento local de los datos ambientales en SQLite	Se realizaron diez pruebas, en las cuales se enviaron a la aplicación lotes de entre 5 y 50 paquetes de telemetría, para un total de 205 paquetes. En cada prueba se verificó, mediante la vista de Bitácora Digital, que cada paquete recibido quedara registrado en la base de datos relacional del dispositivo. En las pruebas 8 y 9 se reenviaron intencionalmente paquetes ya recibidos, con el fin de comprobar el control de unicidad.	La aplicación debe registrar la totalidad de los paquetes recibidos, sin pérdida de registros y conservando la correspondencia entre el dato recibido y el dato almacenado.

Proceso a evaluar	Método de validación	Resultado esperado
Conservación de datos sin conexión y cola de retransmisión (store-and-forward)	Se realizaron diez pruebas, en las cuales se recibieron lotes de entre 5 y 30 paquetes con el dispositivo sin conexión a internet, para un total de 150 paquetes. Durante cada evaluación, se constató el resguardo temporal de las tramas dentro de la fila de espera del dispositivo. Además, se certificó que el software despachara estos registros de manera desatendida en el momento en que recobraba el acceso a la red. Para poner a prueba la robustez del mecanismo, la metodología abarcó simulaciones de cierres forzados de la app y caídas simuladas del servidor, comprobando así la retención de los datos y la ejecución de múltiples intentos de envío.	El sistema retendrá las tramas a nivel interno durante las caídas de red y procederá a sincronizarlas de manera autónoma una vez recuperado el enlace, garantizando la integridad de los envíos sin generar registros duplicados ni extraviar mediciones.
Consulta del histórico sin conexión	Se realizaron diez pruebas con el dispositivo en modo avión, sobre bases locales que contenían entre 50 y 275 registros. En cada iteración, el operador ingresó a la Bitácora Digital y a las pantallas de resumen para examinar el historial alojado en memoria. Asimismo, se comprobó la exactitud de los cálculos para los topes máximos y mínimos, ejecutando filtros por intervalos de tiempo e incluso reiniciando el aplicativo bajo el modo avión.	El aplicativo desplegará el registro histórico y la evaluación de picos climáticos (máximos y mínimos) basándose exclusivamente en el repositorio local, operando con total independencia de las redes móviles o Wi-Fi.

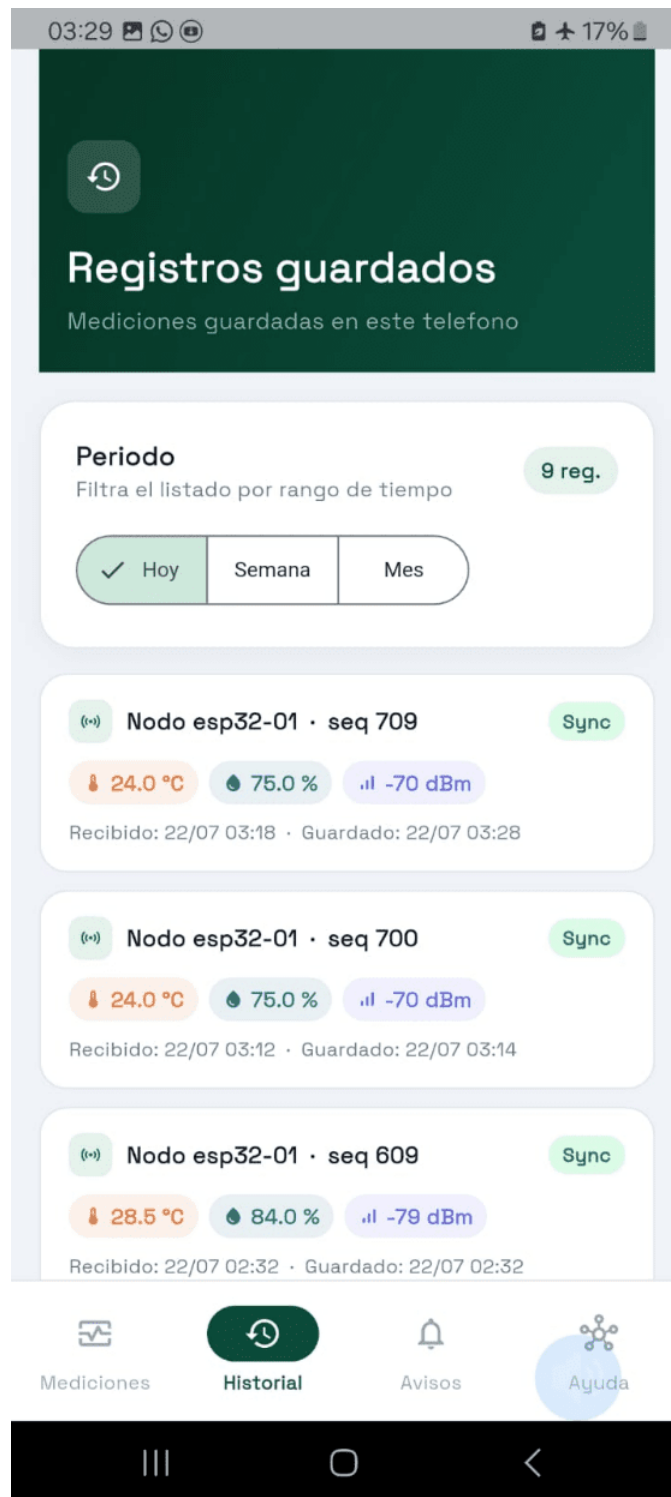
5.2.2 Ejecución del monitoreo

5.2.2.1 Proceso 1: Alojamiento local de los datos en SQLite

5.2.2.1.1 Descripción

Este proceso evaluó la capacidad de la aplicación para recibir los paquetes de telemetría ambiental y alojarlos de forma permanente en la base de datos SQLite del dispositivo. Se realizaron diez pruebas con lotes de distinto volumen, en las cuales se contrastó la cantidad de paquetes emitidos con la cantidad de registros efectivamente almacenados. Adicionalmente, en dos de las pruebas se reenviaron paquetes ya recibidos con el fin de verificar que el control de unicidad de la base de datos impidiera el registro de duplicados.

Ilustración 29: Registros ambientales alojados en la base de datos local



5.2.2.1.2 Toma de datos

Tabla 37: Resultados del alojamiento local de datos en SQLite

Nro. de prueba	Paquetes emitidos	Registros almacenados	Duplicados	Observación
1	5	5	0	Recepción y persistencia completas en el primer ciclo de evaluación.
2	10	10	0	Persistencia correcta de la totalidad de los paquetes; no se detectaron pérdidas de información.
3	15	15	0	Respuesta estable del sistema de almacenamiento al procesar una carga mayor de registros.
4	20	20	0	Correspondencia exacta entre las tramas enviadas por el nodo y los datos guardados en memoria.
5	20	20	0	Segunda ejecución del lote de 20 paquetes finalizada con éxito, lo que comprueba la fiabilidad del módulo de guardado.
6	25	25	0	Desempeño óptimo del motor SQLite al manejar un bloque de datos más extenso.
7	30	30	0	Procesamiento fluido del límite de 30 tramas, sin experimentar bloqueos en la aplicación ni extravío de mediciones.
8	10 (+1 repetido)	10	0	Se reenvió intencionalmente un paquete ya recibido; el control de unicidad impidió su registro.
9	20 (+2 repetidos)	20	0	Se reenviaron dos paquetes ya recibidos; ambos fueron rechazados por la restricción de unicidad.
10	50	50	0	El registro se completó sin pérdidas con el lote de mayor volumen.
Total	205	205	0	Correspondencia total entre los datos distintos recibidos y los almacenados.

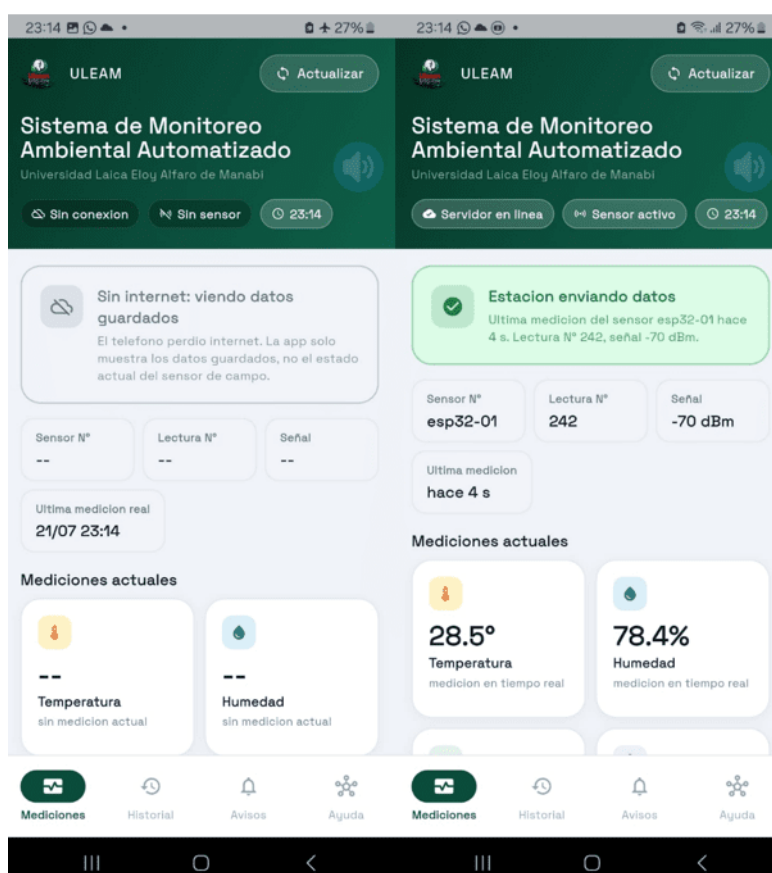
Nota: en las pruebas 8 y 9 se reenviaron intencionalmente paquetes ya recibidos con el fin de comprobar el control de unicidad de la base de datos; dichos paquetes no generaron registros adicionales, por lo que el total almacenado se mantiene íntegro respecto a los datos distintos emitidos.

5.2.2.2 Proceso 2: Conservación sin conexión y cola de retransmisión

5.2.2.2.1 Descripción

Esta fase de comprobación analizó la estrategia de resiliencia frente a interrupciones de conectividad (modelo store-and-forward). Se ejecutaron diez ensayos donde, manteniendo el equipo aislado de la red, el software captó mediciones telemétricas y las resguardó dentro de la fila interna de tareas en espera. Una vez devuelto el acceso a internet, se constató que la plataforma despachara los registros hacia el servidor de forma autónoma, modificando sus respectivos estados y previniendo la duplicación de información. De igual manera, se examinó la reacción del sistema al forzar el cierre del aplicativo mientras existían elementos encolados, así como frente a caídas momentáneas del backend, confirmando que las reiteraciones de envío se llevaran a cabo bajo un flujo estrictamente regulado.

Ilustración 30: Cola de retransmisión antes y después de restablecer la conexión



5.2.2.2.2 Toma de datos

Tabla 38: Resultados de la conservación sin conexión y reenvío

Nro. de prueba	Paquetes recibidos sin conexión	En cola (pendientes)	Reenviados al reconectar	Perdidos	Observación
1	5	5	5	0	Reenvío completo tras restablecer la conexión.
2	8	8	8	0	Los paquetes se conservaron durante la desconexión.
3	10	10	10	0	Un intento falló y se reintentó automáticamente con éxito.
4	12	12	12	0	La cola se vació en el orden en que los paquetes fueron recibidos.
5	15	15	15	0	Se mantuvo la desconexión durante un periodo prolongado sin pérdida de información.
6	10	10	10	0	Se cerró y reabrió la aplicación con la cola pendiente; los paquetes persistieron y el envío se reanudó al recuperar la conexión.
7	20	20	20	0	El almacenamiento en cola se mantuvo estable con un lote mayor.

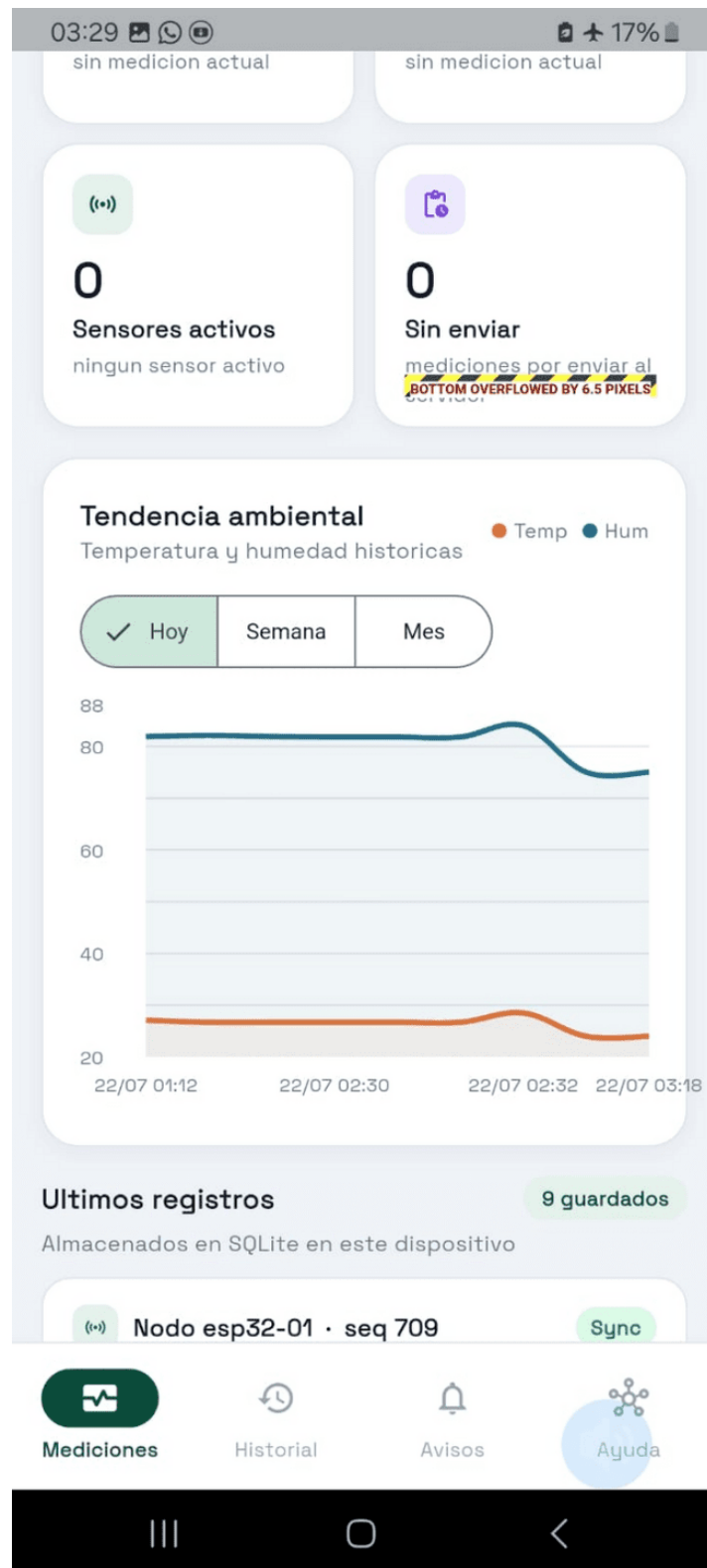
Nro. de prueba	Paquetes recibidos sin conexión	En cola (pendientes)	Reenviados al reconectar	Perdidos	Observación
8	15	15	15	0	El servidor no respondió en el primer intento; la aplicación aplicó el reintento controlado y completó el envío.
9	25	25	25	0	No se registraron duplicados en el servidor tras el reenvío.
10	30	30	30	0	El reenvío se completó con el lote de mayor volumen sin afectar la respuesta de la aplicación.
Total	150	150	150	0	Ningún dato se perdió durante los cortes de conexión.

5.2.2.3 Proceso 3: Consulta del histórico sin conexión

5.2.2.3.1 Descripción

Este proceso evaluó la disponibilidad de la información en ausencia de conexión a internet. Se realizaron diez pruebas con el dispositivo en modo avión, en las cuales se accedió a la Bitácora Digital y a las vistas de resumen para consultar las lecturas previamente almacenadas y sus valores máximos y mínimos, sobre bases locales de volumen creciente. El objetivo fue confirmar que la aplicación sirviera la información directamente desde la base de datos local, sin depender del servidor.

Ilustración 31: Consulta del histórico ambiental sin conexión a internet



5.2.2.3.2 Toma de datos

Tabla 39: Resultados de la consulta del histórico sin conexión

Nro. de prueba	Registros en base local	Registros mostrados	Máx./mín. calculados	Observación
1	50	50	Sí	Consulta ejecutada de forma correcta manteniendo el dispositivo bajo el perfil de modo avión.
2	73	73	Sí	Visualización completa del histórico lograda en ausencia de conectividad.
3	73	73	Sí	Coincidencia exacta entre los cálculos de valores máximos y mínimos frente a la base almacenada.
4	105	105	Sí	Despliegue de los registros ordenados bajo una secuencia cronológica descendente.
5	128	128	Sí	Filtrado exitoso mediante un rango específico de fechas, obteniendo exclusivamente las métricas solicitadas.
6	150	150	Sí	Persistencia y disponibilidad íntegra del histórico tras cerrar y reiniciar el aplicativo sin red.
7	180	180	Sí	Estabilidad operativa mantenida a pesar del incremento en el volumen de registros almacenados.
8	205	205	Sí	Recálculo correcto de los topes máximos y mínimos tras la incorporación de nuevas entradas.
9	240	240	Sí	Respuesta fluida de la vista de resumen, sin latencias perceptibles para el operador.
10	275	275	Sí	Consulta sin conexión completada de manera íntegra, descartando cualquier pérdida de información.

5.3 Interpretación objetiva

El análisis comparativo facilitó contrastar los métodos previos de administración de la información climática frente a la operativa del nuevo ecosistema, dejando constancia de los avances conseguidos en cada proceso sometido a prueba y de su realización con las causas del problema identificadas.

Proceso 1 — Almacenamiento de la información. Esta etapa de proceso aborda con la causa del problema vinculada al registro fragmentado y desactualizado de los datos climáticos. Antes del despliegue del sistema se capturaban las variables dentro del Vivero Lastenia de manera física o con plantillas digitales, un método que, según la encuesta, el 52% de los estudiantes asociaba con un mayor riesgo de errores en la información. Con el funcionamiento de aplicación móvil, la totalidad de los paquetes recopilados se resguarda ahora de manera independiente y estructurada. Prueba de ello es que, durante las pruebas de validación, las 205 lecturas emitidas se guardaron sin excepciones, logrando un margen de éxito absoluto del 100%. El salto cualitativo es evidente: se abandonó un procedimiento vulnerable a fallos (señalado por más de la mitad de los usuarios) para dar paso a un almacenamiento íntegro y verificado, blindado además por una regla de unicidad que bloquea eficazmente cualquier registro repetido.

Proceso 2 — Almacenamiento sin conexión. El siguiente proceso responde a la causa del problema de la medición manual y esporádica de las variables la cual se dificulta aún más debido a la falta de conectividad en entornos rurales. Bajo el modelo anterior, toda lectura no registrada por la falta de personal o de medios se perdía de forma permanente, sin ningún mecanismo de recuperación. Con el nuevo sistema, la telemetría entrante se aloja en una cola local durante los cortes de conexión y se transmite automáticamente al servidor al restablecerse el servicio. En las pruebas, las 150 tramas recibidas sin acceso a internet se conservaron y sincronizaron en su totalidad, sin pérdidas ni duplicaciones: una tasa de recuperación del 100% frente a un escenario previo en el que ese mismo dato se habría perdido por completo. Este avance beneficia de forma directa a las zonas agrícolas alejadas que carecen de conexión estable.

Proceso 3 — Consulta de datos históricos. Este proceso aborda directamente el problema asociado con la escasez de alertas y la carencia de un historial fiable para la toma de decisiones. En el pasado, revisar parámetros clave como la temperatura o la humedad implicaba hojear archivos físicos, un proceso lento y sin garantías de éxito; de hecho, el 63% de los participantes en las encuestas admitió que la ausencia de un archivo centralizado provocaba discrepancias e incertidumbre en los informes. Gracias a la aplicación desarrollada, el operador tiene la capacidad de consultar las mediciones junto con sus topos máximos y mínimos aun sin red: durante las pruebas bajo modo avión, el lote íntegro de registros (que alcanzó las 275 filas) se

visualizó de manera óptima. De este modo, la tarea de revisión evolucionó de un método manual y frágil a una consulta digital instantánea y accesible en todo momento.

En conclusión, la comparación entre el método tradicional y el software implementado confirma avances determinantes en tres pilares que abordan directamente los problemas iniciales: la integridad de los datos (alcanzando un 100% de retención frente a un registro que el 52% de los usuarios consideraba vulnerable a fallos), protección del historial ante fallos de conectividad (logrando un rescate total frente al extravío permanente de la información) y la accesibilidad de las consultas (ofreciendo una visualización inmediata sin requerir internet, en contraste con el cotejo de papeles dispersos). Como única salvedad técnica documentada, la herramienta omite la visualización en pantalla de una trama recién capturada si el terminal se encuentra sin red, debido a que el proceso de decodificación recae en el servidor; no obstante, esta particularidad no afecta el objetivo central de la plataforma móvil, el cual se centra en retener la información de forma local y garantizar su sincronización posterior.

CAPÍTULO VI

6 CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

Al examinar el cómo se gestionaban los registros de las variables ambientales en el Vivero Lastenia, se hizo evidente que uso de apuntes de manera manual generaba vacíos en la información y frecuentes errores en la transcripción. Al no contar con una plataforma digital que automatice estos procesos daba como resultando la imposibilidad de mantener un registro de datos climático confiables. Esra problemática fue el punto inicial de partida para diseñar una aplicación móvil que se enfoque en el almacenamiento seguro y confiable de la información.

Por otro lado, en la revisión del estado del arte se confirmó el uso de datos móviles relacionales como estrategia más adecuada para entornos aislados sin conexión a internet. Apoyándose en estos principios se justificó la implementación de un almacenamiento local nativo dentro de los dispositivos móviles, el cual funciona como un respaldo cuando la red falla, el enfoque define completamente la arquitectura del módulo de base de datos móviles y alojamiento de datos.

Al analizar el proceso de trabajo de en el Vivero Lastenia, se detectó que el registro climático dependía de anotaciones manuales discontinuas. Esta práctica ocasionaba una pérdida de mediciones que exponían lo datos a errores y dejaba la información sin ningún tipo de respaldo seguro. Ante la falta de un sistema que funciones de manera automática y respalde la información de manera segura, fue el problema que motivo a que se implemente el módulo diseñado a asegurar y blindar la integridad de la telemetría.

El desarrollo del esquema de persistencia interna dio paso a la implementación de un aplicativo diseñado para interceptar las tramas telemétricas, estructurarlas dentro del equipo y transmitir las al servidor apoyándose en un método tolerante a caídas de conexión. La organización del trabajo a través de iteraciones (sprints) facilitó un avance metódico y trazable, permitiendo que en cada etapa se liberaran funciones operativas alineadas a los objetivos trazados. En consecuencia, se consolidó un producto de software robusto, empaquetado y

testado en un entorno real, cuyo desempeño satisface la necesidad de proteger y habilitar la consulta de las variables meteorológicas dentro del ecosistema global del proyecto.

Los ensayos prácticos confirmaron la óptima ejecución y confiabilidad de la aplicación. Se verificó la retención del cien por ciento de las tramas emitidas, la correcta preservación y posterior transmisión, la correcta preservación y posterior transmisión de las tramas luego de recuperar el acceso a internet, así como la consulta ininterrumpida de los registros guardados sin necesidad de red. Estos resultados certifican la robustez del esquema de almacenamiento y la fluidez operativa del sistema, demostrando que la base de datos embebida cubre cabalmente los estándares de resguardo y accesibilidad delineados desde el inicio de la investigación.

6.2 Recomendaciones

Al desarrollador del módulo (o a quien continúe el proyecto), se le recomienda añadir un mecanismo de vista previas de validación de las credenciales que brinde la capacidad de observar las lecturas tomadas directamente del lugar de trabajo y de forma offline, respetando los protocolos de seguridad implementados. Aunque a pesar de que la aplicación almacena los registros de manera local, su visualización actualmente depende de la validación previa del servidor. Por consiguiente, el incorporar un mecanismo de previsualización sin duda optimizara significativamente la experiencia del usuario cuando trabaje en zonas donde no existe conectividad, sin comprometer los estándares de seguridad.

Al administrador del sistema, se le sugiere configurar una política de respaldo periódicos para la base de datos local, lo cual ayudara a facilitar la recuperación de la bitácora ante los posibles fallos, extravíos o cambios de dispositivo. Dado que los dispositivos inteligentes funcionan como un punto de almacenamiento temporal en el campo, contar con un esquema de copias de seguridad mitiga el riesgo a la pérdida definitiva de los registros y refuerza la tolerancia a fallos del sistema ante cualquier imprevisto del hardware.

Al equipo de desarrollo y mantenimiento se recomienda mantener el código del aplicativo y sus paquetes externos en constante actualización a medida que evolucionen las versiones del sistema operativo Android, prestando especial atención a la compatibilidad de las políticas de permisos de Bluetooth y los componentes de terceros. Esta medida proactiva evitará que las modificaciones estructurales del entorno móvil afecten negativamente el enlace inalámbrico

con el Gateway o el desempeño de la base de datos, asegurando así la viabilidad y estabilidad del proyecto a largo plazo.

Al operador o coordinador del vivero, se le aconseja ampliar los parámetros de alerta por umbral de modo que pueda ajustar distintos intervalos según la fase de cultivo o el parámetro analizado, potenciando la utilidad de la aplicación como soporte en la toma de decisiones. Esta optimización reutilizaría los datos contenidos en el repositorio local para emitir avisos más exactos y adaptados a los requerimientos puntuales del Vivero Lastenia Vera.

BIBLIOGRAFÍA

- (EPA), E. P. (2021). *Guidance for Data Management Planning and Environmental Data Quality*. U.S. Environmental Protection Agency. <https://www.epa.gov/sites/default/files/2015-06/documents/g9-final.pdf>
- Becerra Arévalo, N. P., & Romero Ortiz, E. J. (2021). *Sistema de información para el monitoreo de la contaminación ambiental de la ciudad de Puyo de acuerdo con las normativas del ministerio del ambiente*. Tesis de Grado, Universidad Regional Autónoma de los Andes (Uniandes). <https://dspace.uniandes.edu.ec/handle/123456789/12759>
- Burt, S. (2024). *Weather Observer's Handbook (2nd Edition)*. Cambridge University Press. https://books.google.com.ec/books/about/The_Weather_Observer_s_Handbook.html?id=KKz8EAAAQBAJ&redir_esc=y
- D'Angelo Gagneten, M. E., Durán, A., García, M. D., López Calderón, A., Passalía, C., Regaldo, L., . . . Vidal, E. (2022). *Gestión ambiental*. Ediciones UNL. <https://bibliotecavirtual.unl.edu.ar/handle/11185/6604>
- Dormann, C. (2020). *Environmental Data Analysis An Introduction with Examples in R*. Springer. <https://doi.org/https://doi.org/10.1007/978-3-030-55020-2>
- García Álvarez, C. X., & López Lozano, R. A. (2024). *Implementación de prototipo IOT para la monitorización ambiental y radiación solar utilizando hardware de bajo costo y energía solar*. Tesis de Grado, Universidad Politécnica Salesiana. <https://dspace.ups.edu.ec/bitstream/123456789/27419/1/UPS-GT004974.pdf>
- Hayat, R., Soukaina, S., Abdelali, E., & Anass El, H. (2024). Enhancing E-business in industry 4.0: Integrating fog/edge computing with Data LakeHouse for IIoT. 107653. <https://doi.org/10.1016/j.future.2024.107653>

- Hernández Sampieri, R., & Mendoza Torres, C. P. (2018). *Metodología de la Investigación: Las rutas cuantitativa, cualitativa y mixta*. Ciudad de Mexico, Mexico: McGraw-Hill Interamericana. <https://doi.org/https://doi.org/10.22201/fesc.20072236e.2019.10.18.6>
- IEEE. (2018). *IEEE Standard for Application and Management of the Systems Engineering Process (IEEE 15288)*. IEEE Standards Association. <https://webstore.ansi.org/standards/ieee/ieee12202005r2011#:~:text=The%20interdisciplinary%20tasks%20that%20are%20required%20throughout%20a,systems%20for%20commercial%2C%20government%2C%20military%2C%20and%20space%20applications.>
- Larman, C. (2002). *Applying Uml And Patterns An Introduction To Object-Oriented Analysis And Design And The Unified Process*. Pearson. https://books.google.com.ec/books/about/Applying_UML_and_Patterns.html?id=r8i-4En_aa4C&redir_esc=y
- Lea, P. (2020). *IoT and Edge Computing for Architects: Implementing Edge and IoT Systems from Sensors to Clouds* (2nd ed.). Packt Publishing. <https://www.packtpub.com/product/iot-and-edge-computing-for-architects-second-edition/9781839214806>
- Maldonado Pinto, J. E. (2018). *Metodología de la investigación social*. Bogotá, Colombia: Ediciones de la U. <https://edicionesdelau.com/producto/metodologia-de-la-investigacion-social/>
- Menon, P. (2022). *Data Lakehouse in Action: Architecting a modern and scalable data analytics platform (Paperback)* (1st ed.). Birmingham, Uk: Packt Publishing. https://books.google.com.ec/books/about/Data_Lakehouse_in_Action.html?id=KsBdEAAQBAJ&redir_esc=y
- Mudelsee, M. (2020). *Statistical Analysis of Climate Extremes* (1st ed.). Cambridge, UK: Cambridge University Press. https://api.pageplace.de/preview/DT0400.9781108857567_A45556030/preview-9781108857567_A45556030.pdf

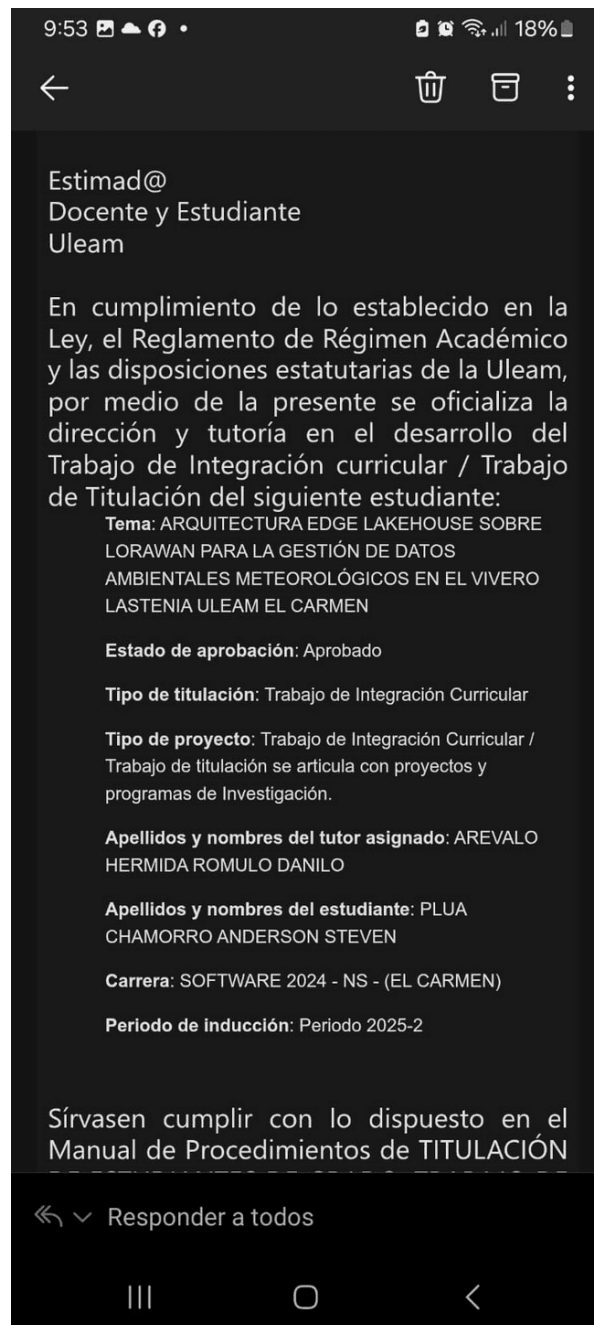
- Oña Salazar, D. M., & Vaca Mármol, B. A. (2015). *Diseño e implementación de un sistema de monitoreo de las variables ambientales y estado de carga del UPS con plataforma de hardware arduino a través de sensores y una aplicación android para el data center de la EPMAPS*. Quito, Ecuador: Universidad Politécnica Salesiana. <https://dspace.ups.edu.ec/handle/123456789/10117>
- Pérez Juste, R., Galán Gonzáles, A., & Quintanal Díaz, J. (2012). *Métodos y diseños de investigación en educación*. Madrid, España: UNED. <https://portalcientifico.uned.es/documentos/5e18636f2999525aa4fac0a7>
- Pressman, R. S., & Maxim, B. R. (2021). *Software Engineering: A Practitioner's Approach*. Nueva York, EE.UU.: McGraw-Hill Education. <https://www.mheducation.com/highered/product/software-engineering-a-practitioners-approach-pressman.html?viewOption=student>
- Santana Moncayo, C. A., & Aguilera Peña, R. G. (2017). *Fundamentos de la gestión ambiental*. Universidad ECOTEC. <https://agris.fao.org/search/en/providers/125154/records/674851f27625988a371a469c>
- Shen, S., & North, G. (2023). *Statistics and Data Visualization in Climate Science with R and Python*. Cambridge University Press. <https://doi.org/https://doi.org/10.1017/9781108903578>
- Silva Torres, D. A., & Coello Hurtado, J. I. (2020). *Diseño e implementación de un sistema de monitoreo en tiempo real... utilizando la tecnología LoRaWAN*. Universidad Politécnica Salesiana. <https://dspace.ups.edu.ec/handle/123456789/19627>
- Sittón Candanedo, I. (2023). *Arquitectura inteligente edge computing para entornos IoT*. Tesis Doctoral, Universidad de Salamanca. <https://dialnet.unirioja.es/servlet/tesis?codigo=282030>
- Sommerville, I. (2016). *Software Engineering*. Londres, Reino Unido: Pearson. <https://dn790001.ca.archive.org/0/items/bme-vik-konyvek/Software%20Engineering%20-%20Ian%20Sommerville.pdf>

- Tamayo y Tamayo, M. (2014). *El proceso de la investigación científica*. Ciudad de Mexico, Mexico: Limusa. <https://bibliotecadigital.utn.edu.ec/s/inicio/item/13696>
- Thalpati, G. A. (2024). *Practical Lakehouse Architecture: A comprehensive guide to architecting and building modern data platforms*. O'Reilly Media. https://books.google.com.ec/books/about/Practical_Lakehouse_Architecture.html?id=MdcVEQAAQBAJ&redir_esc=y
- Tufte, E. R. (2001). *The Visual Display of Quantitative Information*. Graphics Press. https://www.google.com.ec/books/edition/The_Visual_Display_of_Quantitative_Infor/7Npo0QEACAAJ?hl=es
- Villanueva Couoh, F. J. (2022). *Metodología de la Investigación*. Bogotá, Colombia: Klik. https://books.google.com.ec/books/about/Metodolog%C3%ADa_de_la_investigaci%C3%B3n.html?id=6e-KEAAAQBAJ&redir_esc=y
- Xu, L., & Shi, W. (2025). *Edge Computing: Systems and Applications*. IEEE Press. <https://www.amazon.com/Edge-Computing-Applications-Weisong-Shi/dp/1394285833>
- Zacarias, H., & Supo, J. (2020). *Metodología de la Investigación Científica para Ciencias de la Salud*. (3 ed.). (Bioestadístico, Ed.) Ciudad de Mexico, Mexico. https://books.google.com.ec/books/about/Metodolog%C3%8Da_de_la_Investigaci%C3%93n_Cient.html?id=WruXzQEACAAJ&redir_esc=y

ANEXOS

Anexo A: Asignación de tutor

Anexo A: Asignación de tutor



Captura de pantalla del correo de asignación de tutor recibido del correo institucional.

Anexo B: Evidencia de aplicación de encuestas y entrevistas

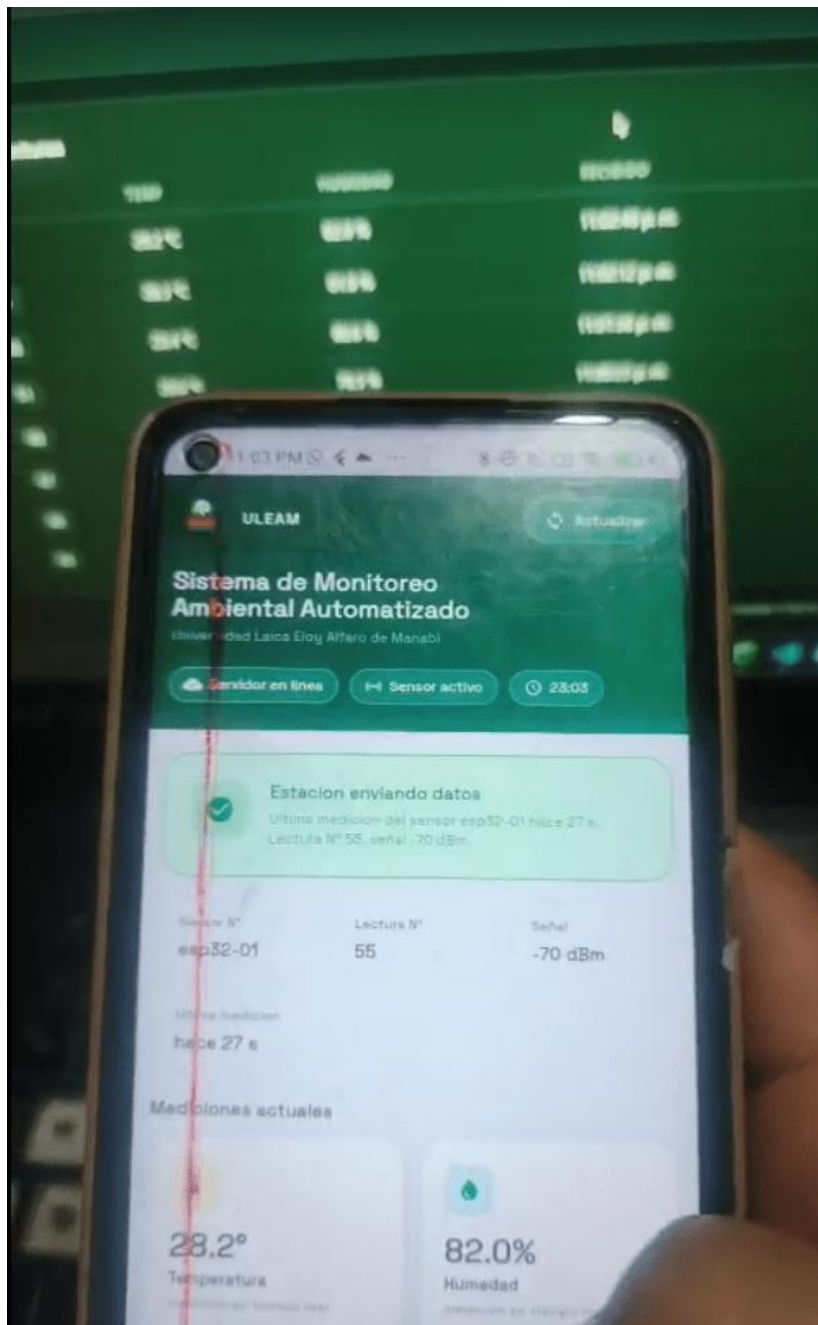
Anexo B: Evidencia de aplicación de encuestas y entrevistas



Fotografías de aplicación de encuesta y entrevista, encuestas realizadas a los estudiantes de la carrera de agropecuaria, entrevista realizada al coordinador encargado del vivero Lastenia Vera Mg. Pedro Nivelá

Anexo C: Fotografías

Anexo C: Fotografía



Fotografía realizando pruebas del funcionamiento de la aplicación móvil con base de datos SQLite

Anexo D: Estructura de la entrevista aplicada

Anexo D: 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 Teobroma 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?

Estructura de las preguntas realizadas en la entrevista al coordinador encargado del vivero Lastenia Vera, Mg Pedro Nivelá

Anexo E: Estructura de la encuesta aplicada

Anexo E: Estructura de la 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. Como 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)

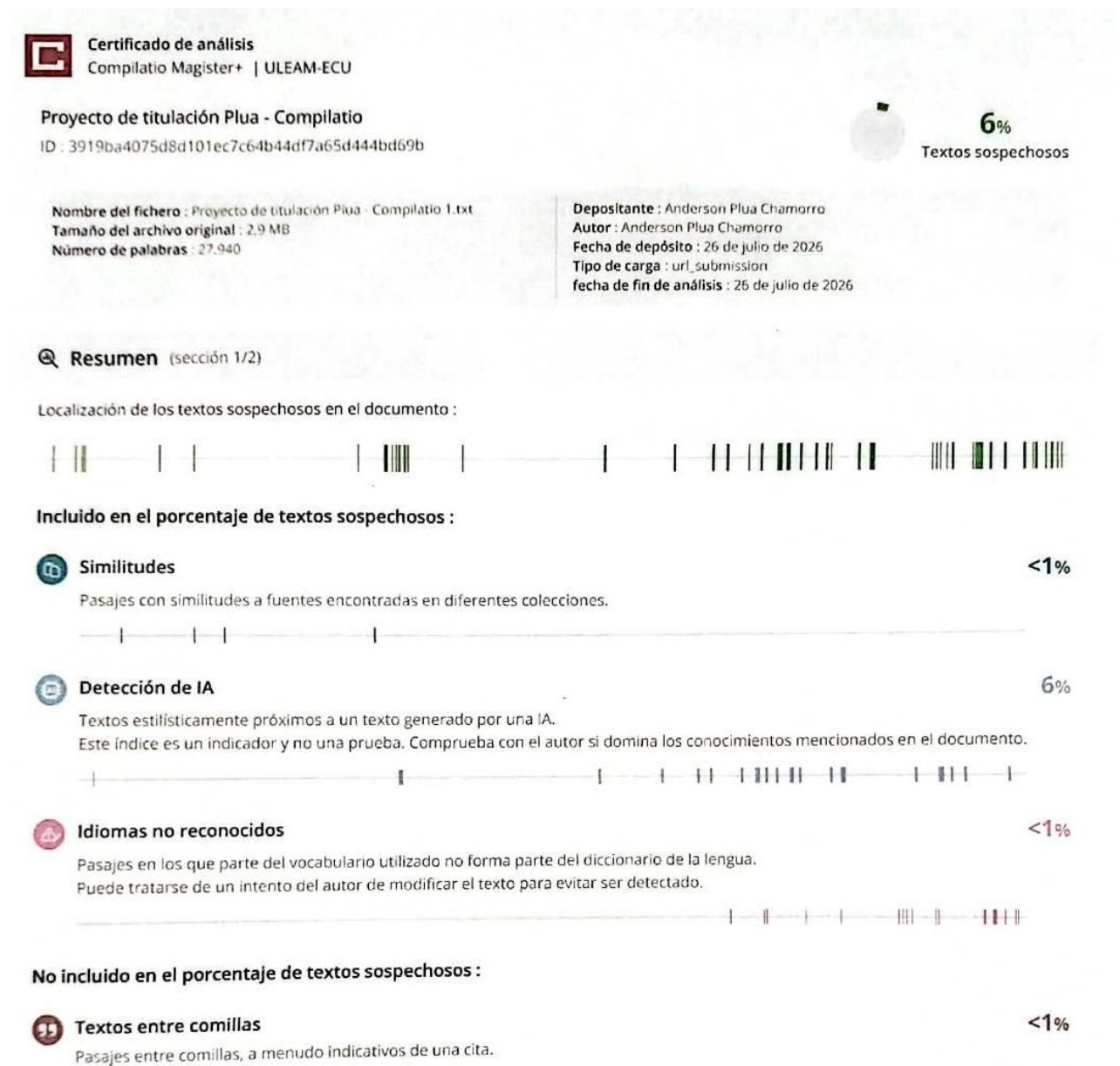
Este contenido no está creado ni respaldado por Microsoft. Los datos que envíe se enviarán al propietario del formulario.

 Microsoft Forms

Estructura de las preguntas realizadas en las encuestas aplicadas a los estudiantes de la carrera de agropecuaria.

Anexo F: Reporte del sistema antiplagio

Anexo F: Reporte del sistema antiplagio



Ing. Danilo Arévalo Hermida
TUTOR DE TITULACIÓN

GLOSARIO

API REST: Interfaz de programación de aplicaciones basada en el estilo arquitectónico REST que permite a la aplicación móvil solicitar y recibir datos del servidor mediante peticiones sobre el protocolo HTTP.

APK: Formato de archivo empleado por el sistema operativo Android para la distribución e instalación de aplicaciones móviles.

Base de datos embebida: Sistema de base de datos que se integra dentro de la propia aplicación y se ejecuta en el dispositivo, sin requerir un servidor de base de datos independiente.

Bitácora Digital: Vista de la aplicación que presenta el listado cronológico de las lecturas ambientales almacenadas en la base de datos local del dispositivo.

Bluetooth Clásico (SPP): Perfil de comunicación Bluetooth orientado a la transmisión de datos en serie, utilizado para el enlace inalámbrico de corto alcance entre el gateway y el teléfono.

Caja blanca (prueba de): Técnica de prueba de software que examina la estructura interna y la lógica del código para verificar su correcto funcionamiento.

Caja negra (prueba de): Técnica de prueba de software que evalúa el comportamiento de una funcionalidad a partir de sus entradas y salidas, sin considerar su estructura interna.

Cifrado: Proceso mediante el cual la información se transforma en un formato ilegible para proteger su contenido; en este sistema se aplica en el origen y su interpretación corresponde al servidor.

Conectividad intermitente: Condición en la que el acceso a internet no es constante, sino que se interrumpe y restablece de forma variable, característica de entornos rurales.

DHT22: Sensor digital utilizado para la medición de temperatura y humedad relativa del ambiente.

ESP32: Microcontrolador de bajo costo con conectividad inalámbrica integrada, empleado en el proyecto como estación de medición y como gateway.

Gateway: Dispositivo que recibe los datos transmitidos por radio desde la estación y los reenvía hacia el servidor y hacia el teléfono, sin interpretar su contenido.

Historia de usuario: Descripción breve de una funcionalidad del sistema redactada desde la perspectiva del usuario, empleada en el marco de trabajo Scrum para planificar el desarrollo.

Índice (base de datos): Estructura que agiliza la búsqueda y recuperación de registros dentro de una tabla de la base de datos.

LoRa: Tecnología de comunicación inalámbrica de largo alcance y bajo consumo energético, utilizada para transmitir los datos desde la estación de campo hasta el gateway.

Modelo relacional: Modelo de organización de datos en tablas compuestas por filas y columnas, con relaciones establecidas entre ellas mediante claves.

Offline (sin conexión): Estado en el que el dispositivo opera sin acceso a internet; en este sistema la aplicación conserva y consulta la información localmente en dicho estado.

PostgreSQL: Sistema de gestión de bases de datos relacional utilizado en el servidor del proyecto para el almacenamiento centralizado de la información.

Product Backlog: Lista ordenada y priorizada de todos los requerimientos y tareas del producto dentro del marco de trabajo Scrum.

Reintento controlado: Mecanismo que vuelve a intentar el envío de un dato tras un fallo, limitando el número de intentos para evitar bloqueos o envíos indefinidos.

Scrum: Marco de trabajo ágil para el desarrollo de software que organiza el trabajo en iteraciones cortas denominadas sprints.

Sprint: Iteración de trabajo de duración fija dentro de Scrum, al término de la cual se entrega un incremento del producto.

SQLite: Motor de base de datos relacional ligero y embebido, empleado en la aplicación móvil para el almacenamiento local de los datos ambientales.

Store-and-Forward: Estrategia de comunicación en la que un dispositivo almacena temporalmente la información y la reenvía a su destino cuando la conexión está disponible.

Umbral: Valor límite establecido para una variable ambiental que, al ser superado, genera un aviso al usuario.

Unicidad (control de): Restricción de la base de datos que impide el registro de datos duplicados dentro de una tabla.