



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

SISTEMA EMBEBIDO CON ARQUITECTURA IOT PARA EL
MONITOREO DE ESTACIÓN METEOROLÓGICA EN EL VIVERO
LASTENIA ULEAM EL CARMEN

ZAMBRANO ALCIVAR ANTHONY JAVIER
AUTOR/ES

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 **ZAMBRANO ALCIVAR ANTHONY JAVIER**, legalmente matriculado en la carrera de Software, período académico 2026-2027, cumpliendo el total de 384 horas, cuyo tema del proyecto es **"SISTEMA EMBEBIDO CON ARQUITECTURA IOT PARA EL MONITOREO DE ESTACIÓN METEOROLÓGICA 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, 24 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

Título de Trabajo de Titulación: SISTEMA EMBEBIDO CON ARQUITECTURA
IOT PARA EL MONITOREO DE ESTACIÓN METEOROLÓGICA EN EL
VIVERO LASTENIA ULEAM EL CARMEN

Modalidad: Proyecto Integrador

Autor: Zambrano Alcivar Anthony Javier

Tutor: Ing. Arévalo Hermida Rómulo Danilo

Tribunal de sustentación:

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

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

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

Fecha de sustentación:

26 de agosto del 2026

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



DECLARACIÓN DE AUTORÍA

La responsabilidad del contenido de este Trabajo de titulación, cuyo tema es: Sistema embebido con arquitectura IoT para el monitoreo de estación meteorológica en el Vivero Lastenia Uleam El Carmen, corresponde exclusivamente a: Zambrano Alcivar Anthony Javier con CI. 1351808744, 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 "Anthony", is positioned above a horizontal line.

Zambrano Alcivar Anthony Javier

C.I. 1351808744

DEDICATORIA

Con inmensa gratitud dedico este proyecto de titulación, en primer lugar, a Dios, por iluminar mi camino, darme la fortaleza y sabiduría para superar cada desafío y permitirme alcanzar esta importante meta de mi vida. A mis padres, por su amor incondicional, sacrificio y apoyo constante, pilares fundamentales en mi formación personal y profesional. A mi familia, por su comprensión, confianza y palabras de aliento durante todo este proceso. Finalmente, me dedico este logro a mí mismo por la perseverancia, el esfuerzo y la determinación que me permitieron convertir este sueño en una realidad.

Anthony Zambrano Alcivar.

AGRADECIMIENTO

Expreso mi más profundo agradecimiento a Dios, por concederme salud, fortaleza, sabiduría y sabiduría para superar cada desafío y permitirme alcanzar esta importante meta académica. A mis padres, por su amor incondicional, sacrificio, apoyo constante y confianza en cada etapa de mi formación. Su ejempló y motivación fueron fundamentales para hacer realidad este logro.

A mi tutor de tesis, Ing. Rómulo Danilo Arévalo Hermida, Mg, por su orientación, paciencia y compromiso durante el desarrollo de este proyecto, compartiendo su conocimiento profesional que contribuyo con mi formación profesional. Finalmente, agradezco a la Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen, y a sus docentes de la carrera de Software, por brindarme los conocimientos, las experiencias y el apoyo necesario para culminar con éxito esta importante etapa se mi vida.

Anthony Zambrano Alcivar.

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	XX
ÍNDICE DE ANEXOS	XXII
RESUMEN	XXIII
ABSTRACT.....	XXIV
CAPÍTULO I	1
1 INTRODUCCIÓN	1
1.1 Introducción	1
1.2 Presentación del tema.....	2
1.3 Ubicación y contextualización de la problemática.....	2
1.4 Planteamiento del problema	3
1.4.1 Problematización.....	3
1.4.2 Génesis del problema.....	3

1.5	Diagrama causa – efecto del problema	4
1.6	Objetivos	5
1.6.1	Objetivo general.....	5
1.6.2	Objetivos específicos	5
1.7	Justificación.....	6
1.8	Impactos esperados	6
1.8.1	Impacto tecnológico.....	6
1.8.2	Impacto social	6
1.8.3	Impacto ecológico.....	7
CAPÍTULO II		8
2	MARCO TEÓRICO.....	8
2.1	Antecedentes históricos.....	8
2.1.1	Sistemas embebidos.....	8
2.1.2	Monitoreo de estación meteorológica.....	8
2.2	Antecedentes de investigaciones relacionadas al tema presentado	9
2.3	Definiciones conceptuales.....	10
2.3.1	Sistema embebido	10
2.3.1.1	Características.....	11
2.3.1.2	Clasificación	11
2.3.1.3	Arquitectura de los sistemas embebidos.....	12
2.3.1.4	Componentes electrónicos de interfaz y acondicionamiento de señal	12

2.3.1.5	Microcontroladores: Arquitectura y Organización interna.....	13
2.3.1.6	Plataformas de Hardware libre: Arduino.....	14
2.3.1.7	Programación y Software libre para sistemas embebidos	14
2.3.2	Arquitectura IoT.....	14
2.3.2.1	Métricas de diseño: potencia, rendimiento y seguridad	15
2.3.3	Monitoreo de estación meteorológica.....	16
2.3.3.1	Fundamentos del monitoreo meteorológico con IoT.....	16
2.3.3.2	Sensores y recolección de datos meteorológicos.....	17
2.3.3.3	Procesamiento y análisis de datos meteorológicos.....	17
2.3.3.4	Calibración y mantenimiento de sensores meteorológicos.....	18
2.3.3.5	Almacenamiento y gestión de datos meteorológicos	19
2.3.3.6	Conectividad y transmisión de datos meteorológicos	19
2.3.3.7	Teledetección y monitoreo remoto del ambiente	20
2.3.3.8	Gemelos digitales y automatización en monitoreo meteorológico.....	21
2.3.4	Metodología de desarrollo ágil (Scrum)	21
2.4	Conclusiones del marco teórico	22
CAPÍTULO III.....		24
3 MARCO INVESTIGATIVO		24
3.1	Introducción	24
3.2	Tipos de investigación.....	24
3.2.1	Investigación de Campo.....	24

3.2.2	Investigación bibliográfica.....	25
3.2.3	Investigación descriptiva	25
3.2.4	Métodos de investigación	25
3.2.4.1	Método experimental.....	25
3.2.4.2	Método analítico	26
3.3	Fuentes de información de datos.....	26
3.3.1	Fuentes primarias.....	26
3.3.1.1	Encuestas	27
3.3.1.2	Entrevista.....	27
3.4	Estrategia operacional para la recolección de datos.....	28
3.4.1	Población.....	28
3.4.2	Muestra censal	28
3.4.3	Análisis de las herramientas de recolección de datos a utilizar	29
3.4.3.1	Encuesta.....	29
3.4.3.2	Entrevista.....	29
3.4.3.3	Estructura de los instrumentos de recolección de datos aplicados	29
3.4.4	Plan de recolección de datos	30
3.5	Análisis y presentación de resultados.....	31
3.5.1	Presentación y descripción de los resultados obtenidos	31
3.5.1.1	Análisis de encuesta realizada a Estudiantes que realizan prácticas en el vivero	31
3.5.1.2	Análisis de entrevista realizada a Ing. Pedro Nivela	34

3.5.2	Informe final del análisis de los datos.....	36
CAPÍTULO IV.....		38
4 MARCO PROPOSITIVO		38
4.1	Introducción	38
4.2	Descripción de la propuesta	38
4.3	Determinación de recursos	39
4.3.1	Humanos	39
4.3.2	Tecnológicos.....	39
4.3.3	Económicos.....	40
4.4	Desarrollo del software mediante metodología Scrum	41
4.4.1	Descripción del producto	41
4.4.1.1	Propósito del producto.....	41
4.4.1.2	Funcionalidades Claves	42
4.4.1.3	Usuario objetivo	42
4.4.1.4	Criterios de éxito del producto	42
4.4.2	Requerimientos del sistema	43
4.4.2.1	Historia de usuario.....	43
4.4.2.2	Requerimientos funcionales	45
4.4.2.3	Requerimientos no funcionales	45
4.4.3	Diseño del Sistema.....	46
4.4.3.1	Casos de Uso	46

4.4.3.2	Diagramas de Estado	49
4.4.4	Arquitectura del sistema	49
4.4.4.1	Arquitectura general	49
4.4.4.2	Tecnologías utilizadas	50
4.4.4.3	Roles Scrum.....	51
4.4.4.4	Definición de Hecho (DoD)	51
4.4.4.5	Criterios Generales	51
4.4.4.6	Criterios Específicos del Proyecto.....	52
4.4.5	Product Backlog.....	52
4.4.6	Sprint 1: Medición automática de datos.....	54
4.4.6.1	Planificación del Sprint	54
4.4.6.2	Sprint Backlog.....	55
4.4.6.3	Desarrollo del Sprint.....	56
4.4.6.4	Sprint Review	57
4.4.6.5	Pruebas del Sprint.....	58
4.4.6.6	Incremento y Entregables del Sprint	60
4.4.6.7	Estado actual del Product Backlog	61
4.4.7	Sprint 2: Procesamiento y validación de datos	61
4.4.7.1	Planificación del Sprint	61
4.4.7.2	Sprint Backlog.....	63
4.4.7.3	Desarrollo del Sprint.....	63

4.4.7.4	Sprint Review	64
4.4.7.5	Pruebas del Sprint.....	66
4.4.7.6	Incremento y Entregables del Sprint	68
4.4.7.7	Estado actual del Product Backlog	68
4.4.8	Sprint 3: Detección de Anomalías	69
4.4.8.1	Planificación del Sprint	69
4.4.8.2	Sprint Backlog.....	70
4.4.8.3	Desarrollo del Sprint.....	71
4.4.8.4	Sprint Review	71
4.4.8.5	Pruebas del Sprint.....	73
4.4.8.6	Incremento y Entregables del Sprint	74
4.4.8.7	Estado actual del Product Backlog	75
4.4.9	Sprint 4: Almacenamiento temporal de datos	75
4.4.9.1	Planificación del Sprint	75
4.4.9.2	Sprint Backlog.....	77
4.4.9.3	Desarrollo del Sprint.....	77
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	81
4.4.10	Sprint 5: Generación de Datos Estructurados en Formato JSON.....	81

4.4.10.1	Planificación del Sprint	81
4.4.10.2	Sprint Backlog	83
4.4.10.3	Desarrollo del Sprint.....	84
4.4.10.4	Sprint Review	84
4.4.10.5	Pruebas del Sprint.....	86
4.4.10.6	Incremento y Entregables del Sprint	87
4.4.10.7	Estado actual del Product Backlog	88
4.4.11	Resultado Final del Producto	88
4.4.11.1	Funcionalidades implementadas.....	88
4.4.11.2	Evidencia final del sistema	89
4.4.11.3	Conclusiones del desarrollo.....	91
CAPÍTULO V		93
5 EVALUACIÓN DE RESULTADOS		93
5.1	Introducción	93
5.2	Presentación y monitoreo de resultados	93
5.2.1	Planificación de la evaluación	93
5.2.2	Ejecución del monitoreo	94
5.2.2.1	Proceso 1 : Adquisición automática de datos	94
5.2.2.2	Proceso 2: Procesamiento y validación de datos	96
5.2.2.3	Proceso 3: Detección de anomalías	97
5.2.2.4	Proceso 4: Almacenamiento en buffer circular	98

5.2.2.5	Proceso 5: Generación del payload JSON	99
5.3	Interpretación objetiva.....	100
CAPÍTULO VI.....		102
6 CONCLUSIONES Y RECOMENDACIONES		102
6.1	Conclusiones	102
6.2	Recomendaciones.....	103
BIBLIOGRAFÍA		104
ANEXOS		110
GLOSARIO		115

ÍNDICE DE TABLAS

Tabla 1: Plan de recolección de datos.....	30
Tabla 2: Resultados Obtenidos en la Encuesta	31
Tabla 3: Resultados Obtenidos Entrevista.	34
Tabla 4: Recursos humanos	39
Tabla 5: Recursos tecnológicos	40
Tabla 6: Recursos económicos.....	41
Tabla 7: Usuario objetivo.....	42
Tabla 8: Historia de usuario 1: Medición automática	43
Tabla 9: Historia de usuario 2: Procesamiento y validación de datos	43
Tabla 10: Historia de usuario 3: Detección de anomalías.....	44
Tabla 11: Historia de usuario 4: Almacenamiento temporal de datos	44
Tabla 12: Historia de usuario 5: Generación de datos en formato JSON	45
Tabla 13: Tecnología utilizadas	50
Tabla 14: Roles Scrum.....	51
Tabla 15: Product Backlog.....	52
Tabla 16: Planificación del Sprint 1.....	54
Tabla 17: Sprint Backlog del Sprint 1.....	55
Tabla 18: Pruebas de caja negra – Sprint 1	58
Tabla 19: Pruebas de caja blanca – Sprint 1	59
Tabla 20: Estado actual del Product Backlog – Sprint 1	61

Tabla 21: Planificación del Sprint 2.....	61
Tabla 22: Sprint Backlog – Sprint 2	63
Tabla 23: Pruebas de caja negra – Sprint 2.....	66
Tabla 24: Pruebas de caja blanca – Sprint 2	67
Tabla 25: Estado actual del Product Backlog – Sprint 2	68
Tabla 26: Planificación del Sprint 3.....	69
Tabla 27: Sprint Backlog – Sprint 3	70
Tabla 28: Pruebas de caja negra – Sprint 3.....	73
Tabla 29: Pruebas de caja blanca – Sprint 3	74
Tabla 30: Estado actual del Product Backlog – Sprint 3	75
Tabla 31: Planificación del Sprint 4.....	75
Tabla 32: Sprint Backlog – Sprint 4	77
Tabla 33: Pruebas de caja negra – Sprint 4.....	79
Tabla 34: Pruebas de caja blanca – Sprint 4	80
Tabla 35: Estado actual del Product Backlog – Sprint 4	81
Tabla 36: Planificación del Sprint 5.....	81
Tabla 37: Sprint Backlog – Sprint 5	83
Tabla 38: Pruebas de caja negra – Sprint 5.....	86
Tabla 39: Pruebas de caja blanca – Sprint 5	86
Tabla 40: Estado actual del Product Backlog – Sprint 5	88
Tabla 41:Ejecución de adquisición de datos	95

Tabla 42: Ejecución de procesamiento y validación de datos	97
Tabla 43: Ejecución de la Detección de Anomalías	98
Tabla 44: Almacenamiento en buffer circular	98
Tabla 45: Ejecución de la generación del JSON.....	99

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Árbol del problema	4
Ilustración 2: Diagrama de Caso de Uso: Lectura de datos meteorológicos	46
Ilustración 3: Diagrama de Caso de Uso: Procesamiento inteligente de datos	46
Ilustración 4: Diagrama de Caso de Uso: Procesamiento inteligente de datos	47
Ilustración 5: Diagrama de secuencia: Lectura de Datos Meteorológico	47
Ilustración 6: Diagrama de secuencia: Procesamiento inteligente de Datos.....	48
Ilustración 7: Diagrama de secuencia: Generación de datos JSON	48
Ilustración 8: Diagrama de estado: Nodo Meteorológico	49
Ilustración 9: Arquitectura general	50
Ilustración 10: Entorno de desarrollo Utilizado	56
Ilustración 11: Adquisición automática de datos	57
Ilustración 12: Sensor DHT22 conectado al ESP32	57
Ilustración 13: Función de validación de datos.....	64
Ilustración 14: Algoritmo de filtrado por media recortada	64
Ilustración 15: Verificación de la validez de las mediciones.....	65
Ilustración 16: Comparación de mediciones para la detección de anomalías.....	71
Ilustración 17: Implementación de la función de detección de anomalías	72
Ilustración 18: Implementación del buffer circular para el almacenamiento de mediciones ..	78
Ilustración 19: Implementación de la función de almacenamiento en el buffer circular	78
Ilustración 20: Construcción del mensaje en formato JSON	84

Ilustración 21: Implementación de la función de generación del payload JSON	85
Ilustración 22: Función de adquisición automática de datos	89
Ilustración 23: Función de procesamiento y validación de datos	90
Ilustración 24: Detección de anomalías	90
Ilustración 25: Almacenamiento en buffer circular	91
Ilustración 26: Generación de payload JSON	91
Ilustración 27: Planificación de la evaluación	93
Ilustración 28: Evidencia de funcionamiento de toma de datos	95
Ilustración 29: Evidencia de procesamiento y validación de datos	96
Ilustración 30: Evidencia de detección de anomalías	97
Ilustración 31: Evidencia generación del JSON	99

ÍNDICE DE ANEXOS

Anexo A: Asignación de tutor	110
Anexo B: Fotografías.....	111
Anexo C: Formato encuesta.....	112
Anexo D: Formato entrevista.....	113
Anexo E: Reporte del sistema antiplagio.....	114

RESUMEN

La presente investigación, surge ante la necesidad de mejorar el monitoreo climático en el Vivero Lastenia Vera de la Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen, dado que la falta de información climatológica en tiempo real obstaculiza la toma de las decisiones para el manejo del cultivo de cacao, favoreciendo así la aparición de enfermedades fúngicas y de la pérdida en la producción. El objetivo general consistió en implementar una estación meteorológica basada en IoT con un sistema embebido para la obtención de datos climatológicos en tiempo real, orientada al monitoreo climático del cultivo de cacao en el vivero. La investigación se realizó bajo un enfoque mixto, considerando una población de 90 personas (89 estudiantes y 1 docente); a los 89 estudiantes se aplicó la encuesta en su totalidad mediante un censo, obteniéndose 89 respuestas válidas, y al docente se le realizó una entrevista como informante clave. Como propuesta tecnológica se realizó una estación meteorológica basada en un microcontrolador ESP32, programada en C++ mediante PlatformIO, capaz de obtener, validar, procesar y almacenar mediciones de temperatura y humedad, detectar anomalías de forma local y generar datos estructurados en formato JSON con el número de secuencia, la temperatura y la humedad para su integración con la arquitectura IoT. La implementación se realizó bajo la metodología Scrum a lo largo de cinco sprints, permitiendo la implementación incremental de las funcionalidades. La información recabada en las pruebas de caja negra y caja blanca evidenció el correcto funcionamiento de los módulos desarrollados, cumpliéndose los criterios de aceptación establecidos en la historia de usuario. Los hallazgos demuestran que la propuesta constituye una alternativa tecnológica confiable para automatizar el monitoreo climático y apoyar la toma de decisiones en el cultivo de cacao.

ABSTRACT

This research arises from the need to improve climate monitoring in the Lastenia Vera Nursery of the Eloy Alfaro Lay University of Manabí, El Carmen Extension, since the lack of climatological information in real time hinders decision-making for the management of cocoa crops, thus favoring the appearance of fungal diseases and loss in production. The general objective was to implement an IoT-based weather station with an embedded system for obtaining real-time weather data, aimed at climate monitoring of cocoa cultivation in the nursery. The research was carried out under a mixed approach, considering a population of 90 people (89 students and 1 teacher); the survey was applied to all 89 students through a census, obtaining 89 valid responses, and an interview was conducted with the teacher in charge of the nursery as a key informant. As a technological proposal, a weather station based on an ESP32 microcontroller was made, programmed in C++ using PlatformIO, capable of obtaining, validating, processing and storing temperature and humidity measurements, detecting anomalies locally and generating structured data in JSON format with sequence number, temperature and humidity for integration with the IoT architecture. The implementation was carried out under the Scrum methodology over five sprints, allowing the incremental implementation of the functionalities. The information collected in the black box and white box tests evidenced the correct functioning of the developed modules, meeting the acceptance criteria established in the user history. The findings show that the proposal constitutes a reliable technological alternative to automate climate monitoring and to support decision-making in cocoa farming.

CAPÍTULO I

1 INTRODUCCIÓN

1.1 Introducción

Actualmente, el desarrollo tecnológico ha tenido un impulso en la creación de sistemas cada vez más eficientes para la obtención y el análisis de los datos ambientales, considerando los sistemas embebidos con arquitectura IoT uno de los principales inventos en este campo. Estos sistemas incorporan componentes electrónicos programables con sensores y módulos de transmisión que permite obtener, procesar y transmitir información en tiempo real. Estas implementaciones se han expandido hacia diversas áreas, entre ellas destaca la agricultura y la gestión ambiental, donde la automatización y el constante monitoreo remoto son uno de los factores determinantes para la toma de decisiones y la optimización de los recursos.

El constante uso de sistemas embebidos con arquitectura IoT representa una solución moderna y de muy bajo costo para su implementación como herramientas tecnológicas capaces de procesar datos de manera local y enviarlos a sitios digitales para su visualización o análisis posterior. Su diseño modular y la capacidad de conexión entre redes inalámbricas ofrecen una mayor flexibilidad para el crecimiento del proyecto orientado al control y monitoreo de las condiciones climatológicas, tales como la temperatura, humedad, presión atmosférica o niveles de precipitación. Estos dispositivos se caracterizan por su menor consumo energético y por la capacidad de operar en entornos tales como el campo, los que lo convierte en una de las alternativas ideales para aplicaciones en viveros o cualquier tipo de cultivo agrícola.

Esta combinación hace que el sistema embebido con arquitectura IoT y el monitoreo meteorológico automatizado busque integrar tecnología sostenible, brindado así una herramienta innovadora que pueda manejar eficientemente la información climática en el vivero Lastenia Vera. Mediante este enfoque, se busca optimizar el control de las variables climatológicas que inciden en el desarrollo del cultivo, para así promover la digitalización en las actividades agrícolas de la zona, contribuyendo el fortalecimiento de las buenas prácticas productivas, basándose en la ciencia y la tecnología. Dicho proyecto busca contribuir a la adopción de herramientas tecnológicas entre los pequeños agricultores del cantón El Carmen, y así fortaleciendo el sector agroindustrial, aportando una solución práctica y replicable.

1.2 Presentación del tema

Los constantes cambios climáticos y las variaciones en las condiciones atmosféricas han creado la necesidad de contar con herramientas tecnológicas que ayuden a monitorear estos cambios. Hoy en día existen estaciones meteorológicas de costos muy elevados que al implementarse en el sector rural agrícola de la zona se ven muy limitada. Considerando este panorama, surge la necesidad de crear una solución que sea accesible basándose en las tecnologías emergentes tales como el Internet de las Cosas (IoT), que facilite la recolección, transmisión y análisis de los datos climatológicos en tiempo real.

En el presente proyecto integrador titulado “Sistema Embebido con Arquitectura IoT para el Monitoreo de Estación meteorológica en el Vivero Lastenia Uleam El Carmen” se plantea el diseño e implementación de un prototipo funcional que facilite la medición de las variables climatológicas tales como la temperatura, humedad, presión atmosférica y velocidad del viento. De tal manera, se procura apoyar con la toma de decisiones en sectores agrícolas, esta gestión ambiental busca prevenir riesgos, beneficiando directamente a la comunidad y el fortalecimiento de la investigación científica del cantón.

1.3 Ubicación y contextualización de la problemática

El vivero Lastenia Vera se encuentra ubicado en la Av. Chone Km 33 y Calle Pedro Tobías Zambrano, El Carmen, Ecuador, donde anteriormente funcionaba el colegio Lastenia Vera Pinargote, fundado en enero de 1986, por un grupo de gestores locales que querían aportar con la educación del cantón ya que eran muy pocas las instituciones fiscales en el entonces. La antigua institución lleva su nombre en honor a Isabel Vera Pinargote, madre de Pedro Tobías Zambrano Vera, filántropo que en sus tiempos donó tres hectáreas de terreno para su construcción. En sus inicios funcionó como una Escuela Fiscal de Niñas llamada Luisa Vega Barberán, en jornada vespertina; no fue hasta septiembre de 1994, cuando inauguró sus propias instalaciones. A partir desde ese instante, el colegio ha experimentado diversos cambios en su infraestructura, personal docente y población estudiantil (Mendoza, 2012).

Actualmente las instalaciones del colegio Lastenia Vera son administradas por la Universidad Laica Eloy Alfaro de Manabí (Uleam) Extensión El Carmen, desde noviembre del 2024, ya que la institución secundaria dejó de funcionar. En este sitio la Uleam implementó un proyecto

denominado “Millón de plantas de cacao” debido a que a lo largo del tiempo El Carmen ha logrado consolidarse como un cantón platanero y cacaotero, sin embargo, enfrenta diferentes dificultades relacionadas directamente con el monitoreo climático que afecta al cultivo de cacao.

1.4 Planteamiento del problema

1.4.1 Problematicación

Hoy en día, la agricultura depende en gran medida de la disponibilidad de información precisa y oportuna sobre los constantes cambios climatológicos. No obstante, en zonas agrícolas del cantón El Carmen, existen problemas producidos por los cambios climáticos y lo que puede favorecer la proliferación de enfermedades fúngicas en el sector agrícola. De la misma manera provoca que los pequeños agricultores tomen decisiones erróneas lo que provoca pérdidas económicas en el cultivo y producción del cacao.

Dado este escenario emerge la necesidad de plantearse preguntas tales como: ¿Qué efectos puede causar el desconocimiento de condiciones climáticas en la producción de cacao?, ¿Por qué los agricultores aún dependen de la toma de decisiones empíricas sobre las actividades?, ¿Cómo se pueden minimizar costos en el cultivo y producción del cacao?

1.4.2 Génesis del problema

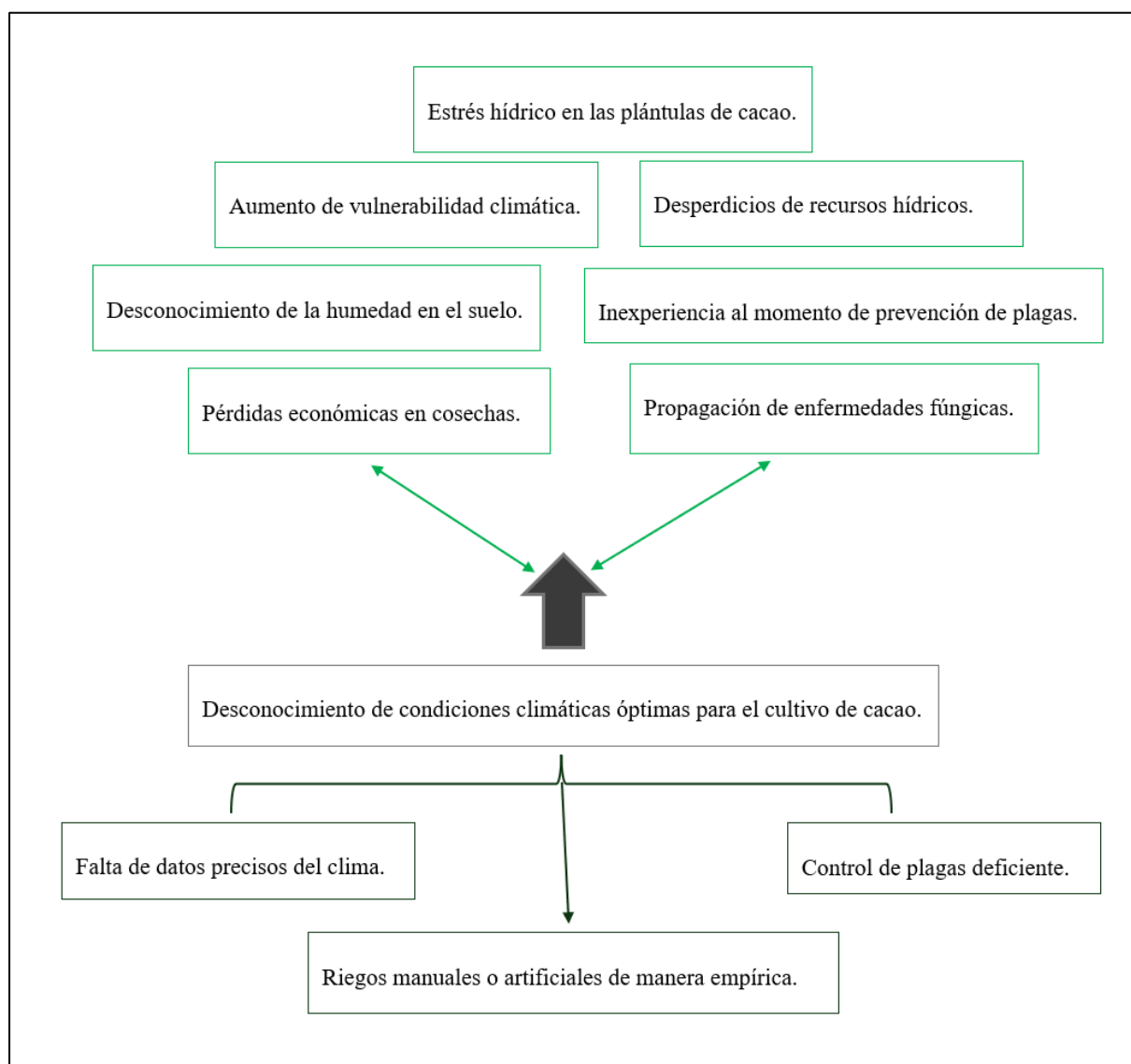
Según información proporcionada por la página web oficial de la Uleam, en noviembre del 2024 surge la necesidad de la creación de un vivero de cacao, para beneficiar e impulsar el crecimiento y desarrollo de la comunidad agrícola del cantón El Carmen. A raíz de ello surgen problemas en el cultivo de cacao, debido a que durante el proceso algunas plántulas se ven afectadas por enfermedades fúngicas, a raíz de la inexperiencia al momento de controlar y prevenir las plagas. Además de ello, el desconocimiento sobre las condiciones climáticas óptimas para el cultivo dificulta la obtención mejor resultados y pueden generar pérdidas significativas en el cultivo.

En el sector agrícola del cantón El Carmen, la falta de datos precisos del clima, los riegos manuales realizados de manera empírica y el control ineficiente de plagas pueden causar pérdidas en la cosecha, afectando así al ingreso económico de las familias campesinas de la

zona. Debido a eso, muchos agricultores dependen principalmente de las experiencias empíricas para la toma de sus decisiones sobre el cultivo. Uno de los factores más importantes es la humedad del suelo, que afectan directamente al cultivo, causando estreses hídricos en las plántulas de cacao y así pérdidas económicas en la producción debido al aumento de la vulnerabilidad climática. Es así como nace la necesidad de monitorear los constantes cambios climáticos en el vivero.

1.5 Diagrama causa – efecto del problema

Ilustración 1: Árbol del problema



1.6 Objetivos

1.6.1 Objetivo general

Implementar una estación meteorológica basada en IoT con un sistema embebido para la obtención de datos climatológicos en tiempo real, orientada al monitoreo climático del cultivo de cacao en el Vivero Lastenia El Carmen.

1.6.2 Objetivos específicos

Analizar la problemática actual que enfrenta la agricultura en el vivero Lastenia para la producción del cacao, debido a los diferentes cambios climáticos que se presentan en la zona.

Buscar cuidadosamente literatura basada en los conceptos, dispositivos de los sistemas empotrados y su combinación con la tecnología IoT, para el monitoreo climatológico.

Aplicar investigación de campo, que ayudará a identificar las condiciones ambientales dentro del vivero de cacao Lastenia Vera, en la zona del cantón El Carmen y el impacto que tiene en su producción.

Definir de manera simple y detallada, los requisitos sobre las necesidades locales del monitoreo del clima para el sector agrícola del cacao, para así determinar los parámetros meteorológicos más importantes.

Diseñar e implementar el sistema embebido de la estación meteorológica, mediante la integración del microcontrolador y los sensores ambientales, para realizar la adquisición, procesamiento y estructuración de los datos climatológicos en formato JSON, para su posterior utilización por el módulo de transmisión.

Validar el funcionamiento del sistema embebido mediante pruebas técnicas y de campo en el vivero Lastenia Vera, para verificar la correcta adquisición, procesamiento y estructuración de los datos climatológicos en formato JSON de acuerdo con los requisitos establecidos.

1.7 Justificación

El presente proyecto es esencial porque propone una solución tecnológica innovadora que permite automatizar y monitorear las variables del cambio climático mediante el uso de sistema embebido IoT. Esta herramienta facilitará la recolección de los datos de una manera precisa en tiempo real, esto representa un avance significativo frente a los métodos manuales, empíricos o tradicionales utilizados en la actualidad en la agricultura del cantón El Carmen. Esta implementación ayudará a mejorar eficientemente la toma de decisiones y en la planificación de las actividades del cultivo de cacao.

Desde una perspectiva diferente el desarrollo de esta estación meteorológica automatizada de bajo costo impulsará la aplicación práctica de la ingeniería de software y de la electrónica y así fomentar la integración entre hardware y software como una solución viable en procesos agrícolas. Además, promueve el uso de tecnologías IoT abierta y accesible, lo que permitirá a futuros estudiantes y agricultores o instituciones que puedan replicar este tipo de proyecto y adaptar el sistema según las necesidades.

1.8 Impactos esperados

1.8.1 Impacto tecnológico

En el impacto tecnológico este proyecto se muestra como una transformación de los métodos tradicionales de registros climatológicos hacia un sistema automatizado, que sea preciso y a su vez accesible. Esta implementación afecta positivamente la forma en que los agricultores y técnicos recopilan información, para así remplazar los procesos manuales y poco confiables por tecnología de monitoreo en tiempo real. A su vez, beneficiará al sector agrícola ofreciendo una herramienta moderna que facilita la toma de decisiones, y así reducir las pérdidas productivas, promoviendo la incorporación de soluciones sostenibles en el desarrollo tecnológico.

1.8.2 Impacto social

En el impacto social este proyecto se refleja en la mejora en la calidad de vida de los agricultores de El Carmen al facilitar el acceso a la información meteorológica confiable y actualizada. Esto afecta positivamente las buenas prácticas agrícolas tradicionales, y así

promover una cultura tecnológica orientada al uso de datos para la toma de decisiones en el cultivo. De igual manera se beneficia la población al minimizar las pérdidas económicas ya sea por fenómenos climáticos o enfermedades fúngicas, así fortaleciendo la productividad y generando mayores oportunidades de desarrollo en el sector agrícola local.

1.8.3 Impacto ecológico

Este proyecto se evidencia en la producción de prácticas agrícolas sostenibles mediante el uso eficiente de los recursos naturales. Esto afecta positivamente el manejo ambiental porque permite un control eficiente evitando el uso excesivo de agua y fertilizante. Además, beneficia al ecosistema reduciendo significativamente la degradación del suelo a causa de los químicos y el desperdicio de recursos hídricos, contribuyendo a la conservación del medio ambiente y al fortalecimiento de una agricultura más ecológica y responsable.

CAPÍTULO II

2 MARCO TEÓRICO

2.1 Antecedentes históricos

2.1.1 Sistemas embebidos

Uno de los principales sistemas embebidos surgió en 1974 con el Intel 8048, precedido por el Z80 de Zilog en el año de 1976 se utilizó el 6502 en las computadoras de Apple II. Durante los años 80 y 90 surgieron los microcontroladores de las empresas como Texas Instruments con el TMS3222010 y Microchip tales como el EEPROM. Uno de los cambios significativos que tuvo el paradigma llegó el año 2005 con Arduino, creado por Massimo Banzi, que democratizó la programación mediante placas de bajo costo y el lenguaje C++ simplificado. En el año 2012, Raspberry Pi lanzó una computadora de placa única que contaba con el sistema operativo de Linux, esto consiguió que las ventas de este alcanzaran los 12.5 millones de unidades. En el 2016, el ESP32 de la empresa Espressif revolucionó el IoT incorporando WiFi nativo para la comunicación inalámbrica (Salas Arriarán, 2025, p. 48).

2.1.2 Monitoreo de estación meteorológica.

La observación meteorológica ha venido evolucionando desde algunos años significativamente desde métodos rudimentarios basados en la apariencia del cielo hasta los sistemas automatizados modernos. En la creación de instrumentos meteorológicos tuvo inicio en el siglo XV, siendo así impulsado por las necesidades agrícolas de ese tiempo y la creciente navegación marítima, donde fue un factor muy importante predecir las condiciones climáticas, previniendo así tormentas mortales. En el año de 1441, el príncipe coreano Munjong creó el primer pluviómetro, seguido por inventos como el termómetro del agua en el año 1593, y el barómetro en 1643. El primer sistema que medía la humedad surgió el año 1664. No obstante, la transformación de las observaciones tradicionales a las automáticas mostró un desafío con interrupciones en el registro climatológico. Estas estaciones automáticas actuales superan limitaciones satelitales que ofrecen mayor densidad de observaciones, operación continua y reducción de errores humanos. Los paneles solares eliminaron la dependencia eléctrica, facilitando movilidad y la expansión global del monitoreo ambiental (Hecini & Hamaizia, 2020, p. 1).

2.2 Antecedentes de investigaciones relacionadas al tema presentado

El proyecto titulado Análisis, diseño, construcción e implementación de un sistema web para la gestión de datos meteorológicos mediante la utilización de una API, este estudio se realizó en Quito, Ecuador, este proyecto tiene como objetivo general realizar un sistema web open source que recoja datos meteorológicos de varios formatos a través de un API, utilizando la Metodología ágil SCRUM y las tecnologías tales como Node.js, MongoDB Atlas y la API de Weatherstack, este proyecto tiene como principal resultado un sistema web funcional con tiempo de carga de 2.98 segundos, con la capacidad de soportar 1000 usuarios concurrentes, implementación exitosa de consultas de clima actual y datos históricos de Ecuador, y con la funcionalidad de descargar dicha información en múltiples formatos tales como (CSV, JSON, XML, TXT, XLSX), (Constante Jauregui, 2021).

La tesis titulada Diseño e implementación de un prototipo de estación meteorológica agrícola autosustentable para el monitoreo de parámetros ambientales en cultivos de cacao mediante Raspberry Pi este estudio tuvo lugar en Guayaquil, Ecuador su objetivo general es diseñar e implementación de un prototipo de estación meteorológica agrícola autosustentable para el monitoreo de parámetros ambientales en cultivos de cacao mediante Raspberry Pi. Este proyecto implementó la metodología experimental con una arquitectura cliente-servidor con la tecnología IoT para la obtención de los datos meteorológicos mediante 8 sensores climatológicos. Este proyecto tuvo como resultado un sistema funcional autosustentable mediante energía solar, transmisión de los datos vía WiFi a la plataforma de ThingSpeak, con un sistema de alertas SMS mediante módulo GSM cuando se excede umbrales críticos, visualización web y móvil en tiempo real, con la recopilación exitosa de más de 60,575 registros durante cuatro meses de prueba en condiciones climatológicas reales en Guayaquil (López Vargas & Balladares Feijoo, 2021).

El proyecto titulado Diseño de un prototipo de sistema de control para temperatura y humedad del almacenamiento de arroz, se desarrolló en Guayaquil, Ecuador, su objetivo general es diseñar un sistema a escala para controlar dichas variables mediante LabVIEW 2019 y el hardware LattePanda. Se incorporó una metodología de control en lazo cerrado con sensores DHT22 y los motores DC que eran controlados por PWM. No obstante, se integró una interfaz SCADA en LabVIEW y una plataforma de monitoreo remoto mediante Ubidots. Estos resultados demostraron un control estable de temperatura a 30 °C y húmedas al 60% , y así

validando la efectividad del sistema en la conservación del arroz (Chichande Solano & Matamoros Galán, 2021).

Los presentes trabajos revisados guardan una estrecha relación con el presente estudio, ya que todos abordan la aplicación de tecnologías embebidas e IoT para el monitoreo de las variables ambientales, enfocándose especialmente en contexto agrícola. La primera tesis, se enfoca en el control de temperatura y humedad en el almacenamiento de arroz, este trabajo de investigación comparte con el presente trabajo la utilización de sistemas embebidos para la supervisión de las condiciones ambientales, aunque su propósito se limita al control en espacios cerrados. La segunda tesis, está orientada al diseño de una estación meteorológica agrícola autosustentables para el cultivo de cacao, coincide en el uso de sensores IoT y plataformas de bajo costo para la adquisición de datos climáticos, siendo así un antecedente directo en el ámbito de la agricultura de precisión. La tercera tesis, se centra en un sistema web para la distribución de datos meteorológicos mediante una API, integrándose así con el campo. Permitiendo así orientarse en la gestión y difusión de información del clima, aunque sin abordar el diseño físico de una estación ni el control de variables en tiempo real.

A diferencia de los estudios anteriores, este proyecto propone un sistema embebido IoT de monitoreo automatizado diseñado particularmente para una estación meteorológica en el vivero Lastenia Vera, esto incorpora la recolección, transmisión y procesamiento remoto de datos en tiempo real, con la finalidad de optimizar el seguimiento de condiciones atmosféricas locales. Este enfoque busca recopilar datos, sino también ofrecer una solución adaptable y escalable, y aplicable a diversos entornos agrícolas de la región.

2.3 Definiciones conceptuales

2.3.1 Sistema embebido

El sistema embebido o también llamado empotrado es un sistema electrónico creado para desarrollar funciones dedicadas al mismo, a diferencia con los dispositivos de propósitos general como una computadora que están diseñado para cubrir un amplio rango de necesidades. En este sistema todo tipo de componente se encuentra soldado en una misma placa (Sánchez Freire, 2023, p. 87).

2.3.1.1 Características

Su principal característica es miniaturizar su menor consumo de energía, su gran capacidad de procesamiento y por la interconexión entre ellos. Debido a su evolución, han aparecido nuevos conceptos tecnológicos, los cuales hace algunas décadas atrás eran inesperados. Algunos de ellos son: los dispositivos versátiles, domótica, teléfonos móviles, internet de las cosas, ciudades inteligentes, industria 4.0, entre otros (Valiente et al., 2021, pp. 12-13).

Según Paredes et al. (2024), menciona que todo tipo de sistema embebido cuentan con 4 principales características: **Fiabilidad:** estos sistemas embebidos están diseñados para representar máxima fiabilidad y además procesar datos sin fallas todo el tiempo. **Rendimiento:** a pesar de los principios del diseño del sistema en un chip y contando con la posibilidad de particionar el sistema, el rendimiento puede ser adaptado a una aplicación objetivo. **Costo:** los sistemas embebidos son accesible a cualquier persona debido a que no requieren capacidades sofisticadas en la fabricación. **Tamaño:** estos sistemas pueden ser reducidos a un tamaño muy pequeño, lo que facilita su utilización en dispositivos celulares.

2.3.1.2 Clasificación

De acuerdo con Nureni (2025), los sistemas embebidos se clasifican considerando varios factores, como los requisitos de rendimiento y funcionalidad, rendimiento del microcontrolador, y el comportamiento determinista del sistema. A continuación, varias clasificaciones: Capacidad de procesamiento del microcontrolador, y del nivel de complejidad requerida por la aplicación. Los sistemas de pequeñas escalas, basada en microcontroladores de 8 a 16 bits, operan con baja potencia, y recursos muy limitados y sin sistema operativo. Porque lo emplean en tareas simples como juguetes o reloj digital. Además de ello la clasificación basada en comportamiento determinista que distingue sistemas de tiempo real duro, en los que el incumplimiento de un plazo puede causar fallos críticos, de sistemas real blando, donde se toleran pequeñas desviaciones temporales sin afectar el funcionamiento general del mismo.

2.3.1.3 Arquitectura de los sistemas embebidos

La arquitectura de un sistema embebido se refiere a la organización ordenada, funcional de un conjunto de hardware y software que permite ejecutar tareas específicas dentro de un dispositivo. Según Jagtap (2018), un sistema embebido puede operar de manera independiente o ser parte de un sistema mayor. Esta arquitectura se define como los componentes interactúan para controlar, procesar y responden en tiempo real a las entradas del entorno. La arquitectura de un sistema embebido está conformada por tres elementos fundamentales que son hardware, software y el sistema operativo. Además de ello menciona que existen dos tipos de arquitectura entre ellas están: Arquitectura Von Neumann, en la cual las instrucciones y los datos comparten el mismo bus de memoria lo que es una limitante en la ejecución simultánea de los procesos. Mientras que en la arquitectura Harvard, separa la memoria y los buses de datos e instrucciones, permitiendo así el acceso en paralelo y mayor velocidad de procesamiento.

2.3.1.4 Componentes electrónicos de interfaz y acondicionamiento de señal

2.3.1.4.1 Selección de sensores

La selección de sensores es esencial en sistemas IoT, considerando los parámetros con precisión, sensibilidad, rango y deriva, además de los factores como el costo, la disponibilidad y el tiempo de entrega. Estos sensores pueden entregar salidas analógicas (0-5V, 4-20mA, señales de audio) o digitales (LOW/HIGH en protocolos TTL, I2C, SPI). Estas salidas analógicas requieren conversiones ADC tras acondicionamientos de señal, mientras tanto que las digitales se enlazan directamente al procesador. Estos sensores inteligentes incluyen procesamiento embebido para las tareas complejas de censado y actuación (James et al., 2021, pp. 35-36).

2.3.1.4.2 Circuitos de interfaz electrónica

Los circuitos de interfaz toman las señales del sensor al procesador embebido, desarrollando funciones como amplificación, autenticación, filtrado de ruido y conversión de impedancia. Es importante ajustar el voltaje de salida del sensor al voltaje nominal del microcontrolador (típicamente 3.3V o 5V) mediante amplificadores de alta impedancia basados en amplificadores operacionales. Estos amplificadores (pasa-bajos, pasa-altos, pasa-banda) por lo que eliminan ruidos de señales medias, y así pudiendo implementar analógica o digitalmente.

Dando así un diseño apropiado de interfaz garantizando la salida reproducible y señales adecuadas para el procesamiento, medición y almacenamiento de información (James et al., 2021, p. 37).

2.3.1.5 Microcontroladores: Arquitectura y Organización interna

2.3.1.5.1 Arquitectura de los microcontroladores

La arquitectura de los microcontroladores está constituida en un solo circuito componentes similares a los de una computadora, tales como la CPU, memoria RAM, EEPROM y periféricos desde operaciones simples hasta sistemas complejos de control. Esta propuesta ha venido evolucionando a partir de la arquitectura de Von Neuman, que anteriormente utilizaba una única memoria para los datos instrucciones, mientras que la arquitectura Harvard separa ambas memorias, admitiendo una mayor velocidad y paralelismo en la gestión de datos. Gran parte de los microcontroladores modernos adquieren la arquitectura RISC, se destaca por un conjunto reducido de instrucciones que se ejecutan con una mayor eficiencia. Asimismo, cuentan con una memoria no volátil para almacenar el programa principal, memoria RAM o EEPROM para los datos, y con líneas de entrada/salida (E/S) para tener la comunicación entre sensores y periféricos. Debido a ello incorporan recursos auxiliares como temporizadores, los convertidores A/D y los módulos de comunicación, que aumentan su capacidad y versatilidad en los sistemas embebidos IoT (Díaz Ronceros, 2020).

2.3.1.5.2 Organización de los microcontroladores

Los microcontroladores están organizados por bloque internos como la CPU, la memoria, los registros los buses, los periféricos de entrada y salida, que trabajan en conjunto para ejecutar las instrucciones del programa almacenado. Esta CPU realiza las etapas de captura, la decodificación y ejecución, el apuntador de la pila y de unidad aritmética y lógica. Conforme a su arquitectura, los microcontroladores pueden ser CISC o RISC, siendo esto último lo más eficiente para tareas específicas. En los sistemas embebidos IoT, esta organización permite el procesamiento autónomo de los datos de sensores y la comunicación con plataformas externas sin depender de un equipo central (Espinosa, 2021, p. 19).

2.3.1.6 Plataformas de Hardware libre: Arduino

Arduino es una plataforma de hardware libre que combina diversos componentes electrónicos y software de programación para la creación de proyectos interactivos. Su principal objetivo es facilitar el uso de la electrónica y de los sistemas embebidos en el ámbito educativo, científico e incluso en la industria. Y las placas de Arduino, como la UNO, Mega o Ethernet, incorporan microcontrolador que permiten leer, los datos de los sensores y controlar actuadores a través de instrucciones programadas. Gracias a su arquitectura abierta, cualquier persona puede fabricar sus propias placas y modificar su código fuente. No obstante, Arduino se puede programar desde un entorno de desarrollo libre llamado (IDE Arduino), permitiendo así la conexión entre distintos sensores, módulos y sistemas IoT (Sánchez Freire, 2023, pp. 89-90).

2.3.1.7 Programación y Software libre para sistemas embebidos

El software libre desempeña un papel importante en la programación de sistemas embebido, ya que promueven la accesibilidad y la personalización de las herramientas educativas y del desarrollo. Uno ejemplo que ha sido destacado es el de eUCCvm, una máquina virtual de código abierto diseñada por la Universidad Tecnológica Nacional de Mendoza para la enseñanza de la programación en sistemas embebidos. Dicha herramienta permite compilar, además de ello simular programas en múltiples arquitecturas como Arduino, PIC, ESP32 y Raspberry PI, empleando lenguajes como C y ensamblador. Esta implementación en los entornos educativos facilitan el aprendizaje tanto presencial como virtual, sin necesidad de requerir un equipo avanzado (Pérez Monte et al., 2024).

2.3.2 Arquitectura IoT

El internet de las cosas (IoT) se define como la interconexión de un objeto físico a través de internet, permitiendo el intercambio de los datos entre los dispositivos y sistemas inteligentes. Este enfoque, introducido por Kevin Ashton en 1999, evolucionó significativamente en los años 2008 y 2009 con el desarrollo de tecnologías RFID y M2M. Esta arquitectura se basa en redes globales que manejan protocolo HTTP y SMTP, posibilitando la comunicación autónoma entre equipos. En la actualidad, existen miles de millones de dispositivos conectados que transforman sectores como la educación, la salud y la industria, optimizando recursos (Salazar & Silvestre, 2015).

2.3.2.1 Métricas de diseño: potencia, rendimiento y seguridad

2.3.2.1.1 Potencia

La eficiencia y potencia de los sistemas embebidos se centra en reducir el consumo energético sin afectar el rendimiento funcional del hardware y software. Investigaciones recientes destacan la utilización escalada y dinámica de voltajes y frecuencia (DVFS) como una técnica clave para ajustar el consumo según la carga de trabajo, y así logrando ahorros de energía de un 47%. No obstante, los diseños de bajo consumo a nivel de procesador y de microarquitectura permiten la optimización de la energía en los dispositivos portátiles e IoT. La gestión dinámica de la energía (DPM) apaga componentes inactivos, mientras que la medición precisa del consumo garantizando la evaluación verídica del desempeño energético. Dicha estrategia combinada beneficia a los sistemas embebidos más sostenibles, confiables y con una mayor autonomía operativa (Sofianidis et al., 2025).

2.3.2.1.2 Rendimiento

El rendimiento establece una métrica fundamentalmente en el diseño de los sistemas embebidos IoT, ya que un microcontrolador ejecuta las instrucciones y procesa dichos datos. Según Hu et al. (2022), el rendimiento puede ser mejorado mediante arquitectura optimizadas que reduzcan los ciclos de espera y aumentas dicha velocidad de ejecución. A pesar de que el procesamiento fuera de orden permite incrementar la eficiencia hasta en un 33.3% mejorando así la fluidez del flujo de instrucciones. El rendimiento evalúa la capacidad del hardware para realizar tareas en el menor tiempo posible, para así utilizar menos recursos de manera óptima. En consecuencia, de esto al medir y mejorar el rendimiento, para así garantizar la funcionalidad y las respuestas eficientes de los sistemas embebidos, al momento del monitoreo automatizado.

2.3.2.1.3 Seguridad

Los sistemas embebidos son demasiado vulnerables a una gran cantidad de ataques que pueden afectar su seguridad. Uno de los ataques más comunes es: ataque de agotamiento de recursos, este ataque podría consumir la energía necesaria que necesita en sistema para realizar tareas computacionales o en el uso de periféricos o sensores. No obstante, son vulnerables a los ataques físicos, los cuales tienen acceso a la parte física del sistema y podrían realizar una instrucción, que busca alterar su integridad o realizar ataque de espionaje en el bus del sistema,

causando daños a los sensores o periféricos. Sin embargo, este tipo de sistema ya sea integrado o convencional, comparten un mismo objetivo de seguridad genérica que son: la confidencial, la integridad y disponibilidad. Además suelen ser atacados e infectados por programa maligno que son uno de los ataques más comunes que inteligentemente son capaces de vulnerar cualquier tipo de acceso robando así información importante del usuario (Aloseel et al., 2021, pp. 961-982).

2.3.3 Monitoreo de estación meteorológica.

El monitoreo remoto del ambiente consiste en la observación, medición y análisis continuo de los sistemas ambientales, por medio del uso de sensores ubicados a ciertas distancias del suelo, esto permite abarcar amplias áreas geográficas para obtener datos de las variables físicas o biológicas. Este proceso es conocido como teledetección, permite capturar imágenes o señales del entorno terrestre o acústicos a través de distintas plataformas, como satélites, aeronaves o vehículos aéreos no tripulados llamados (UAV), que registran la reflexión y la absorción electromagnética de la luz solar en la superficie del suelo, la vegetación y el agua (Acevedo, 2015, p. 185).

2.3.3.1 Fundamentos del monitoreo meteorológico con IoT

La recolección de los datos en la estación meteorológica IoT se enfoca en la medición continua de los parámetros como temperatura, humedad, presión, viento y precipitación. Estos sensores deben estar calibrados y su salida filtrada para la reducción de ruido y detectar valores atípicos. La frecuencia de muestreo se ajusta según el parámetro y la necesidad de resolución, así equilibrando la precisión durante el consumo energético. Las lecturas se registran con marcas de tiempo en formato ISO 8601 y son almacenados localmente para posteriormente ser transmitidos al servidor mediante los protocolos MQTT o HTTP. Es esencial contener metadatos; coordenadas, altura del sensor, versión de firmware para la trazabilidad. Estos datos pasan por un filtro de comprobaciones antes de ser analizados o visualizados. No obstante, se deben mantener registros de los fallos y reintentos para minimizar la pérdida de la información (Shovic, 2021).

2.3.3.2 Sensores y recolección de datos meteorológicos

Los sistemas meteorológicos basados en IoT manejan sensores electrónicos de bajo costo para registrar las variables ambientales claves, tales como la temperatura humedad, presión atmosférica, los cuales son gestionados por los microcontroladores que son capaces de procesar y transmitir datos en tiempo real. Dichos dispositivos, apoyados de un protocolo de comunicación como lo es MQTT, que permite generar un flujo continuo y confiable de información que pueda ser empleado para el análisis ambiental, predicción de las condiciones climatológicas o evaluación de la calidad del aire. No obstante, de su fácil uso combinado de sensores calibrados y algoritmos de inferencia mejora la precisión de las mediciones y demostrando así que la tecnología es accesible y puede ofrecer resultados cercanos a los de una estación profesional (Fahim et al., 2023).

Según Jasso-Reyes et al. (2025), menciona en su investigación que una estación meteorológica IoT de bajo presupuesto diseñada para recopilar continuamente las variables climatológicas esenciales mediante los sensores integrados. Este sistema utiliza un microcontrolador ESP8266 como una unidad central, conectando a los sensores de la temperatura, humedad, presiones atmosféricas (BME280) y radiación UV (GUVA-S12SD). Dichos sensores permiten adquirir datos primarios y a partir de ellos, calcular indicadores derivados como punto de rocío, sensación térmica, la densidad del aire y la altitud. Esta estación transmite la información en tiempo real a través de una red Wi-Fi, además de su almacenamiento local en una tarjeta micro-SD para su análisis posterior. Durante 29 días de funcionamiento el prototipo generó registro cada diez segundos. Estos datos obtenidos fueron comparados con estaciones meteorológicas comerciales y mostraron coherencia en la mayoría de las variables, confirmando así que los sensores de bajo costo pueden ofrecer mediciones confiables cuando integran adecuadamente en un sistema IoT

2.3.3.3 Procesamiento y análisis de datos meteorológicos

El procesamiento de datos ambientales incluye técnicas como reclasificación, esto permite filtrar, reorganizar información para facilitar la interpretación de sus datos. Una de las técnicas más utilizadas es la técnica de reclasificación, usada para seleccionar rangos específicos de valores y así identificar los patrones tales como: vegetación, cuerpos de agua o zonas relevantes dentro de una imagen ambiental. Cuando se requiere un análisis más exacto, se emplean

métodos de análisis multivariado, los cuales permiten manejar una gran cantidad de volúmenes de datos y así reducir la complejidad mediante las herramientas como PCA y SVD. Es por eso por lo que el aprendizaje automático contribuye a clasificar los datos meteorológicos, ya sea mediante los métodos supervisados o no supervisados, como el análisis de clústeres. Esta técnica se implementa comúnmente en software GIS o los lenguajes de programación especializados, permitiendo un estudio eficiente y preciso de la información ambiental (Acevedo, 2024, p. 198).

Los sistemas meteorológicos basados en IoT, el procesamiento de los datos comienza con la recolección continua de variables como temperatura, humedad, presión y la velocidad del viento mediante las redes de sensores inalámbricas. Dichos datos son enviados a una unidad central, donde se ejecutan y se procesan automáticamente permitiendo su depuración, estructurando así el análisis. Los modelos de Machine Learning identificando patrones temporales y relaciones complejas entre las variables, generando predicciones climáticas en un corto plazo con una mayor efectividad. Esta combinación de datos históricos y las mediciones en tiempo real fortalecen la capacidad de análisis. Dicho enfoque integrado permite detectar las anomalías, emitida por alertas oportunas para así mejorar la toma de decisiones en sectores sensibles al clima. Estas plataformas web y aplicaciones móviles simplifican la visualización de la información que ha sido procesada, y así fortaleciendo la respuesta ante los eventos meteorológicos críticos (Singh et al., 2024).

2.3.3.4 Calibración y mantenimiento de sensores meteorológicos

Según Krüger et al. (2024), la calibración de los sensores meteorológicos tales como el pluviómetro que es de bajo costo es utilizado en infraestructuras IoT para mejorar las redes de monitoreo ambiental. Se evaluaron más de 50 sensores mediante pruebas de laboratorio y variaciones significativas entre las unidades, iniciando así la calibración de fábrica no es suficiente. El estudio presentó dos plataformas IoT llamadas: Raspberry Pi y Arduino, para la transición de los datos meteorológicos. Esto permite la conectividad vía Wi-Fi, GSM y Sigfox según el requerimiento energético. Esto destaca en la importancia del mantenimiento y de los ajustes continuos de los sensores. Cabe recalcar que los sensores económicos pueden ser útiles, pero necesitan una calibración para garantizar que los datos sean confiables.

De acuerdo con Nan et al. (2025) los sensores de temperatura de bajo costo suelen presentar errores debido a la radiación solar, la humedad y otras variaciones ambientales externas. Para ratificar las desviaciones, este estudio plantea un modelo de calibración respaldada en redes neuronales artificiales, que se ajustan automáticamente a las mediciones según las condiciones reales del entorno. Este autor compara el método con las técnicas tradicionales, demostrando así el enfoque neuronal que se encargade reducir significativamente la incertidumbre. Debido a ello, se evalúa la estabilidad del sensor a lo largo del tiempo, descubriendo que el mantenimiento periódico, la recalibración son fundamentales para garantizar así la precisión y prevenir futuros errores.

2.3.3.5 Almacenamiento y gestión de datos meteorológicos

Conforme a lo planteado por Hachimi et al. (2022), la gestión de los datos meteorológicos requieren incorporar fuentes como sensores IoT, las estaciones automáticas y los conjuntos de reanálisis atmosféricos, que permite obtener registros continuos y de alta calidad. Una arquitectura moderna que combina capas de adquisición, almacenamiento, procesamiento y aplicación, donde las bases NoSQL como la de MongoDB facilitan el manejo de grandes cantidades de información heterogénea. Mediante las técnicas de inteligencia artificial y aprendizaje automático es viable atribuir datos faltantes, corrigiendo así las inconsistencias generar pronósticos precisos. Los modelos como las redes neuronales y XGBoost han demostrado un alto rendimiento al estimar variables críticas, como la temperatura y la evaporación. Dicha herramienta mejora la toma de decisiones agrícolas, especialmente en zonas semiáridas, al momento de transformar los datos dispersos en indicadores fiables para riego el manejo climatológico.

2.3.3.6 Conectividad y transmisión de datos meteorológicos

Los sistemas meteorológicos basado en IoT, la transmisión eficiente de los datos depende de una infraestructura inalámbrica que sea capaz de enlazar los sensores distribuidos con un nodo central de control. En la propuesta analiza, como los nodos recolectan las variables ambientales y estas son enviadas mediante enlaces inalámbricos hacia un controlador coordinado, que organiza el flujo de información dentro del invernadero. Una pasarela construida en Raspberry Pi actúa como un punto de conectividad, que integra la red interna y proporcionar la conexión hacia la nube para el almacenamiento y supervisión remota de los datos. Esta arquitectura

garantiza prolongación en la comunicación entre sensores, controladores y los actuadores, que facilitan un monitoreo constante y decisiones automatizadas que se basan en la información en tiempo real (Albuali et al., 2023).

Según Ariffudin y Musa (2022), la comunicación de los datos meteorológicos es importante para el funcionamiento oportuno de las estaciones automáticas. El estudio realizado compara el protocolo tradicional FTP con MQTT, evidenciando que FTP provoca retrasos significativos, con un mayor consumo de ancho de banda y frecuentes pérdidas de la información. MQTT, está diseñada para entornos IoT, que utiliza paquetes livianos y un sistema publish/subscribe que reduce el tráfico y mejora la estabilidad de este. Esto muestra una transmisión con más rapidez, y confiable energéticamente eficiente con MQTT. No obstante, disminuye la carga del módulo GSM y mejora la disposición de los datos en tiempo real.

2.3.3.7 Teledetección y monitoreo remoto del ambiente

La teledetección satelital, permite monitorear las variables ambientales de manera continua en gran escala, mejorando la comprensión de los procesos climáticos y de la calidad del aire. Explicando que los sensores remotos proporcionan datos espaciales homogéneos que complementan las mediciones terrestres, especialmente donde no existe ningún tipo de estación meteorológica. Es por ello que la teledetección se presenta como una herramienta crucial para el monitoreo ambiental y de la comprensión de los procesos atmosféricos a gran escala (Muskatel et al., 2021).

Según Tyagi y Tiwari (2025), el Internet de las Cosas (IoT) con una arquitectura de los sensores interconectados capaz de recopilar, transmitir y de procesar datos agrícolas sin la necesidad de la intervención humana, facilitando así la automatización, la interoperabilidad y el monitoreo continuo. Dichas aplicaciones incluyen; agricultura de precisión, monitoreo de ganado, automatización de invernaderos, gestión inteligente del riego y por último están las estaciones meteorológicas IoT que permite a los usuarios recolectar y registrar las variables climatológicas en tiempo real. Esto favorece las decisiones basadas en los datos, optimizando recursos y mejoras en el rendimiento agrícola.

2.3.3.8 Gemelos digitales y automatización en monitoreo meteorológico

Según Tyagi y Tiwari (2025), los gemelos digitales se basan en la integración de datos en tiempo real y modelos virtuales capaces de replicar los comportamientos de sistemas físicos, lo que permite un monitoreo constante y automatizado del entorno. A través de los sensores, drones, satélites y las estaciones meteorológicas, estos sistemas capturan parámetros ambientales que permiten evaluar las condiciones climáticas y anticipar variaciones. La información recolectada pasa por los procesos de depuración, fusión, y de transformación, garantizando así su utilidad en los modelos digitales. La réplica virtual facilita las simulaciones climáticas, las predicciones y el análisis de dichos escenarios posibles, vigorizando así la toma de decisiones. Dentro de este panorama, la estación meteorológica constituye el punto de partida de todo el proceso: es el componente que captura de manera automatizada y continua los parámetros ambientales del entorno, generando los datos que posteriormente pueden alimentar modelos más avanzados. Por ello, el presente trabajo se centra en el diseño de un nodo de medición confiable como base indispensable de cualquier sistema de monitoreo automatizado.

2.3.4 Metodología de desarrollo ágil (Scrum)

Scrum es una metodología ágil que se enfoca en la colaboración y la entrega constante de valor al cliente. Esto permite que los proyectos se dividan en integraciones pequeñas llamadas “Sprints”, que persisten de una a cuatro semanas. Cada sprint o iteración empieza con una reunión de planificación en la que se eligen tareas y se establecen objetivos. Durante el sprint, el equipo trabaja en las tareas asignadas y se reúnen diariamente para revisar avances. Esta metodología origina la adaptabilidad y la entrega incremental de valor, siendo perfecto para proyectos que requieren flexibilidad y respuesta rápidas a los cambios (Turley & Rad, 2019).

La metodología de desarrollo ágil Scrum se describe por varios elementos claves, entre ellos se encuentra el Product Owner, responsable de definir y priorizar los requisitos del proyecto para maximizar el cumplimiento del marco de trabajo y eliminar los inconvenientes que puedan afectar el rendimiento del equipo. El grupo de desarrollo está constituido por profesionales multidisciplinarios que se encarga de ejecutar las tareas necesarias para cumplir con los objetivos del Sprint. Dichos elementos se trabajan en conjunto de manera coordinada para

lograr la entrega incremental y constante de productos de alta calidad , permitiendo así una ágil adaptación a los cambios y necesidades del mercado (Estrada Velasco et al., 2021).

Por otro lado Tymkiw et al.(2020), destaca las partes más fundamentales de Scrum, entre ellas está la planificación del Sprint, donde se presenta el trabajo completado durante un periodo corto de tiempo, la reunión diaria, en la que el equipo actualiza el progreso y planifica el trabajo del día, la revisión el Sprint, donde se muestra el trabajo realizado al Product Owner para su respectiva evaluación y posibles ajustes. No obstante, el Sprint permite al equipo reflexionar sobre el desempeño, para así identificar áreas de mejora y establecer acciones concretas para optimizar continuamente el proceso de desarrollo en los siguientes períodos. Esta etapa, trabajadas de manera iterativa, garantizando así un flujo constante de valor y de adaptación a los cambios constantes del proyecto.

La metodología Scrum, según Alberto et al. (2020), se caracteriza por su enfoque colaborativo y su orientación hacia la entrega continua del producto al cliente. Es por esto, que el trabajo se divide los proyectos en Sprints, los ciclos cortos que duran entre una y cuatro semanas, cada uno empieza con una planificación detallada y quien define y prioriza los requisitos del proyecto. El Scrum Master es el encargado de facilitar el proceso y eliminar dificultades, el equipo de desarrollo que está conformado por personas profesionales multidisciplinarios que llevan a cabo las tareas necesarias. La constante iteración de estos elementos permite una entrega incremental y constante del software de alta calidad, permitiendo así adaptarse de manera ágil a los cambios y necesidades del clima.

2.4 Conclusiones del marco teórico

El análisis histórico y conceptual ha demostrado que los sistemas embebidos IoT han surgidos evoluciones desde dispositivos de control simple hacia las plataformas inteligentes que son capaces de procesar los datos, comunicarse inalámbricamente y de operar de manera autónoma. Esta transformación tecnológica ha logrado que los sensores, los microcontroladores y los módulos de comunicación se conviertan en soluciones compactas, eficientes con una alta precisión, convirtiendo así a la tecnología IoT en un componente necesario, para los sistemas de monitoreo constante y automatización avanzada.

Por otra parte, el monitoreo meteorológico ha crecido significativamente desde prácticas manuales y de baja frecuencia hacia sistemas automatizados que registran los parámetros climáticos en tiempo real. Esta evolución ha fortalecido la capacidad de anticipar los fenómenos atmosféricos, para así optimizar recursos agrícolas y así reducir los riesgos climáticos, evidenciando la necesidad de tecnologías capaces de proveer datos confiables, oportunos y procesables.

La combinación del sistema embebido IoT con el monitoreo meteorológico demuestra ser una combinación robusta y coherente. Mientras que IoT contribuye una infraestructura tecnológica para la recolección y transmisión de datos. El monitoreo meteorológico se encarga de limitar los parámetros requeridos y orientando así la interpretación de la información climática. Ambas variables construyen una solución que es capaz de modernizar la observación ambiental, ofreciendo así un sistema automatizado y muy eficiente, a su vez estable para la toma de decisiones de manera informada.

Con relación con la metodología Scrum, el desarrollo del sistema se realiza de manera iterativa e incremental mediante la ejecución de los Sprints, permitiendo así la entrega constante de las funcionalidades y la integración de la retroalimentación constante, durante todo el proceso. Este enfoque resulta especialmente adecuado para el sistema embebido que requiere ajustes periódicamente debido a los cambios del entorno que puedan afectar la precisión de los sensores. A través de las revisiones y de las validaciones continuas al final de cada sprint, se garantiza la mejora progresiva del prototipo, asegurando que vaya evolucionando hasta alcanzar un funcionamiento óptimo, estable y adaptado a las necesidades del contexto actual.

CAPÍTULO III

3 MARCO INVESTIGATIVO

3.1 Introducción

En este presente capítulo se expone el marco investigativo que sirvió de fundamento, para el desarrollo del sistema embebido con arquitectura IoT, destinado al monitoreo meteorológico del vivero Lastenia Vera. En este apartado se describen los enfoques metodológicos empleados para la ejecución de la investigación, los cuales permitieron, analizar e interpretar la información necesaria para identificar la problemática existente y establecer los requerimientos, que orientaron el diseño de la propuesta tecnológica. Asimismo, se presentan las fuentes de información utilizadas y las técnicas de la recolección de datos aplicadas durante el estudio. De igual manera, se describen la población y la muestra, consideradas en la investigación, así como las estrategias utilizadas para la aplicación de la encuesta y la entrevista. Finalmente, se presentan los resultados obtenidos y su análisis, los cuales permitieron identificar las necesidades que fundamentaron el desarrollo del sistema embebido propuesto.

3.2 Tipos de investigación

3.2.1 Investigación de Campo

Es un método cualitativo que recopila datos que implica la observación e interacción con las personas y sus entornos naturales. Es por eso que, en lugar de controlar las variables en un laboratorio o depender únicamente de datos digitales, permitiendo así que la investigación sitúe al investigador en el contexto en que estudia, lo cual capta el comportamiento, matices y las dinámicas sociales tal como sucede en la vida real (Rebollo & Ábalos, 2022).

Esta investigación de campo se utilizó durante la instalación y las pruebas de la estación meteorológica en el ambiente real, específicamente en el vivero donde se desarrolló la investigación. Fue necesario medir las variables climáticas como la temperatura y la humedad, evaluando así el desempeño del sistema embebido durante la adquisición y el procesamiento de las mediciones. Este enfoque, permitió recolectar datos empíricos que sirvieron para evaluar si el prototipo respondió correctamente a las condiciones climatológicas.

3.2.2 Investigación bibliográfica

La investigación bibliográfica o también llamada revisión de literatura es el proceso sistemático de examinar estudios previos con el fin de evaluar conceptos, teóricos y resultados relacionados con el tema investigado. Dicho análisis bibliográfico permite identificar los vacíos, tendencias y aportes existentes, proporcionando las bases científicas para el desarrollo del estudio actual. No obstante, contribuye con los fundamentos críticos para la construcción del marco teórico y la validación del enfoque adoptado en la investigación (Donthu et al., 2021).

En el presente proyecto se aplicó la investigación bibliográfica, mediante una revisión sistemática de estudios previos, artículos científicos y documentos técnicos que permitieron fundamentar el marco conceptual y la metodología del prototipo IoT. Dicha estrategia permitió analizar metodología y los resultados existentes para identificar y sustentar decisiones tecnológicas orientadas al desarrollo del sistema.

3.2.3 Investigación descriptiva

La investigación descriptiva se centra en observar, describir y documentar los aspectos de una situación o fenómeno que no ha sido estudiado, ya sea natural o social, para entender sus características, relaciones, cambios y tendencias. No involucra intervención ni el control experimental, y el objetivo principal es responder preguntas como ¿qué?, ¿quién?, ¿cuándo? y ¿dónde? sobre el fenómeno estudiado (Alban et al., 2020).

Dicha investigación se aplicó al describir las características del sistema embebido, funcionamientos de sensores, flujo de datos, y condiciones ambientales del vivero. No obstante, fue útil para determinar el comportamiento del sistema en escenarios reales. A partir de estos datos se podrán delimitar los parámetros, tendencias y patrones sin necesidad de manipular dichas variables.

3.2.4 Métodos de investigación

3.2.4.1 Método experimental

El método experimental es la manipulación controlada de variables para observar los efectos y los resultados, evaluando así el comportamiento de un sistema bajo las condiciones reales o ya

sea simuladas. Dicho método permite comprobar hipótesis, medir el desempeño del prototipo y validar funciones mediante pruebas directas, centrándose así en la validez del estudio realizado previamente (Vera et al., 2020, p. 109).

Este método se aplicó cuando se realizaron pruebas del sistema embebido IoT instalado en el vivero o zona de estudio, evaluando así la precisión de sensores ambientales, el procesamiento de las mediciones y el funcionamiento del sistema en tiempo real. Este método permitió comparar los resultados esperados con resultados obtenidos, validando así su correcto funcionamiento del sistema en tiempo real.

3.2.4.2 Método analítico

Este método analítico consiste en descomponer un fenómeno en partes más pequeñas sus componentes esenciales para estudiar su estructura, relaciones y el funcionamiento, con el único propósito de comprender de manera integral cada una de sus partes. Para posteriormente repetir el mismo proceso, de forma conjunta y así poder asimilar de manera holística e integral (Blácido et al., 2022).

Este método se utilizó al momento de los análisis de los datos meteorológicos obtenidos en la investigación, permitiendo identificar tendencias, patrones y fluctuaciones en las variables monitoreadas. Asimismo, permitió comparar los valores almacenados con registros históricos, facilitando el análisis de comportamiento ambiental y la interpretación técnica de los resultados ambientales obtenidos del sistema IoT.

3.3 Fuentes de información de datos

3.3.1 Fuentes primarias

Las fuentes primarias corresponden a aquellas de las cuales se obtiene información de primera mano, directamente relacionada con las situaciones o fenómenos estudiados. Este tipo de fuente permite recopilar datos a partir de la observación y experiencia directa en el contexto donde se desarrolla la investigación, proporcionando información específica y actualizada para el análisis de la problemática (Amaya & Vladimir, 2020).

En la presente investigación, las fuentes primarias estuvieron constituidas por la información recopilada directamente en el vivero Lastenia Vera, a través de dos instrumentos: la encuesta aplicada a los estudiantes que realizan prácticas en el lugar y la entrevista realizada al ingeniero responsable del vivero. Estas fuentes permitieron obtener datos de primera mano sobre las condiciones actuales del monitoreo climático, las dificultades presentes en el manejo del cultivo de cacao y las necesidades reales que fundamentaron el desarrollo de la propuesta tecnológica.

3.3.1.1 Encuestas

La encuesta es un método de la investigación que se utiliza para obtener información ya sea de gran número de personas o ya sea de un grupo pequeño. Además de ello es una herramienta versátil y accesible que permite a los investigadores recolectar información sobre los comportamientos, actitudes u opiniones y demografía de una población en específico. Las encuestas pueden realizarse de diferentes maneras que son: encuesta de investigación cuantitativa, encuesta cualitativa, encuesta por internet, encuesta por correo, encuesta por teléfono, encuesta en personas, encuestas mixtas (Medina et al., 2023).

En este proyecto se aplicó una encuesta a los estudiantes de la carrera de Agropecuaria de la ULEAM, Extensión El Carmen, que realizan sus prácticas en el vivero Lastenia Vera, ya que este grupo posee un conocimiento directo sobre las condiciones ambientales del cultivo y de las dificultades actuales de monitoreo climático. Esta encuesta abordó temas generales como las técnicas de medición utilizadas actualmente, los problemas que enfrentan con la recolección manual de los datos, la importancia del monitoreo ambiental y la aceptación de un sistema automatizado mediante la tecnología IoT.

3.3.1.2 Entrevista

La entrevista es un método empírico, basado en la comunicación interpersonal establecida entre el investigador y el sujeto o los sujetos de estudio, para adquirir respuestas verbales a las interrogantes planteadas sobre el problema. Existen diferentes tipos de entrevistas, entre ellas la entrevista estructurada, que se caracteriza por formular preguntas previamente definidas y estandarizadas, y la entrevista no estructurada o de exploración, que permite una conversación más flexible y adaptada al desarrollo de la interpretación (Feria Ávila et al., 2020).

En la presente investigación, la entrevista se aplicó al Ing. Pedro Nivelá, responsable del vivero Lastenia Vera, quien aportó información técnica sobre los métodos de medición utilizados, las dificultades que presentaba durante el control de las variables climatológicas, las necesidades de automatización y la utilidad esperada del sistema IoT propuesto.

3.4 Estrategia operacional para la recolección de datos

3.4.1 Población

La población, según Casteel y Bridier (2021), es el conjunto de individuos, grupos, organizaciones u otras entidades que son de objeto de interés en la investigación y hacia quienes pueden generalizarse los resultados adquiridos. Esta población delimita el alcance del estudio y establece el contexto que permite identificar las subpoblaciones específicas. Además de ello, indica las características o límites que establecen quienes formarán parte o quedan fuera del estudio a realizar, facilitando así a los lectores comprendan con precisión el grupo a analizar.

En la presente investigación, la población está conformada por 90 personas: 89 estudiantes de la carrera de Agropecuaria de la Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen, que realizan prácticas en el vivero Lastenia Vera, y 1 docente responsable del vivero. Los estudiantes participan en actividades relacionadas con el mantenimiento y seguimiento del cultivo de cacao, por lo que constituyen la población a la que se aplicó la encuesta, mientras que al docente se le aplicó la entrevista como informante clave.

3.4.2 Muestra censal

La muestra corresponde al grupo de individuos del cual se recopila la información para el análisis. En estudios donde la población es reducida, es posible trabajar con la totalidad de sus integrantes en lugar de seleccionar un subconjunto (Martínez, 2020, pp. 16-17). Según Hurtado de Barrera (2000), en las poblaciones pequeñas o finitas no se selecciona una muestra, sino que se trabaja con todos sus integrantes para no afectar la validez de los resultados. Este procedimiento, conocido como muestra censal o censo, se aplica cuando la población es finita y accesible en su totalidad, lo que permite evitar el error de muestreo y garantizar que los resultados representen al grupo completo.

Debido a que la población de estudiantes es reducida y de fácil acceso, no se aplicó un cálculo de muestra ni una selección probabilística. En su lugar, se trabajó con la totalidad de la población, aplicando la encuesta a los 89 estudiantes que realizan prácticas en el vivero. Este tipo de estudio, conocido como censo, permite recopilar información del total de los individuos involucrados, garantizando la máxima representatividad de los resultados y evitando el error de muestreo.

3.4.3 Análisis de las herramientas de recolección de datos a utilizar

3.4.3.1 Encuesta

Se utilizó la encuesta como técnica de recolección de información dirigida a los estudiantes de la carrera de Agropecuaria de la ULEAM, Extensión El Carmen, que realizan sus prácticas en el Vivero Lastenia Vera. Su aplicación permitió recopilar información relacionada con las prácticas actuales de monitoreo ambiental, las dificultades en la recolección de datos y la percepción sobre la implementación de un sistema automatizado basado en IoT.

3.4.3.2 Entrevista

Se utilizó la entrevista como técnica de recolección de información dirigida al ingeniero Ing. Pedro Nivelá responsable del Vivero Lastenia. Su aplicación permitió obtener información cualitativa relacionada con la gestión del vivero, las condiciones del cultivo de cacao, los métodos utilizados para el seguimiento de las variables climatológicas y las principales dificultades identificadas en el manejo del cultivo.

3.4.3.3 Estructura de los instrumentos de recolección de datos aplicados

3.4.3.3.1 Estructura de encuesta

El instrumento utilizado para la encuesta estuvo conformado por 30 preguntas cerradas. Las preguntas estuvieron orientadas a recopilar información relacionada con las prácticas actuales de monitoreo ambiental, las dificultades asociadas con la recolección de datos y la percepción sobre la implementación de un sistema automatizado basado en IoT. La encuesta fue aplicada en línea durante un período de dos meses mediante la plataforma Microsoft Forms, obteniéndose 89 respuestas válidas, las cuales fueron consideradas para el análisis de los

resultados. Cabe destacar que, debido a la extensión de instrumento, para el análisis de resultados expuestos en este capítulo se seleccionaron únicamente las 6 preguntas más relevantes y significativas que justifican de manera directa la problemática de la investigación. La estructura de la encuesta aplicada se puede revisar en el ANEXO C.

3.4.3.3.2 Estructura de entrevista

El instrumento utilizado para la entrevista estuvo conformado por una guía de 13 preguntas abiertas, dirigida al Ing. Pedro Nivelá, responsable del Vivero Lastenia Vera. Las preguntas abordaron aspectos relacionados con la gestión del vivero, las condiciones del cultivo de cacao, los métodos utilizados para el seguimiento de las variables climatológicas y las principales dificultades identificadas en el manejo del cultivo. Al igual que con la encuesta, en la sección de resultados de este capítulo solo se expone el análisis de las 6 preguntas más determinantes para sustentar los objetivos del proyecto. La estructura de la entrevista aplicada se presenta en el ANEXO D.

3.4.4 Plan de recolección de datos

Tabla 1: Plan de recolección de datos

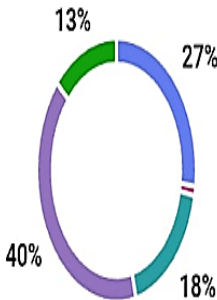
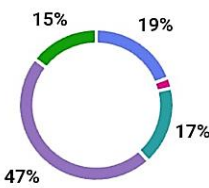
ACTIVIDAD	RESPONSABLES	FECHA
Ejecución de encuestas	Anderson Plua, Castro Dereck, López Bryan, Zambrano Anthony, Guerrero David, Rodríguez Kevin.	29/01/2026, 21/04/2026
Realización de entrevista	Anderson Plua, Castro Dereck, López Bryan, Zambrano Anthony, Guerrero David, Rodríguez Kevin.	15/04/2026, 11:30

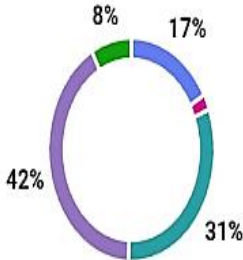
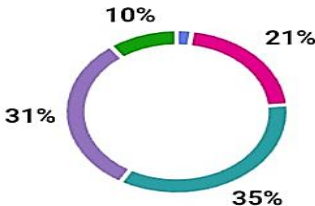
3.5 Análisis y presentación de resultados

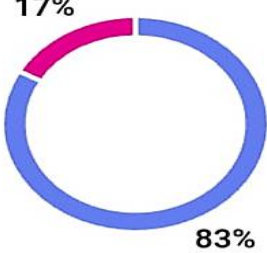
3.5.1 Presentación y descripción de los resultados obtenidos

3.5.1.1 Análisis de encuesta realizada a Estudiantes que realizan prácticas en el vivero

Tabla 2: Resultados Obtenidos en la Encuesta

PREGUNTA	GRÁFICO	INTERPRETACIÓN
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?	Pregunta 1 - Totalmente en desacuerdo 24 - En desacuerdo 1 - Ni de acuerdo ni en desacuerdo 16 - De acuerdo 36 - Totalmente de acuerdo 12 	A pesar de que una parte de los encuestados considera que el proceso actual permite obtener la información, la mayoría evidencia que no se cuenta con los datos suficientes que sean claros y oportunos sobre el estado ambiental de las plantas. Dicha situación puede entorpecer la toma de decisiones adecuadas en el manejo del vivero, provocando problemas en el riego que causa el estrés hídrico o afectaciones en el desarrollo de los cultivos.
2. ¿Está usted de acuerdo que el personal que realiza actividades en el vivero cuenta conocimiento necesario para realizar el monitoreo de las	Pregunta 2 - Totalmente en desacuerdo 17 - En desacuerdo 2 - Ni de acuerdo ni en desacuerdo 15 - De acuerdo 42 - Totalmente de acuerdo 13 	A pesar de que los encuestados consideren que el personal posee conocimientos para la realización del monitoreo, se evidencia que no todos cuentan con la preparación esencial. Debido a ello surgen inconsistencias en la recolección e interpretación de los datos ambientales,

PREGUNTA	GRÁFICO	INTERPRETACIÓN
variables climáticas?		afectando así la precisión del monitoreo limitando así las implementaciones correctivas oportunas en el vivero.
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 ?	Pregunta 3 - Totalmente en desacuerdo 15 - En desacuerdo 2 - Ni de acuerdo ni en desacuerdo 28 - De acuerdo 37 - Totalmente de acuerdo 7 	Estos resultados evidencian que el estrés hídrico en las plantas de cacao tiene relación en gran medida, a la falta de información necesaria sobre las condiciones climáticas. Esta carencia limita la capacidad de tomar decisiones adecuadas correspondientes al riego, lo que provoca tanto el exceso como la escasez del agua, afectando directamente a la salud productiva del cultivo.
4. ¿Con que facilidad puede identificar las enfermedades del cultivo de cacao mediante la observación directa?	Pregunta 4 - Nunca 2 - Rara vez 19 - A veces 31 - Frecuentemente 28 - Siempre 9 	Se evidencia que la identificación de las enfermedades en el cultivo de cacao mediante la actividad de observación directa no siempre es la adecuada, lo que evidencia dificultades en la detección de problemas fitosanitarios. Dicha situación suele ocasionar que las enfermedades avancen sin control generando pérdidas en la producción que afectan a la

PREGUNTA	GRÁFICO	INTERPRETACIÓN																		
		calidad del cultivo, resaltando así la implementación de una herramienta tecnológica.																		
5. ¿Conoce usted que enfermedades pueden propagarse en el cultivo de cacao debido a los cambios climático ?	Pregunta 5 ● Sí 74 ● No 15  <table border="1"> <caption>Data for Pregunta 5</caption> <thead> <tr> <th>Respuesta</th> <th>Cantidad</th> <th>Porcentaje</th> </tr> </thead> <tbody> <tr> <td>Sí</td> <td>74</td> <td>83%</td> </tr> <tr> <td>No</td> <td>15</td> <td>17%</td> </tr> </tbody> </table>	Respuesta	Cantidad	Porcentaje	Sí	74	83%	No	15	17%	Los resultados obtenidos reflejan la mayoría de los encuestados no tienen conocimientos sobre la propagación de las enfermedades en dicho cultivo de cacao a causa de los constantes cambios climatológicos. Dicha falta de información puede impedir la prevención oportuna de las plagas y de las enfermedades, incrementando así el riesgo de las afectaciones en la producción lo que evidencia es la necesidad de contar con un sistema que proporcione datos claros sobre las condiciones del clima.									
Respuesta	Cantidad	Porcentaje																		
Sí	74	83%																		
No	15	17%																		
6.¿Con que frecuencia ha aplicado algún producto insecticida o abono sin conocer si la planta realmente lo necesita?	Pregunta 6 ● Nunca 23 ● Rara vez 23 ● A veces 29 ● Frecuentemente 10 ● Siempre 4  <table border="1"> <caption>Data for Pregunta 6</caption> <thead> <tr> <th>Respuesta</th> <th>Cantidad</th> <th>Porcentaje</th> </tr> </thead> <tbody> <tr> <td>Nunca</td> <td>23</td> <td>26%</td> </tr> <tr> <td>Rara vez</td> <td>23</td> <td>26%</td> </tr> <tr> <td>A veces</td> <td>29</td> <td>33%</td> </tr> <tr> <td>Frecuentemente</td> <td>10</td> <td>11%</td> </tr> <tr> <td>Siempre</td> <td>4</td> <td>4%</td> </tr> </tbody> </table>	Respuesta	Cantidad	Porcentaje	Nunca	23	26%	Rara vez	23	26%	A veces	29	33%	Frecuentemente	10	11%	Siempre	4	4%	En esta pregunta se observa que, en algunas ocasiones, se colocan productos como los insecticidas o los abonos sin conocer de que planta realmente requiere dicho producto. Esto puede generar un uso inadecuado de los insumos agrícolas, afectando así la salud del cultivo y del
Respuesta	Cantidad	Porcentaje																		
Nunca	23	26%																		
Rara vez	23	26%																		
A veces	29	33%																		
Frecuentemente	10	11%																		
Siempre	4	4%																		

PREGUNTA	GRÁFICO	INTERPRETACIÓN
		medio ambiente, además incrementan los costos innecesarios, lo que resalta la importancia de contar con información precisa para la toma de decisiones.

3.5.1.2 Análisis de entrevista realizada a Ing. Pedro Nivelá

Tabla 3: Resultados Obtenidos Entrevista.

PREGUNTA	RESPUESTA	INTERPRETACIÓN
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?	“... Por lo general el cacao es tropical y subtropical, entonces por lo menos debemos tener un rango entre 28 a 30 grados Celsius, con un porcentaje de humedad relativa no mayor a 75% para evitar problemas de enfermedades principalmente...”	El Ing. encargado menciona que, al existir un rango de temperatura, de humedad óptimo hoy en día existen variaciones climáticas en la zona muy extremas que afectan al cultivo del cacao evidenciado así que existe una necesidad muy importante en la supervisión de esta.
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?	“ A pesar de que la planta de cacao tiene muchos mecanismos para poder regular los procesos metabólicos, pero ya en un estado avanzado de falta de agua, si se produce un estrés hídrico...”	En esta pregunta se manifiesta que la falta de agua si causa problemas de estrés hídrico por los que la planta eleva su sistema metabólico a un nivel extraordinario tratando de evitar la fatiga producida por la falta de agua.

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?	“...tenemos muchas fluctuaciones en el clima en esta zona, por la nubosidad y tantos factores más, vientos la heliofanía que es otro factor importante en las plantas, entonces estas fluctuaciones producen estrés en las plantitas de cacao...”	Basado en la respuesta se pudo detectar que la falta de un monitoreo ambiental a través de un sistema automatizado afecta principalmente en el correcto crecimiento.
4. ¿Qué indicadores o rasgos físicos en las plántulas le permite determinar que la gestión de datos ambientales es exitosa o deficiente?	"...Principalmente el color, el brillo de las plantas es un indicador que nos permite establecer el estado de agua, salud y estado nutritivo... para que la planta se nutra tiene que haber capacidad de campo, oxígeno y agua. Si usted tiene esas condiciones, las plantas serán vigorosas..."	Esto valida la necesidad del sistema IoT propuesto, ya que un monitoreo preciso permite mantener estas condiciones óptimas que el ojo humano detecta a través del brillo de la planta.
5. ¿Cuáles son los principales problemas que enfrenta el Vivero Lastenia relacionados con la gestión del riego y demás variables ambientales en el cacao?	“...problema más grande del riego por la disponibilidad de agua... nosotros tenemos que improvisar pozo...artesanal tenemos que estar bombeando a dos cisternas...entonces eso se ha convertido en un problema, los costos principalmente el combustible ...por la subida entonces eso nos afecta”	En vista a la respuesta se pudo detectar que para llevar a cabo el proceso de riego existen varios factores como el alto precio del combustible, y el desperdicio del agua que se produce al no saber cuándo es necesario regar.

PREGUNTA	RESPUESTA	INTERPRETACIÓN
6. ¿Cuáles son los parámetros técnicos (humedad, temperatura, radiación) que usted considera vitales para el crecimiento óptimo del cacao en esta etapa?	“...necesita capacidad de campo, balance de oxígeno y agua en el suelo, temperatura adecuada en ambiente controlado...”	La información obtenida del entrevistado da a conocer que para que una planta de cacao crezca en óptimas condiciones es necesario tener un buen balance entre agua y temperatura, pero no siempre se logra esto porque las condiciones actuales han sido un problema en el vivero.

3.5.2 Informe final del análisis de los datos

Tomando como base el análisis de la encuesta aplicada a los estudiantes que realizan prácticas en el vivero y de la entrevista realizada al Ingeniero Pedro Nivelá, se logró identificar que el proceso actual de monitoreo de las condiciones ambientales presenta limitaciones que afectan la gestión adecuada del cultivo del cacao. Los resultados obtenidos mediante ambos instrumentos evidencian que, la información climática disponible no siempre es suficiente, oportuna ni precisa para apoyar la toma de decisiones relacionadas con el manejo del vivero.

Los resultados de la encuesta, específicamente en las preguntas 1 y 2 muestran que la mayoría de los participantes considera que, el monitoreo climático actual y conocimiento técnico disponible no garantiza un apropiado control de las variables climatológicas. Esta percepción coincide con lo manifestado por el entrevistado en las preguntas 1 y 6, quien indicó que el desarrollo optimo de las plántulas de cacao, depende del mantenimiento de las condiciones adecuadas de temperatura, humedad y de equilibrio hídrico, parámetros que requieren un monitoreo constante, para así evitar alteraciones que afecten el crecimiento del cultivo.

Por otro lado, las respuestas obtenidas en las preguntas 3 y 6, de la encuesta evidencian que la falta de información climática oportuna incide en la aplicación inadecuada de riego,

fertilizantes e insecticidas, causando situaciones de estrés hídrico y un uso ineficaz de los recursos agrícolas. Dichos resultados guardan relación con las respuestas del entrevistado en las preguntas 2 y 5, donde se señala que la disponibilidad del agua, las constantes variaciones del clima y los elevados costos asociados al proceso de riego, que representa una de las principales dificultades para el manejo del vivero, incrementando así el riesgo de afectar el desarrollo de las plantas.

De igual forma, la información obtenida en las preguntas 4 y 5 de la encuesta, demuestra que los estudiantes presentan dificultades para identificar oportunamente las enfermedades asociadas a los cambios del clima, además de que desconocen los efectos que las variaciones ambientales producen sobre el cultivo. Dicha situación es consistente con lo expresado por el entrevistado en las preguntas 3 y 4, quien manifestó que las fluctuaciones climáticas generan estrés en las plántulas, y que indicadores visuales como el color y el brillo de las hojas reflejan el estado hídrico y nutritivo de la planta. Esto evidencia la necesidad de disponer de información climatológica en tiempo real que complemente la observación visual.

Los resultados obtenidos mediante la encuesta y la entrevista permiten concluir que el vivero Lastenia Vera presenta limitaciones en el monitoreo de las variables climatológicas, debido a que este se realiza de forma manual y sin información continua en tiempo real. La dependencia de observaciones manuales y de la falta de información ambiental continua, dificultan la detección temprana de condiciones desfavorables, incrementando el riesgo de estrés hídrico, favoreciendo la aparición de las enfermedades y reduciendo la eficiencia en la administración de los recursos utilizados en el cultivo.

Por lo tanto, se justifica el desarrollo de un sistema embebido con arquitectura IoT para el monitoreo de estación meteorológica en el Vivero Lastenia ULEAM EL Carmen. El cual permitirá realizar la adquisición automática de las variables de temperatura y humedad, su procesamiento en tiempo real y la generación de información ambiental confiable para apoyar la toma de decisiones relacionadas con el manejo del cultivo de cacao. La implementación de esta propuesta contribuirá a la optimización y el control de las condiciones del clima, mejorando así el uso de los recursos disponibles y fortaleciendo las actividades de monitoreo dentro del vivero.

CAPÍTULO IV

4 MARCO PROPOSITIVO

4.1 Introducción

En el presente capítulo describe el proceso de diseño e implementación del sistema embebido, para el monitoreo meteorológico automatizado del vivero Lastenia Vera, desarrollado sobre el microcontrolador ESP32 con el framework Arduino en el entorno PlatformIO. Se detalla la arquitectura modular adoptada, las decisiones técnicas tomadas y estructuración de los datos climáticos.

El desarrollo se organiza mediante la metodología Scrum, dividiendo el trabajo en cinco Sprints con objetivos específicos. Cada Sprints entrego una funcionalidad operativa que fue validada antes de avanzar a la siguiente etapa, lo que permitió detectar y corregir los errores de manera temprana si afectar los módulos ya completados.

El sistema resultante realiza mediciones constantemente de la temperatura y de la humedad cada 15 minutos, aplica filtrado estadístico para así realizar un filtrado estadístico para eliminar lecturas anómalas, valida los datos dentro del rangos físicos aceptables, detecta anomalías climáticas y genera un payload JSON estructurado listo para ser entregado al módulo de seguridad, que es responsabilidad de otro integrante del equipo.

4.2 Descripción de la propuesta

Esta propuesta consiste en la creación de un sistema embebido con una arquitectura IoT que está orientado al monitoreo automático de las variables climatológicas el en vivero Lastenia. Dicho sistema estará basado en el uso de microcontroladores ESP32, el cual permitirá la recolección, procesamiento y estructuración de los datos ambientales tales como la temperatura, humedad, presión atmosférica, entre otros. Este sistema está enfocado principalmente a agricultores, técnicos y responsables del vivero, quienes requieren una toma de decisiones en el manejo del cultivo.

La solución propuesta permite automatizar la recolección de los datos, reduciendo la investigación empírica y aumentando la precisión en la información obtenida. Una de sus

principales funcionalidades se encuentra en la medición continua de las variables del clima cada 15 minutos, el procesamiento de los datos recopilados mediante la técnica de filtrado y la validación, la detección de la anomalía y de la generación de datos en un formato JSON preparados para ser entregado al módulo de transmisión que será el encargado de la comunicación con sistemas externos .

4.3 Determinación de recursos

4.3.1 Humanos

Para realizar este proyecto, es de suma importancia disponer de varios recursos que faciliten el desarrollo de los objetivos planteados, cubriendo así el capital humano, contando con los recursos tecnológicos y los recursos económicos.

Tabla 4: Recursos humanos

Recurso	Cargo	Función
Anthony Javier Zambrano Alcivar	Desarrollador	Diseño del sistema embebido, programación en C++, con la integración de los sensores, implementación de algoritmos de procesamiento de datos.
Ing. Danilo Arévalo	Tutor	Supervisión del proyecto, revisión técnica, validación del cumplimiento de objetivos.
Personal del vivero Lastenia	Usuario final	Pruebas del sistema, validación de funcionamiento entorno real.

4.3.2 Tecnológicos

Es de suma importancia de contar con la tecnología tanto como es software y del hardware para realizar la propuesta, considerando ambos al momento de desarrollar la propuesta del sistema embebido, lo que asegura que los equipos empleados cumplan con ciertas especificaciones para así facilitar el trabajo del programador.

Tabla 5: Recursos tecnológicos

Recurso	Característica	Descripción
Laptop	Sistema Operativo: Microsoft Windows 10 Pro. RAM: 16 GB Procesador: Intel(R) Core(TM) i5-7440HQ CPU @ 2.80GHz, de 7ma Generación.	Utilizada para el desarrollo del sistema y programación en PlatformIO.
PlatformIO	Ultima versión estable, con extensión de VS Code.	Entorno de desarrollo para programación del ESP32.
C y C++ (Arduino)	Lenguaje de programación embebido	Desarrollo de firmware del sistema.
ESP32	Microcontrolador 240 MHz, WiFi/Bluetooth integrado.	Unidad principal donde se procesa el sistema.
Sensor DHT22	Sensor digital de temperatura y humedad	Permite medir condiciones
Panel solar 5-6 V 180 mA(107x62mm)	Módulo de alimentación de energía	Brinda energía para los circuitos electrónicos.
Baterías de Li-ion 18650 3.7 v 3400 mAh	Baterías recargables	Almacenar energía durante los días nublados
Módulo cargador TP4056	Cargador de baterías	Módulo de carga de baterías

4.3.3 Económicos

Los recursos económicos están reservados para el desarrollo de la propuesta y las pruebas que se desarrollarán, lo que se detalla a continuación en la siguiente tabla.

Tabla 6: Recursos económicos

Cantidad	Recurso	Precio unitario	Subtotal
1	Laptop	\$370.00	\$370.00
200	Horas de programación	\$5.00	\$1,000.00
4	Meses de internet	\$30.00	\$120.00
1	ESP32	\$10.00	\$10.00
1	Sensor DHT22	\$3.00	\$3.00
1	Cables	\$2.00	\$2.00
1	Panel solar	\$20.00	\$20.00
1	Módulo cargador	\$10.00	\$10.00
1	Baterías de Li-ion	\$2.00	\$2.00
1	Caja de protección IP65	\$7.00	\$7.00
TOTAL:			\$1,544.00

4.4 Desarrollo del software mediante metodología Scrum

La metodología Scrum es uno de los marcos de trabajo ágil que se encarga de organizar el desarrollo del software en ciclos cortos e interactivos llamados Sprints, cada uno con un objetivo en específico y con un entregable al finalizar. Dicha metodología permite validar cada uno de los componentes del sistema en etapas tempranas, facilitando así la detección de los errores que puedan surgir durante la incorporación de mejora continuas.

En el presente proyecto, se escogió esta metodología debido a que en un sistema embebido que requiere constantemente de pruebas con hardware real, sensores y las condiciones climáticas variables. El uso de Scrum permitió organizar el desarrollo en Sprints independientes, implementando inicialmente funcionalidades básicas como la lectura de los sensores, para posteriormente unirlos a procesos más avanzados tales como el filtrado de datos, validación, detección de anomalías y de la generación de los datos estructurados en un formato JSON, garantizando así que cada módulo funcione correctamente antes de avanzar con el siguiente.

4.4.1 Descripción del producto

4.4.1.1 Propósito del producto

Para desarrollar este sistema embebido con la arquitectura IoT que sea capaz de monitorear las variables meteorológicas de manera automatizada, para así procesar los datos en tiempo real y

generar información que sea confiable para poder tomar decisiones adecuadas basadas en los datos obtenidos, y así tener un mejor manejo en el cultivo del cacao aumentando así la producción de este.

4.4.1.2 Funcionalidades Claves

- Medición de la temperatura
- Medición de la humedad
- Estabilización automática de los sensores
- Filtración de datos mediante media recortada
- Validación de los datos por rangos
- Detección de anomalías
- Cálculo de tendencia (sube, baja, estable)
- Generación de datos en formato JSON

4.4.1.3 Usuario objetivo

Tabla 7: Usuario objetivo

Tipo de usuario	Principales funcionalidades
Estudiantes y agricultores	Consultar sobre condiciones climáticas para la toma de decisiones.
Técnico agrícola	Observación de los datos ambientales y monitoreo del sistema.
Investigador	Uso de datos para estudio s climáticos y agrícolas.

4.4.1.4 Criterios de éxito del producto

- Este sistema deberá realizar mediciones automáticas de temperatura y de humedad cada 15 minutos de forma continua.
- Los datos que se generen deben ser precisos y validados dentro del rango aceptable.
- El sistema deberá identificar anomalías en las variables climatológicas.
- Los datos deberán generarse correctamente en un formato JSON para la integración con otros sistemas.

4.4.2 Requerimientos del sistema

4.4.2.1 Historia de usuario

4.4.2.1.1 Historia de usuario 1: Medición automática de datos

Tabla 8: Historia de usuario 1: Medición automática

HISTORIA DE USUARIO			
Id de historia	HU01	Título:	Medición automática de datos
Prioridad:	Alta	Usuario:	Técnico agrícola
Riesgo de desarrollo:	Medio	Tamaño de la tarea:	Grande
Como:	Técnico agrícola		
Quiero:	Que el sistema mida automáticamente las variables climáticas		
Para:	Obtener los datos sin intervención manual		
Criterios de aceptación:	El sistema deberá realizar mediciones de temperatura y de húmedas de manera continua cada 15 minutos		
Dependencias:	Sensor DHT22 correctamente configurado		

4.4.2.1.2 Historia de usuario 2: Procesamiento y validación de datos

Tabla 9: Historia de usuario 2: Procesamiento y validación de datos

HISTORIA DE USUARIO			
Id de historia	HU02	Título:	Procesamiento y validación de datos
Prioridad:	Alta	Usuario:	Administrador
Riesgo de desarrollo:	Bajo	Tamaño de la tarea:	Media
Como:	administrador		
Quiero:	Que las mediciones obtenidas por los sensores sean procesadas y validadas		
Para:	Mantener información ambiental confiable para el monitoreo del cultivo		
Criterios de aceptación:	El sistema deberá procesar y validar las mediciones, descartando valores inconsistentes antes de su almacenamiento.		
Dependencias:	Módulo de procesamiento de datos		

4.4.2.1.3 Historia de usuario 3: Detección de anomalías

Tabla 10: Historia de usuario 3: Detección de anomalías

HISTORIA DE USUARIO			
Id de historia	HU03	Título:	Detección de anomalías
Prioridad:	Alta	Usuario:	Técnico agrícola
Riesgo de desarrollo:	Medio	Tamaño de la tarea:	Media
Como:	Técnico agrícola		
Quiero:	Que el sistema identifique automáticamente comportamientos anormales en las variables ambientales monitoreadas		
Para:	Detectar oportunamente cambios ambientales que puedan afectar el cultivo de cacao.		
Criterios de aceptación:	El sistema deberá identificar como anomalías las mediciones que excedan los umbrales establecidos		
Dependencias:	Procesamiento y validación de datos		

4.4.2.1.4 Historia de usuario 4: Almacenamiento temporal de datos

Tabla 11: Historia de usuario 4: Almacenamiento temporal de datos

HISTORIA DE USUARIO			
Id de historia	HU04	Título:	Almacenamiento temporal de datos
Prioridad:	Alta	Usuario:	Administrador
Riesgo de desarrollo:	Bajo	Tamaño de la tarea:	Media
Como:	Administrador		
Quiero:	Que el sistema guarde temporalmente las mediciones obtenidas		
Para:	Para conservar un historial reciente de las mediciones y facilitar su análisis y comparación.		
Criterios de aceptación:	El sistema deberá almacenar hasta 10 mediciones en un buffer circular, reemplazado automáticamente los datos más antiguos cuando se alcance dicho limite		
Dependencias:	Módulo DataBuffer		

4.4.2.1.5 Historia de usuario 5: Generación de datos en formato JSON

Tabla 12: Historia de usuario 5: Generación de datos en formato JSON

HISTORIA DE USUARIO			
Id de historia	HU05	Título:	Generación de datos estructurados
Prioridad:	Alta	Usuario:	Administrador
Riesgo de desarrollo:	Bajo	Tamaño de la tarea:	Media
Como:	Administrador		
Quiero:	Que las mediciones se generen en un formato compatible con el Gateway		
Para:	Asegurar su correcta transmisión hacia el Gateway.		
Criterios de aceptación:	Los datos deben ser generados en un formato JSON válido con todos los campos requeridos.		
Dependencias:	Función preparePayload		

4.4.2.2 Requerimientos funcionales

RF01: Permitir la adquisición de datos meteorológicos.

RF02: Permitir el procesamiento de datos.

RF03: Permitir la detección de anomalías.

RF04: Permitir el análisis de tendencias.

RF05: Permitir generación de datos estructurados en formato JSON.

4.4.2.3 Requerimientos no funcionales

RNF01: Permitir el almacenamiento temporal de hasta 10 mediciones mediante un buffer circular.

RNF02: Permitir la disponibilidad continua del sistema durante su operación.

RNF03: Permitir la escalabilidad permitiendo futuras modificaciones.

RNF04: Permitir seguridad básica para detectar alteraciones durante la transmisión.

RNF05: Permitir la mantenibilidad del código fuente.

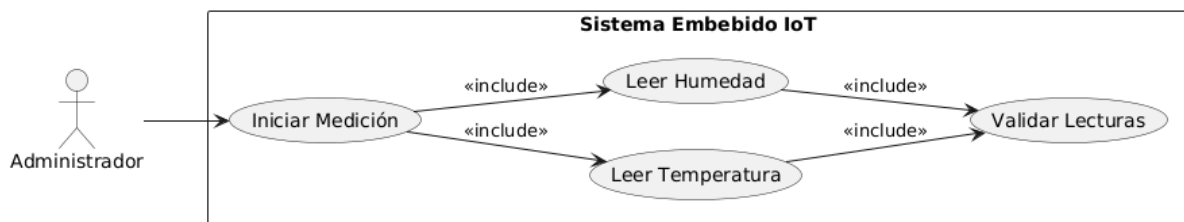
RNF06: Permitir la confiabilidad al momento de validar las lecturas recolectadas.

4.4.3 Diseño del Sistema

4.4.3.1 Casos de Uso

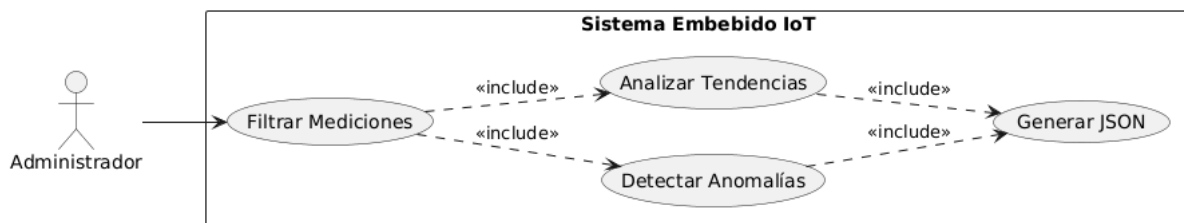
4.4.3.1.1 Caso de uso: Lectura de datos meteorológicos (Entrada)

Ilustración 2: Diagrama de Caso de Uso: Lectura de datos meteorológicos



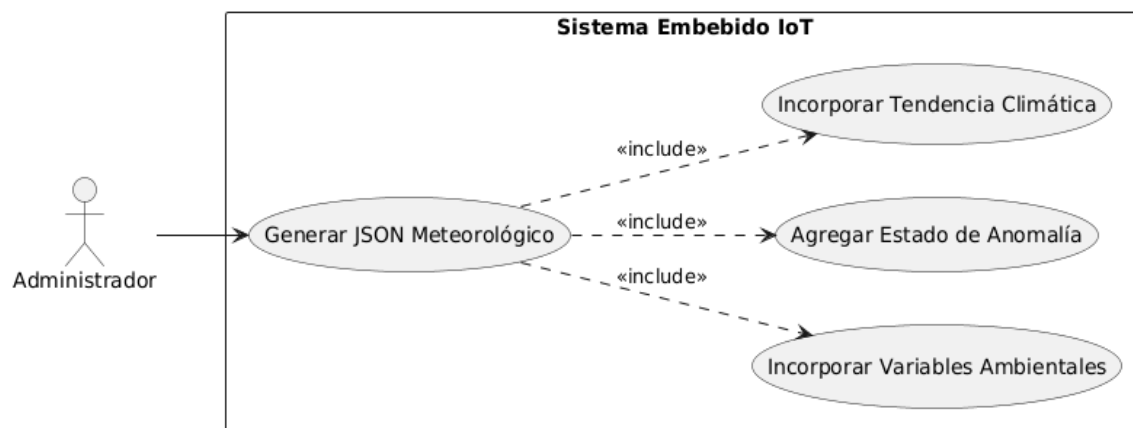
4.4.3.1.2 Caso de uso: Procesamiento inteligente de datos (Proceso)

Ilustración 3: Diagrama de Caso de Uso: Procesamiento inteligente de datos



4.4.3.1.3 Caso de uso: Generación de datos estructurados (Salida)

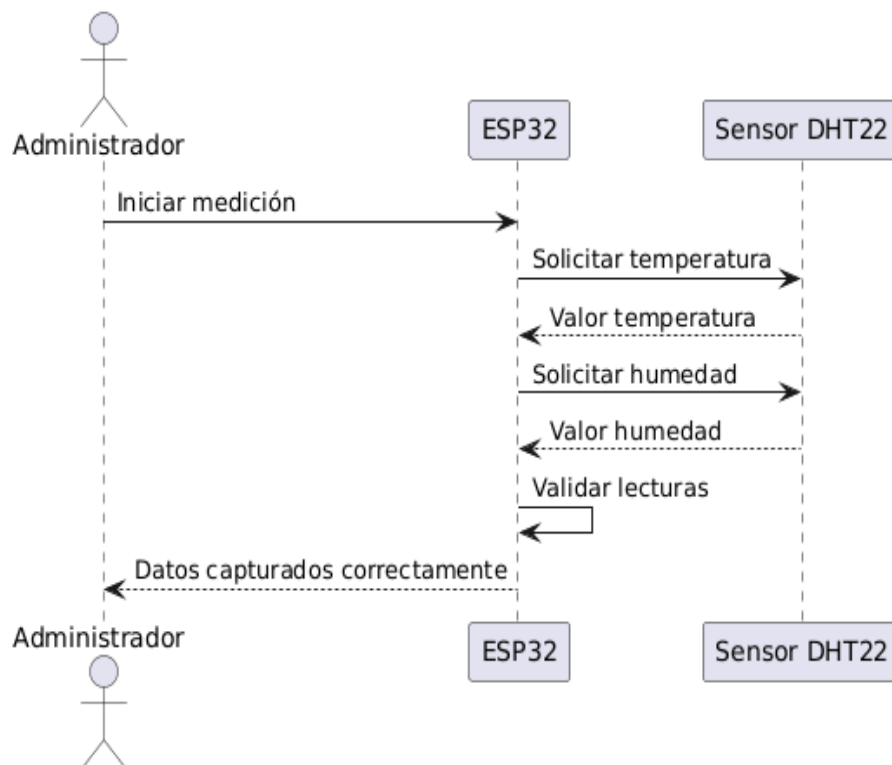
Ilustración 4: Diagrama de Caso de Uso: Procesamiento inteligente de datos



4.4.3.1.4 Diagramas de Secuencia

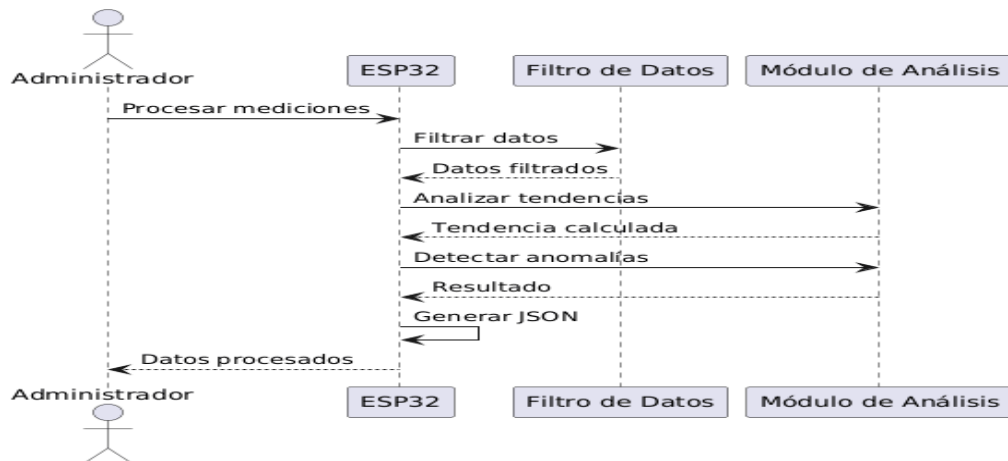
4.4.3.1.5 Diagrama de secuencia: Lectura de Datos Meteorológicos

Ilustración 5: Diagrama de secuencia: Lectura de Datos Meteorológico



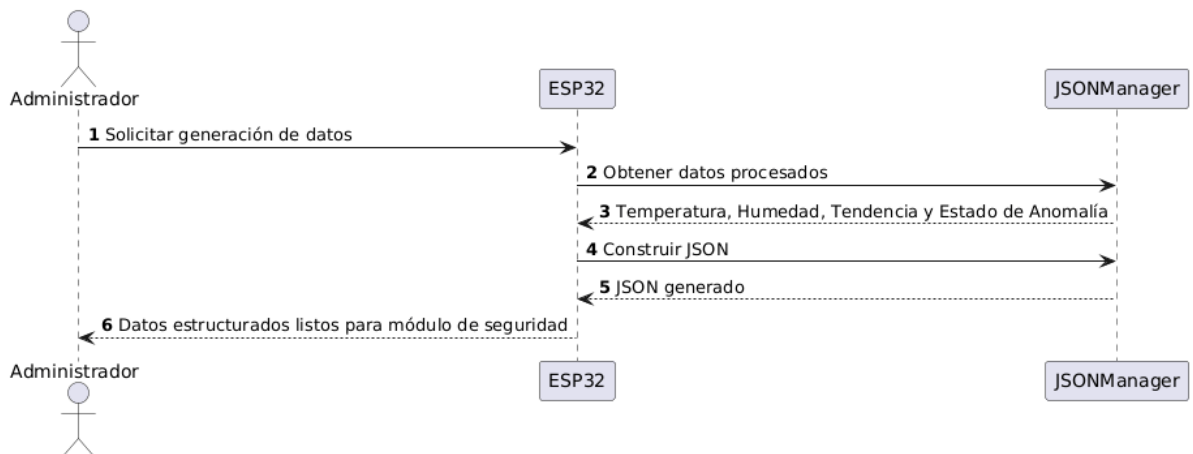
4.4.3.1.6 Diagrama de secuencia: Procesamiento inteligente de Datos

Ilustración 6: Diagrama de secuencia: Procesamiento inteligente de Datos



4.4.3.1.7 Diagrama de secuencia: Generación de datos JSON

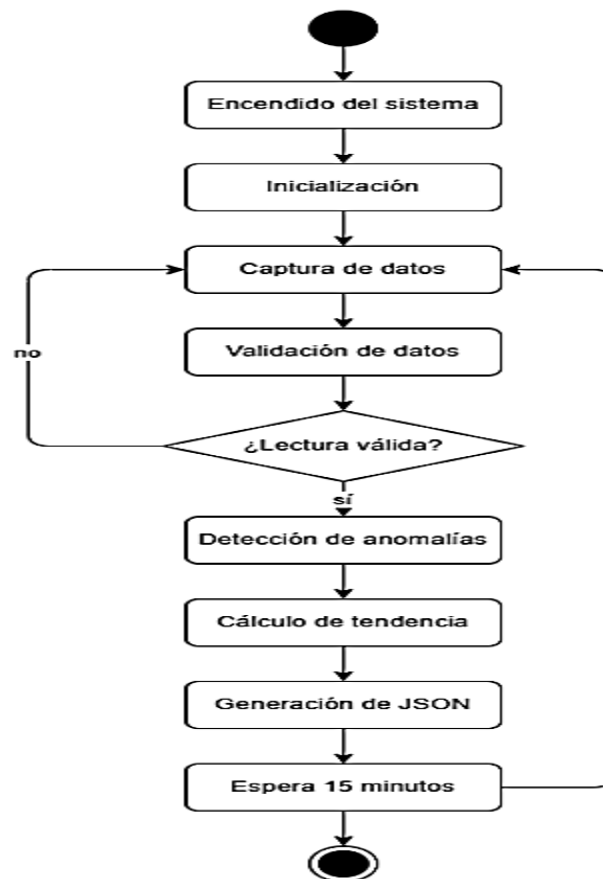
Ilustración 7: Diagrama de secuencia: Generación de datos JSON



4.4.3.2 Diagramas de Estado

4.4.3.2.1 Diagrama de estado: Nodo Meteorológico

Ilustración 8: Diagrama de estado: Nodo Meteorológico

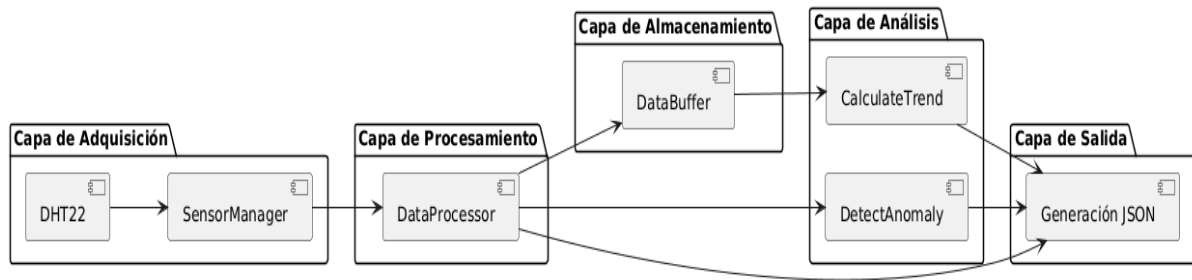


4.4.4 Arquitectura del sistema

4.4.4.1 Arquitectura general

El sistema embebido implementa una arquitectura modular basada en componentes. Cada módulo tiene una responsabilidad específica dentro del proceso de adquisición, procesamiento, almacenamiento temporal y estructurado de datos meteorológico temporal y estructurado de los datos climatológicos. Esta arquitectura facilitará la mantenibilidad, escalabilidad y reutilización de código, permitiendo una separación clara de responsabilidades entre los diferentes componentes del sistema.

Ilustración 9: Arquitectura general



4.4.4.2 Tecnologías utilizadas

Tabla 13: Tecnología utilizadas

Tecnología	Tipo	Uso dentro del proyecto
ESP32	Hardware	Microcontrolador encargado de la adquisición y procesamiento de datos meteorológicos.
DHT22	Hardware	Sensor utilizado para medir temperatura y humedad relativa del ambiente.
Visual Studio Code	IDE	Entorno utilizado para el desarrollo del programa y ejecutar el firmware del ESP32.
PlatformIO	IDE/Extensión	Framework utilizado para la programación del microcontrolador ESP32.
C++	Lenguaje de Programación	Lenguaje utilizado para implementar la lógica del sistema embebido IoT.
DHT Sensor Library	Librería	Permite la lectura y la gestión de datos del sensor DHT22
Hardware Serial	Librería	Permite la comunicación serial entre dispositivos externos y el ESP32
Preferences	Librería	Permite almacenar el número de secuencia en la memoria flash NVS del ESP32, conservándolo entre reinicio del dispositivo
JSON	Formato de datos	Formato utilizado para la estructura de la información obtenida generada por el sistema

4.4.4.3 Roles Scrum

Tabla 14: Roles Scrum

Rol	Participante	Función principal
Product Owner	Ing. Pedro Nivela	Definir requisitos, priorizar los objetivos, funcionalidades y validar el producto.
Scrum Master	Anthony Javier Zambrano Alcivar	Es el líder del equipo de desarrollo es el encargado de que se cumplan las reglas y los objetivos.
Scrum Team	Anthony Javier Zambrano Alcivar	Son quienes desarrollan y entregan el producto final como lo es el sistema embebido.

4.4.4.4 Definición de Hecho (DoD)

4.4.4.5 Criterios Generales

- El código debe ejecutarse correctamente en el medio de desarrollo PlatformIO sin presentar errores de compilación.
- Las funcionalidades implementadas deben cumplir con los requisitos establecidos en las historias de usuario correspondientes.
- Los módulos del sistema deben unirse correctamente sin afectar el funcionamiento de otros componentes.
- Las lecturas recolectadas por los sensores deben ser validas previamente antes de ser procesadas o transmitidas.
- El sistema debe crear correctamente la estructura de datos en formato JSON para así realizar el intercambio de información.
- Los algoritmos de procesamiento deben ejecutar correctamente el filtrado de los datos, análisis de las tendencias y detección de anomalías.
- El firmware debe mantener una arquitectura modular que facilite en mantenimiento y la escalabilidad del proyecto
- Las pruebas funcionales deben demostrar que las mediciones recolectadas corresponden al valor valido proporcionados por los sensores conectados al ESP32.
- La documentación técnica del código fuente debe mantenerse actualizado durante el desarrollo del proyecto.

- El sistema debe almacenar temporalmente las mediciones en un buffer circular de 10 posiciones.

4.4.4.6 Criterios Específicos del Proyecto

- El sistema debe realizar mediciones continuas de temperatura y de humedad cada 15 minutos sin la intervención manual.
- Las mediciones obtenidas de la temperatura y la humedad deben validarse antes de ser procesadas para así evitar el uso de datos anómalos o nulos.
- El módulo de filtrado deberá eliminar los valores atípicos y calcular una medición representativa a partir de múltiples muestras obtenidas por el sensor.
- El sistema debe guardar las mediciones validas en la memoria temporal para así identificar tendencias en el cambio del clima.
- El análisis de las tendencias debe ser comparadas con las mediciones actuales con la previamente almacenadas para así determinar si las variables ambientales se encuentran en un rango estable, en aumento o en descenso.
- El módulo de detección de anomalías debe de reconocer los cambios significativos en las variables monitoreadas cuando superen los umbrales definidos en el sistema.
- El sistema debe crear correctamente una estructura JSON que contenga el número de secuencia, la temperatura y la humedad. La tendencia y el estado de anomalía se calculan de forma interna en el nodo como diagnóstico local y no se incluyen en la estructura transmitida.
- El sistema debe almacenar la última medición realizada para utilizarla como referencia en análisis posteriores.
- El buffer circular debe conservar las ultimas 10 mediciones, remplazando automáticamente los datos más antiguos al alcanzar su capacidad máxima.
- La arquitectura modular debe permitir la integración de nuevos sensores sin afectar el funcionamiento de los módulos existentes.

4.4.5 Product Backlog

Tabla 15: Product Backlog

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Lectura automática de la temperatura	HU01	Alta	Por hacer

Nro.	Tarea	Historia de usuario	Prioridad	Estado
2	Lectura automática de la humedad	HU01	Alta	Por hacer
3	Validación de datos obtenidos por los sensores	HU01	Alta	Por hacer
4	Implementación de medición continua cada 15 minutos.	HU01	Media	Por hacer
5	Almacenamiento temporal mediante múltiples muestras	HU01	Media	Por hacer
6	Procesamiento de datos de temperatura	HU02	Alta	Por hacer
7	Validación de las mediciones obtenidas por los sensores	HU02	Alta	Por hacer
8	Filtrado de los datos mediante varias muestras	HU02	Alta	Por hacer
9	Almacenamiento temporal de datos procesados	HU02	Media	Por hacer
10	Identificación de cambios bruscos de la temperatura	HU03	Alta	Por hacer
11	Identificación de cambios bruscos de la humedad	HU03	Alta	Por hacer
12	Marcado local de las anomalías detectadas	HU03	Alta	Por hacer
13	Implementación del buffer circular de 10 posiciones	HU04	Alta	Por hacer
14	Almacenamiento de la última medición como referencia	HU04	Alta	Por hacer
15	Reemplazo automático de datos antiguos en el buffer	HU04	Media	Por hacer
16	Generación de mensajes en formato JSON	HU05	Alta	Por hacer
17	Inclusión de secuencia, temperatura y humedad en JSON	HU05	Alta	Por hacer
18	Detección local del estado de anomalía	HU05	Media	Por hacer
19	Cálculo local de tendencias	HU05	Media	Por hacer

4.4.6 Sprint 1: Medición automática de datos

4.4.6.1 Planificación del Sprint

Tabla 16: Planificación del Sprint 1

SPRINT 1: Medición Automática de datos			
Duración:	1 semana (05/01/2026 – 12/01/2026)	Prioridad:	Alta
Objetivo:	Implementar la adquisición automática de datos climatológico s a través de los sensores conectados al ESP32, permitiendo la lectura continua de temperatura y de humedad cada 15 minutos.		
Historias de usuario:	HU01: Medición automática de datos		
Tareas de desarrollo:	• Instalación y configuración inicial del ESP32. • Configuración del sensor DHT22. • Implementación de la lectura automática de temperatura. • Implementación de la lectura automática de la humedad. • Configuración del intervalo de medición cada 15 minutos. • Pruebas de funcionamiento del proceso de adquisición de datos. • Ajustes finales y corrección de errores		
Plan de entrega:	• Configuración del ESP32. • Instalar y configurar el entorno de PlatformIO, dependencias y parámetros de comunicación para el desarrollo del sistema embebido. • Tiempo de desarrollo: 4 horas • Fecha de Inicio y Fecha Final: 06/01/2026 - 06/01/2026. • Implementación de lectura de sensores. • Crear los módulos encargados de recopilar automáticamente las mediciones de temperatura y humedad.		

SPRINT 1: Medición Automática de datos

- Tiempo de desarrollo: 7 horas
- Fecha de Inicio y Fecha Final: 07/01/2026 - 07/01/2026.
- **Configuración del intervalo de medición.**
 - Configurar el sistema para realizar mediciones automáticas cada 15 minutos.
 - Tiempo de desarrollo: 8 horas
 - Fecha de Inicio y Fecha Final: 08/01/2026 - 08/01/2026.
- **Pruebas de funcionamiento del sistema.**
 - Verificar el correcto funcionamiento de las mediciones y la estabilidad de los datos recolectados.
 - Tiempo de desarrollo: 2 horas.
 - Fecha de Inicio y Fecha Final: 09/01/2026 - 09/01/2026.
- **Ajustes finales y corrección de errores.**
 - Resolver cualquier problema, error, inconveniente encontrado en las pruebas, finalizar los detalles menores y entregar el proyecto.
 - Tiempo de desarrollo: 21 horas
 - Fecha de Inicio y Fecha Final: 09/01/2026 - 12/01/2026.

4.4.6.2 Sprint Backlog

Tabla 17: Sprint Backlog del Sprint 1

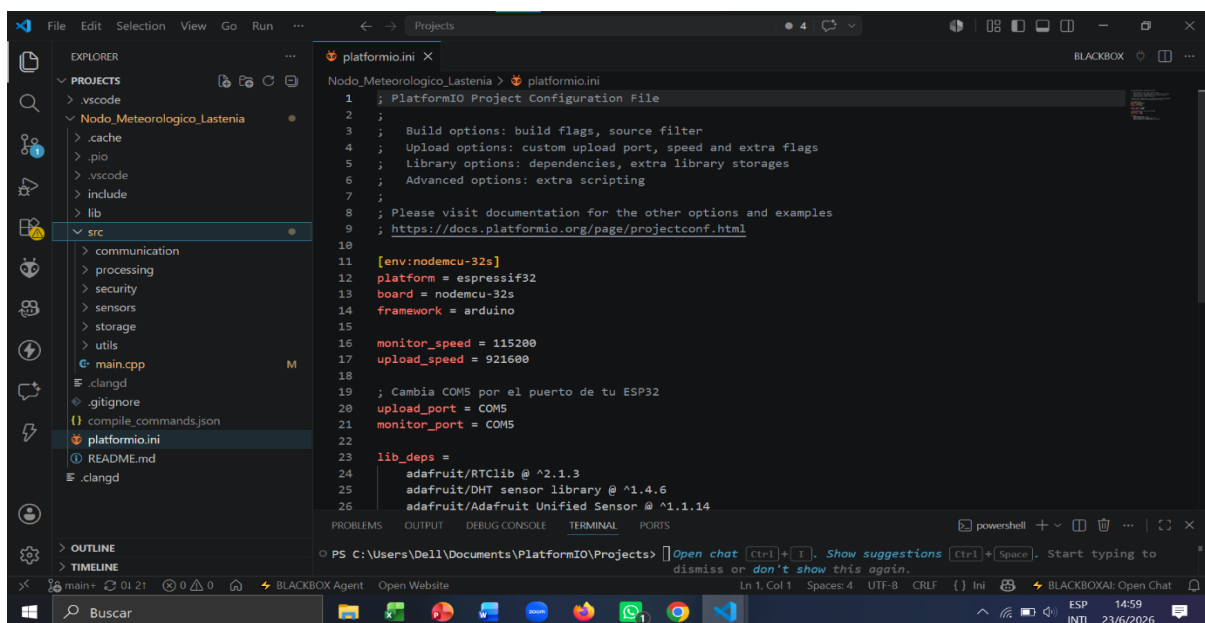
Historia de usuario.	Tarea	Duración estimada
HU01	Configuración del ESP32	4 horas
HU01	Implementación de lectura de sensores	7 horas
HU01	Configuración del intervalo de medición cada 15 minutos	8 horas
HU01	Pruebas de funcionamiento del sistema	2 horas
HU01	Ajustes finales y corrección de errores	21 horas

4.4.6.3 Desarrollo del Sprint

4.4.6.3.1 Funcionalidad de adquisición automática de datos

En esta funcionalidad se desarrolló el mecanismo de lectura automática del sensor DHT22, el cual permite obtener datos de temperatura y humedad de manera continua cada 15 minutos, sin necesidad de la intervención del usuario. El sistema espera 2000ms antes de iniciar la lectura para que el sensor alcance estabilidad térmica. Luego de ese tiempo toma 5 muestras consecutivas con un intervalo de 500ms entre cada una, almacenándola temporalmente en un arreglo para obtener una lectura continua del sensor.

Ilustración 10: Entorno de desarrollo Utilizado



Esta ilustración 11 muestra el entorno de desarrollo utilizado para la implementación del nodo sensor, donde se aprecia la configuración del proyecto en PlatformIO sobre Visual Studio Code. Este entorno permite la gestión de las dependencias, librerías y también la compilación del firmware para que el microcontrolador ESP32, el cual actuara como una unidad central de procesamiento del nodo meteorológico. La estructura del proyecto se organiza en módulos independientes que permiten la mantenibilidad del proyecto.

Ilustración 11: Adquisición automática de datos

```
// Toma SAMPLE_COUNT muestras y aplica media recortada (descarta min y max)
static SensorData sampleAndFilter() {
    delay(SENSOR_STABILIZATION_TIME);

    SensorData samples[SAMPLE_COUNT];
    for (int i = 0; i < SAMPLE_COUNT; i++) {
        samples[i] = sensorManager.readSensors();
        delay(SAMPLE_INTERVAL);
    }
}
```

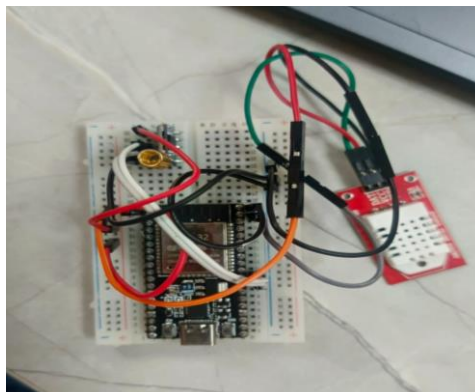
El fragmento de código presentado implementa la adquisición automática de datos del sensor DHT22. Inicialmente se declara un arreglo para almacenar las cinco muestras consecutivas de la temperatura y de humedad. Posteriormente, mediante un ciclo de repetición, el sistema realiza las lecturas con un intervalo de 500 ms entre cada una, registrando así las mediciones necesarias para la adquisición continua de los datos climatológicos.

4.4.6.4 Sprint Review

Durante el primer Sprint se implementó la funcionalidad de la adquisición automática de datos meteorológicos mediante el microcontrolador ESP32 y el sensor DHT22. Se desarrollaron los módulos necesarios para obtener lecturas de temperatura y humedad, validar la información recolectada y almacenarla temporalmente para su posterior procesamiento dentro del sistema embebido.

Evidencia de implementación

Ilustración 12: Sensor DHT22 conectado al ESP32



Retroalimentación recibida:

- Se recomendó mejorar la organización del código mediante la separación de responsabilidades en módulos independientes.
- Se sugirió la implementación de validaciones que permitan detectar lecturas inválidas provenientes del sensor.
- Se indicó documentar adecuadamente las funciones desarrolladas para facilitar el mantenimiento del sistema.

Mejoras realizadas:

- Se reorganizó el proyecto utilizando módulos independientes para sensores, almacenamiento y configuración.
- Se incorporaron mecanismos de validación para evitar procesamientos de lecturas erróneas.
- Se añadieron comentarios y documentación básica en las funciones principales del sistema.

Resultados del Sprint: Se completó satisfactoriamente la adquisición automática de datos climáticos, permitiendo al sistema obtener mediciones confiables de la temperatura y humedad para las siguientes etapas.

4.4.6.5 Pruebas del Sprint

4.4.6.5.1 Pruebas de Caja negra

- **Pruebas del comportamiento de la lectura y el muestreo ante distintas entradas del sensor**

Tabla 18: Pruebas de caja negra – Sprint 1

Nro.	Entrada	Salida esperada	Resultado
1	El sensor DHT22 conectado y estable	Se obtiene 5 muestras consecutivas de temperatura y humedad	Pasa

2	Espera de estabilización de 2000 ms antes de medir	El sistema aguarda ese tiempo antes de la primera muestra	Pasa
3	Intervalo de 500 ms entre muestras	Cada muestra con la pausa configurada	Pasa
4	Sensor desconectado o sin respuesta	La lectura se marca como no válida	Pasa

4.4.6.5.2 Pruebas de Caja Blanca

➤ Pruebas a la lógica interna de la lectura y el muestreo del sensor

Tabla 19: Pruebas de caja blanca – Sprint 1

Método	Acción esperada	Acción obtenida	Observación
Lectura de temperatura	Ejecutar la lectura del sensor DHT22 para obtener la temperatura.	El sistema obtiene correctamente el valor de temperatura proporcionado por el sensor.	La lectura de temperatura se realizó correctamente
Lectura de humedad	Ejecutar la lectura del sensor DHT22 para obtener la humedad.	El sistema obtuvo correctamente el valor de humedad proporcionado por el sensor.	La funcionalidad de respondió de acuerdo con lo esperado durante las pruebas realizadas.
Validación de lectura NaN	Ejecutar la condición isnan() cuando el sensor	El sistema detectó la lectura inválida y	En las primeras pruebas fue necesario ajustar la validación para descartar

Método	Acción esperada	Acción obtenida	Observación
	retorna una lectura inválida.	evitó que continuara con el proceso.	correctamente las lecturas inválidas. Después de la corrección la función operó de forma adecuada.
Muestreo múltiple	Ejecutar el ciclo for encargado de obtener las 5 lecturas consecutivas.	El ciclo completo las 5 iteraciones programadas y almacenó las muestras obtenidas.	Inicialmente se ajustó el intervalo entre muestras para mejorar la estabilidad de las lecturas. Una vez realizadas el ajuste, el proceso funcionó correctamente.

4.4.6.6 Incremento y Entregables del Sprint

Se completaron satisfactoriamente las siguientes funcionalidades planificadas en el Sprint 1:

- Configuración e inicialización del microcontrolador ESP32.
- Integración y configuración del sensor DHT22.
- Implementación de la lectura automática de temperatura y humedad.
- Implementación de la validación básica de las mediciones obtenidas.
- Verificación del correcto funcionamiento del proceso de adquisición de datos.

4.4.6.7 Estado actual del Product Backlog

Tabla 20: Estado actual del Product Backlog – Sprint 1

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Lectura automática de la temperatura	HU01	Alta	Cumplida
2	Lectura automática de la humedad	HU01	Alta	Cumplida
3	Validación de datos obtenidos por los sensores	HU01	Alta	Cumplida
4	Implementación de medición continua cada 15 minutos	HU01	Media	Cumplida

4.4.7 Sprint 2: Procesamiento y validación de datos

4.4.7.1 Planificación del Sprint

Tabla 21: Planificación del Sprint 2

SPRINT 2: Procesamiento y validación de datos			
Duración:	1 semana (12/01/2026 – 19/01/2026)	Prioridad:	Alta
Objetivo:	Implementar los mecanismos de procesamiento, validación y filtrado de las mediciones obtenidas por los sensores, garantizando la calidad y confiabilidad de los datos antes de ser utilizados por el sistema embebido.		
Historias de usuario:	HU02: Procesamiento y validación de datos		
Tareas de desarrollo:	• Procesamiento de mediciones de temperatura y humedad. • Validación de datos obtenidos por los sensores. • Aplicación de filtro para eliminar de valores atípicos. • Verificación del funcionamiento del procesamiento y validación. • Ajustes finales y corrección de errores		

SPRINT 2: Procesamiento y validación de datos

Plan de entrega:

- **Procesamiento de mediciones de temperatura y humedad.**
 - Desarrollar los procedimientos encargados de preparar y de organizar las mediciones recolectadas por los sensores para su posterior análisis.
 - Tiempo de desarrollo: 4 horas
 - Fecha de Inicio y Fecha Final: 13/01/2026 - 13/01/2026.
- **Validación de datos obtenidos por los sensores.**
 - Implementar mecanismos que permitan verificar que las lecturas obtenidas por los sensores son válidas y consistentemente antes de ser utilizadas por el sistema.
 - Tiempo de desarrollo: 5 horas
 - Fecha de Inicio y Fecha Final: 14/01/2026 - 14/01/2026.
- **Aplicación de filtros para eliminación de valores atípicos.**
 - Realizar algoritmos de filtrado basados en varias muestras para así reducir el impacto de las lecturas erróneas.
 - Tiempo de desarrollo: 5 horas
 - Fecha de Inicio y Fecha Final: 15/01/2026 - 15/01/2026.
- **Verificación del funcionamiento del procesamiento y validación de datos.**
 - Ejecutar pruebas para comprobar que las mediciones sean procesadas correctamente, descartando las lecturas invalidadas y verificando así el funcionamiento del algoritmo de filtrado mediante media recortada.
 - Tiempo de desarrollo: 5 horas.
 - Fecha de Inicio y Fecha Final: 16/01/2026 - 16/01/2026.
- **Pruebas de funcionamiento del sistema .**
 - Verificar el correcto funcionamiento de los módulos de validación, filtrado y de almacenamiento temporal, para así garantizar la integridad de los datos procesados.
 - Tiempo de desarrollo: 2 horas.
 - Fecha de Inicio y Fecha Final: 16/01/2026 - 16/01/2026.
- **Ajustes finales y corrección de errores.**
 - Resolver cualquier problema, error, inconveniente encontrado en las pruebas, finalizar los detalles menores y entregar el proyecto.
 - Tiempo de desarrollo: 21 horas

SPRINT 2: Procesamiento y validación de datos	
	Fecha de Inicio y Fecha Final: 12/01/2026 - 19/01/2026.

4.4.7.2 Sprint Backlog

Tabla 22: Sprint Backlog – Sprint 2

Historia de usuario.	Tarea	Duración estimada
HU02	Procesamiento de mediciones de temperatura y la humedad.	4 horas
HU02	Validación de datos obtenidos por los sensores	5 horas
HU02	Aplicación de filtros para la eliminación de valores atípicos	5 horas
HU02	Verificación del funcionamiento del procesamiento y validación	5 horas
HU02	Pruebas de funcionamiento del sistema	2 horas
HU02	Ajustes finales y corrección de errores	21 horas

4.4.7.3 Desarrollo del Sprint

4.4.7.3.1 Funcionalidad de procesamiento y validación de datos

En esta funcionalidad se desarrolló el mecanismo de procesamiento y de la validación de las lecturas que se han obtenido por los sensores, garantizando que únicamente los datos coherentes y dentro de rangos físicos aceptables sean utilizados en el sistema.

Ilustración 13: Función de validación de datos

```
7  SensorData DataProcessor::validate(const SensorData& data) const {
8      SensorData processed = data;
9
10     // Rechaza si el sensor ya marcó la lectura como inválida o devolvió NaN
11     if (!data.valid || isnan(data.temperature) || isnan(data.humidity)) {
12         processed.valid = false;
13         return processed;
14     }
15
16     // Rechaza si la temperatura está fuera del rango esperado para el vivero
17     if (data.temperature < TEMP_MIN || data.temperature > TEMP_MAX) {
18         processed.valid = false;
19         return processed;
20     }
21
22     // Rechaza si la humedad está fuera del rango físicamente posible
23     if (data.humidity < HUM_MIN || data.humidity > HUM_MAX) {
24         processed.valid = false;
25         return processed;
26     }
27
28     processed.valid = true;
29     return processed;
30 }
```

El sistema verifica en primera instancia que la lectura no contenga valores nulos, descartando inmediatamente si el sensor reportó un fallo. Posteriormente valida que la temperatura se encuentre entre 15 grados y 45 grados, y que la humedad esté entre 0% y 100%, que son los rangos físicamente aceptables para el entorno del vivero. Adicionalmente se aplica una media recortada a las 5 muestras adquiridas, descartando el valor máximo y mínimo de cada variable para eliminar las lecturas falsas, obteniendo así un valor representativo del estado real del ambiente.

Ilustración 14: Algoritmo de filtrado por media recortada

```
57  SensorData filtered;
58  filtered.temperature = (sumTemp - minTemp - maxTemp) / (SAMPLE_COUNT - 2);
59  filtered.humidity    = (sumHum - minHum - maxHum) / (SAMPLE_COUNT - 2);
60  filtered.timestamp   = millis() / 1000UL;
61  filtered.valid       = isfinite(filtered.temperature) && isfinite(filtered.humidity);
62  return filtered;
```

4.4.7.4 Sprint Review

Durante el Sprint 2 se realizó la funcionalidad de procesamiento y validación de datos climatológicos obtenidos por el sensor DHT22. Se implementaron mecanismos para verificar la validez de las mediciones obtenidas, se aplicaron filtros para reducir el impacto de valores no numéricos NaN o que estuvieran fuera de los rangos establecidos para temperatura y humedad. Asimismo, se incorporó un algoritmo de filtrado basado en media recortada para reducir el efecto de lectura atípica y mejorar la confiabilidad de los datos procesados.

Evidencia de implementación

Ilustración 15: Verificación de la validez de las mediciones

```
7  SensorData DataProcessor::validate(const SensorData& data) const {
8      SensorData processed = data;
9
10     // Rechaza si el sensor ya marcó la lectura como inválida o devolvió NaN
11     if (!data.valid || isnan(data.temperature) || isnan(data.humidity)) {
12         processed.valid = false;
13         return processed;
14     }
15 }
```

Retroalimentación recibida

- Se recomienda fortalecer, los mecanismos de validación para evitar el procesamiento de los datos erróneos.
- Se sugiere incorporar filtros que reduzcan el efecto de la lectura errónea, producida por el ruido del sensor.
- Se indica la optimización del almacenamiento temporal, para así garantizar la disponibilidad de los datos en ciclos posteriores.

Mejoras realizadas

- Se implementaron validaciones para verificar la integridad de las mediciones climáticas antes de ser procesadas.
- De agrego algoritmos de filtrado basado en múltiples muestras para así mejora la precisión de los datos.
- Se optimizo el almacenamiento temporal utilizando un buffer circular, asimismo se mejoró la integración de los módulos de adquisición, procesamiento y almacenamiento.

Resultados del Sprint: Se completo satisfactoriamente la funcionalidad de procesamiento y validación de datos garantizando así, que las mediciones realizadas por los sensores sean consistentes y confiables, antes de ser utilizadas en las etapas de detección de anomalías y la generación de datos estructurados.

4.4.7.5 Pruebas del Sprint

4.4.7.5.1 Pruebas de Caja Negra

- **Pruebas del comportamiento de la validación y el filtrado ante distintas entradas**

Tabla 23: Pruebas de caja negra – Sprint 2

Nro.	Entrada	Salida esperada	Resultado
1	Temperatura de 30 °C y humedad de 75 % (dentro de rango)	La medición es aceptada como válida	Pasa
2	Temperatura de 50 °C (fuera del rango 15–45 °C)	La medición se descarta	Pasa
3	Humedad de 120 % (fuera del rango 0–100 %)	La medición se descarta	Pasa
4	El sensor devuelve un valor NaN	La medición se descarta de procesarse	Pasa
5	Cinco muestras con una lectura atípica	El filtrado por media recortada descarta el valor máximo y el mínimo y promedia el resto	Pasa

4.4.7.5.2 Pruebas de Caja Blanca

- **Pruebas a la lógica interna de la validación de rangos y el filtrado por media recortada**

Tabla 24: Pruebas de caja blanca – Sprint 2

Método	Acción esperada	Acción obtenida	Observación
Validación de temperatura	Ejecutar la condición que verifica si la temperatura se encuentra entre 15°C y 45°C.	La función retornó true para valores válidos y false para valores fuera del rango establecido.	Durante las primeras pruebas fue importante ajustar los límites de validación para adaptar a las condiciones del vivero. Posteriormente, la función validó correctamente las mediciones.
Validación de humedad	Ejecutar la condición que verifica si la humedad se encuentra entre 0 % y 100 %.	La función aceptó únicamente valores dentro del rango establecido.	La validación de humedad se ejecutó conforme a lo esperado, permitiendo únicamente el procesamiento de valores válidos.
Validación de lectura NaN	Ejecutar la condición isnan() cuando el sensor retorna una lectura inválida	El sistema detectó la lectura inválida y la descarta sin procesarla..	En las primeras pruebas se detectó, que algunas lecturas inválidas continuaban en el flujo del programa. Después de ajustar la condición de validación, el problema quedó solucionado.
Muestreo múltiple	Ejecutar el ciclo for encargado de obtener las muestras.	El ciclo completó las 5 iteraciones configuradas.	Fue necesario ajustar el intervalo entre muestras para obtener lecturas más estables. Una vez realizado el ajuste, el proceso funcionó correctamente.

Método	Acción esperada	Acción obtenida	Observación
Filtrado por media recortada	Ejecutar el algoritmo que elimina el valor máximo y mínimo antes del promedio.	El promedio se calculó correctamente utilizando las muestras restantes.	El algoritmo redujo la influencia de valores atípicos, obteniendo así mediciones más estables y representativas del entorno.

4.4.7.6 Incremento y Entregables del Sprint

Sprint 2: Procesamiento y validación de datos

Se completaron las siguientes funcionalidades:

- Procesamiento de mediciones de temperatura y humedad.
- Implementación de la validación de datos obtenidos por los sensores.
- Implementación del filtrado mediante media recortada.
- Integración del módulo de procesamiento con el módulo de adquisición.
- Verificación del correcto funcionamiento del procesamiento y validación.

4.4.7.7 Estado actual del Product Backlog

Tabla 25: Estado actual del Product Backlog – Sprint 2

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Procesamiento de datos de temperatura	HU02	Alta	Cumplida
2	Validación de las mediciones obtenidas por los sensores	HU02	Alta	Cumplida
3	Aplicación del filtrado mediante media recortada	HU02	Alta	Cumplida

4.4.8 Sprint 3: Detección de Anomalías

4.4.8.1 Planificación del Sprint

Tabla 26: Planificación del Sprint 3

SPRINT 3: Detección de Anomalías			
Duración:	1 semana (19/01/2026 – 26/01/2026)	Prioridad:	Alta
Objetivo:	Implementar mecanismos de análisis que permitan identificar cambios anormales en las variables ambientales monitoreadas.		
Historias de usuario:	HU03: Detección de Anomalías		
Tareas de desarrollo:	• Comparación de las mediciones actuales y anteriores. • Configuración de umbrales para la detección de anomalías. • Implementación del algoritmo de detección de anomalías. • Marcado local de los cambios significativos. • Integración con el módulo de procesamiento de datos. • Pruebas de funcionamiento del sistema • Ajustes finales y corrección de errores		
Plan de entrega:	• Comparación entre mediciones actuales y anteriores. • Implementar el mecanismo que utiliza la medición anterior almacenada para compararla con la medición actual y detectar así variaciones significativas. • Tiempo de desarrollo: 6 horas • Fecha de Inicio y Fecha Final: 19/01/2026 - 19/01/2026. • Configuración de umbrales para detección de anomalías. • Definir los límites de variación permitidos para la temperatura y la humedad, utilizados como referencia para la detección de comportamientos anormales.		

SPRINT 3: Detección de Anomalías

- Tiempo de desarrollo: 4 horas
- Fecha de Inicio y Fecha Final: 20/01/2026 - 20/01/2026.
- **Implementación del algoritmo de detección de anomalías.**
 - Desarrollar la lógica encargada de determinar si una medición presenta una variación fuera de los parámetros establecidos.
 - Tiempo de desarrollo: 5 horas
 - Fecha de Inicio y Fecha Final: 21/01/2026 - 21/01/2026.
- **Marcado local de los cambios detectados significativos.**
 - Implementar mecanismos que permitan marcar una medición como anómala, cuando detecte variaciones importantes en las condiciones ambientales.
 - Tiempo de desarrollo: 5 horas.
 - Fecha de Inicio y Fecha Final: 22/01/2026 - 22/01/2026.
- **Pruebas de funcionamiento del sistema.**
 - Verificar el correcto funcionamiento del módulo de detección de anomalías y su integración con los demás módulos de adquisición y procesamiento de datos.
 - Tiempo de desarrollo: 2 horas.
 - Fecha de Inicio y Fecha Final: 23/01/2026 - 23/01/2026.
- **Ajustes finales y corrección de errores.**
 - Resolver cualquier problema, error, inconveniente encontrado en las pruebas, finalizar los detalles menores y entregar el proyecto.
 - Tiempo de desarrollo: 22 horas
 - Fecha de Inicio y Fecha Final: 19/01/2026 - 26/01/2026.

4.4.8.2 Sprint Backlog

Tabla 27: Sprint Backlog – Sprint 3

Historia de usuario.	Tarea	Duración estimada
HU03	Comparación de mediciones actuales y anteriores	6 horas
HU03	Configuración de umbrales para detección de anomalías	4 horas

Historia de usuario.	Tarea	Duración estimada
HU03	Implementación del algoritmo de detección de anomalías	5 horas
HU03	Marcado local de los cambios detectados significativo	5 horas
HU03	Pruebas de funcionamiento del sistema	2 horas
HU03	Ajustes finales y corrección de errores	22 horas

4.4.8.3 Desarrollo del Sprint

4.4.8.3.1 Funcionalidad de detección de anomalías

En esta funcionalidad se realizó el mecanismo de detección de variaciones anormales en las lecturas del sensor, comparando la medición actual con la medición anterior. El sistema calcula la diferencia absoluta entre la temperatura actual y la anterior, y entre la humedad actual y la anterior. Si la temperatura cambia más de los 2°C o la humedad más del 5% entre un ciclo y el siguiente, la medición se marcará como una anomalía. Este estado se registra localmente en el nodo como diagnóstico interno y queda disponible en el monitor del dispositivo, sin incorporarse al payload JSON que se transmite.

Ilustración 16: Comparación de mediciones para la detección de anomalías

```

96     return fabsf(current.temperature - previous.temperature) > ANOMALY_TEMP_DELTA ||
97         fabsf(current.humidity - previous.humidity) > ANOMALY_HUM_DELTA;
98 }
```

4.4.8.4 Sprint Review

Durante el Sprint 3 se realizó la funcionalidad de detección de anomalías en la variable climatológica monitoreadas por el sistema. Además de eso se implementaron mecanismos de comparación entre mediciones actuales y anteriores, permitiendo así identificar cambios significativos en los valores de la temperatura y de la humedad. No obstante, se configuraron los umbrales de referencia para determinar cuándo una variación puede ser considerada

medición anómala, para así facilitar la identificación del comportamiento fuera de los parámetros establecidos.

Evidencia de implementación

Ilustración 17: Implementación de la función de detección de anomalías

```
// Retorna true si temperatura o humedad superan los umbrales definidos en Config.h
static bool detectAnomaly(const SensorData& current, const SensorData& previous) {
    return fabsf(current.temperature - previous.temperature) > ANOMALY_TEMP_DELTA ||
           fabsf(current.humidity - previous.humidity) > ANOMALY_HUM_DELTA;
}
```

Retroalimentación recibida

- Se recomienda ajustar los umbrales de detección para así evitar falsos positivos durante la medición.
- También se sugiere optimizar la comparación entre registros constantes para así mejorar la precisión en la detección.
- Se indicó documentar los criterios utilizados para clasificar una medición como errónea.

Mejoras realizadas

- Se optimizaron los parámetros de comparación entre las mediciones consecutivas.
- Se adjuntaron los umbrales de la detección para mejorar la precisión del análisis.
- Se fortaleció la integración entre los módulos de procesamiento y la detección de las anomalías.
- Por último, se documentó la lógica utilizada para identificar variaciones significativas en los datos meteorológicos.

Resultados del Sprint: se completó satisfactoriamente la funcionalidad de la detección de las anomalías, permitiendo así identificar los cambios significativos en las condiciones climáticas monitoreadas. Esta funcionalidad mejora la capacidad del sistema para así reconocer el comportamiento extraño en los datos y preparar la información para las siguientes etapas del proceso de monitoreo meteorológico.

4.4.8.5 Pruebas del Sprint

4.4.8.5.1 Pruebas de Caja Negra

- Pruebas del comportamiento de la detección de anomalías ante distintas variaciones

Tabla 28: Pruebas de caja negra – Sprint 3

Nro.	Entrada	Salida esperada	Resultado
1	Temperatura actual 28 °C, anterior 31 °C (diferencia de 3 °C, mayor a 2 °C)	Se marca como anomalía	Pasa
2	Temperatura actual 28 °C, anterior 28.5 °C (diferencia de 0.5 °C)	No se marca como anomalía	Pasa
3	Humedad actual 70 %, anterior 78 % (diferencia de 8 %, mayor a 5 %)	Se marca como anomalía	Pasa
4	Temperatura que sube más de 0.5 °C respecto a la anterior	La tendencia se calcula como “subiendo”	Pasa
5	Temperatura que se mantiene dentro de 0.5 °C	La tendencia se calcula como “estable”	Pasa

4.4.8.5.2 Pruebas de Caja Blanca

- Pruebas a la lógica interna de la detección de anomalías y el cálculo de tendencia

Tabla 29: Pruebas de caja blanca – Sprint 3

Método	Acción esperada	Acción obtenida	Observación
Comparación entre mediciones	Comparar la medición actual con la anterior.	El sistema obtuvo correctamente la diferencia entre ambas mediciones.	La comparación se ejecutó correctamente y permitió calcular la variación registrada.
Detección de anomalías de temperatura	Supera el umbral de 2°C.	El sistema marco la medición como una anomalía.	La detección funcionó correctamente para variaciones superiores al umbral configurado.
Detección de anomalías de humedad	Superar el umbral del 5%.	El sistema identificó la anomalía.	El algoritmo detectó correctamente los cambios significativos de la humedad
Validación de variaciones normales	Ingresar cambios menores al umbral.	La medición no fue marcada como anomalía.	El sistema evito generar falsas alertas cuando los cambios fueron normales.

4.4.8.6 Incremento y Entregables del Sprint

Sprint 3: Detección de anomalías

Se completaron las siguientes funcionalidades:

- Comparación de mediciones actuales y anteriores.
- Configuración de umbrales para la detección de anomalías.
- Implementación del algoritmo de detección de anomalías.
- Identificación de cambios significativos en temperatura y humedad.
- Marcado local de las anomalías detectadas.

4.4.8.7 Estado actual del Product Backlog

Tabla 30: Estado actual del Product Backlog – Sprint 3

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Identificación de cambios bruscos de la temperatura	HU03	Alta	Cumplida
2	Identificación de cambios bruscos de la humedad	HU03	Alta	Cumplida
3	Marcado local de las anomalías detectadas	HU03	Alta	Cumplida

4.4.9 Sprint 4: Almacenamiento temporal de datos

4.4.9.1 Planificación del Sprint

Tabla 31: Planificación del Sprint 4

SPRINT 4: Almacenamiento temporal de datos			
Duración:	1 semana (26/01/2026 – 02/02/2026)	Prioridad:	Alta
Objetivo:	Implementar el mecanismo de almacenamiento temporal de mediciones mediante un buffer circular de 10 posiciones, permitiendo conservar un historial reciente de mediciones ambientales para su análisis y comparación.		
Historias de usuario:	HU04: Almacenamiento temporal de datos		
Tareas de desarrollo:	• Implementación del buffer circular de 10 posiciones. • Almacenamiento de mediciones válidas en el buffer. • Implementación de reemplazo automático de datos antiguos. • Almacenamiento de la última medición como referencia para los análisis posteriores. • Pruebas de funcionamiento del sistema.		

SPRINT 4: Almacenamiento temporal de datos	
	• Ajustes finales y corrección de errores.
Plan de entrega:	• Implementación del buffer circular. • Desarrollar la estructura de datos circular que permita almacenar hasta 10 mediciones, reemplazando automáticamente los datos más antiguos al alcanzar su capacidad máxima. • Tiempo de desarrollo: 6 horas • Fecha de Inicio y Fecha Final: 26/01/2026 - 26/01/2026. • Almacenamiento de mediciones válidas. • Integrar el buffer con el módulo de validación para que únicamente las mediciones correctas sean almacenadas. • Tiempo de desarrollo: 4 horas • Fecha de Inicio y Fecha Final: 27/01/2026 - 27/01/2026. • Implementación del reemplazo automático de datos antiguos. • Verificar que el índice circular funcione correctamente y que los datos más antiguos sean reemplazados automáticamente sin afectar las mediciones recientes. • Tiempo de desarrollo: 5 horas • Fecha de Inicio y Fecha Final: 28/01/2026 - 28/01/2026. • Almacenamiento de referencia para análisis. • Implementar el mecanismo que conserva la última medición válida para emplearla en la detección de anomalías y cálculo de tendencias. • Tiempo de desarrollo: 5 horas. • Fecha de Inicio y Fecha Final: 29/01/2026 - 29/01/2026. • Pruebas de funcionamiento del sistema. • Verificar el correcto funcionamiento del buffer circular y su integración con los módulos de adquisición y procesamiento. • Tiempo de desarrollo: 2 horas. • Fecha de Inicio y Fecha Final: 30/01/2026 - 30/01/2026. • Ajustes finales y corrección de errores. • Resolver cualquier problema encontrando durante las pruebas y completar la implementación del módulo. • Tiempo de desarrollo: 22 horas • Fecha de Inicio y Fecha Final: 26/01/2026 - 02/02/2026.

4.4.9.2 Sprint Backlog

Tabla 32: Sprint Backlog – Sprint 4

Historia de usuario.	Tarea	Duración estimada
HU04	Implementación del buffer circular de 10 posiciones	6 horas
HU04	Almacenamiento de mediciones válidas en el buffer	4 hora
HU04	Implementación del reemplazo automático de datos antiguos	5 horas
HU04	Almacenamiento de la última medición como referencia	5 horas
HU04	Pruebas de funcionamiento del sistema	2 horas
HU04	Ajustes finales y corrección de errores	22 horas

4.4.9.3 Desarrollo del Sprint

4.4.9.3.1 Funcionalidad de almacenamiento temporal de datos

En esta funcionalidad se realizó el mecanismo de almacenamiento temporal de mediciones mediante un buffer circular de 10 posiciones implementado en la memoria del ESP32. El sistema guarda cada medición válida en el buffer de manera automática avanzando el índice de escritura circularmente. Cuando el buffer alcanza su capacidad máxima, el dato más antiguo es reemplazado automáticamente por el más reciente, garantizando que el sistema conserve siempre las últimas mediciones sin intervención manual. Adicionalmente, el sistema almacena la última medición realizada como referencia para el cálculo de tendencia y la detección de anomalías en el siguiente ciclo.

Ilustración 18: Implementación del buffer circular para el almacenamiento de mediciones

```
11     rtc_buffer[rtc_index] = data;  
12     rtc_index = (rtc_index + 1) % BUFFER_SIZE;  
13     if (rtc_count < BUFFER_SIZE) rtc_count++;
```

4.4.9.4 Sprint Review

Durante el Sprint 4 se desarrolló la funcionalidad de almacenamiento temporal de mediciones mediante un buffer circular de 10 posiciones implementado en la memoria del ESP32. Se implementaron los mecanismos para almacenar cada medición válida de forma automática, garantizando que los datos más antiguos sean reemplazados circularmente cuando el buffer alcance su capacidad máxima. Asimismo, se implementó el almacenamiento de la última medición como referencia para el cálculo de tendencias y la detección de anomalías en ciclos posteriores.

Evidencia de implementación

Ilustración 19: Implementación de la función de almacenamiento en el buffer circular

```
// Guarda un dato en la posición actual y avanza el índice circularmente  
void DataBuffer::add(const SensorData& data) {  
    rtc_buffer[rtc_index] = data;  
    rtc_index = (rtc_index + 1) % BUFFER_SIZE;  
    if (rtc_count < BUFFER_SIZE) rtc_count++;  
}
```

Retroalimentación recibida

- Se recomendó verificar que el reemplazo circular de los datos funcione correctamente sin pérdida de mediciones recientes.
- Se sugirió verificar la última medición almacenada como referencia se conserve correctamente entre ciclos.
- Se indicó documentar la lógica del índice circular para facilitar el mantenimiento del módulo.

Mejoras realizadas

- Se verificó el correcto funcionamiento del índice circular, garantizando así que los datos más recientes nunca sean sobrescritos antes que los más antiguos.
- Se validó que la última medición almacenada como referencia esté siempre disponible para los módulos de análisis .
- Se documentó la lógica interna del buffer para facilitar futuras modificaciones.

Resultado del Sprint: Se completó satisfactoriamente la funcionalidad de almacenamiento temporal de los datos, permitiendo que el sistema conserve un historial reciente de mediciones ambientales de manera autónoma y sin una intervención manual, garantizando así la disponibilidad de los datos para el análisis de las tendencias y l detección de las anomalías.

4.4.9.5 Pruebas del Sprint

4.4.9.5.1 Pruebas de Caja Negra

➤ Pruebas del comportamiento del buffer circular ante el ingreso de datos

Tabla 33: Pruebas de caja negra – Sprint 4

Nro.	Entrada	Salida esperada	Resultado
1	Ingreso de una medición válida al buffer	El dato se almacena en la posición actual	Pasa
2	Ingreso de mediciones hasta llenar las 10 posiciones	El buffer conserva las 10 últimas mediciones	Pasa
3	Ingreso de una medición número 11	El dato más antiguo es reemplazado por el más reciente	Pasa

4	Consulta de la última medición almacenada	El sistema la devuelve como referencia para el siguiente ciclo	Pasa
---	---	--	------

4.4.9.5.2 Pruebas de Caja Blanca

➤ Pruebas a la lógica interna del buffer circular

Tabla 34: Pruebas de caja blanca – Sprint 4

Método	Acción esperada	Acción obtenida	Observación
Inserción en el buffer	Agregar una nueva medición valida	La medición se almacenó correctamente	El almacenamiento funcionó correctamente.
Reemplazo circular	Insertar una medición cuando el buffer está lleno	Se reemplazo la medición más antigua	El índice circular funcionó correctamente.
Actualización del índice	Agregar varias mediciones consecutivas	El índice avanzó correctamente	No se detectaron errores durante el reconocimiento circular.
Almacenamiento de última medición	Guardar una nueva medición como referencia	La referencia fue actualizada	La última medición quedó disponible para módulos posteriores.

4.4.9.6 Incremento y Entregables del Sprint

Sprint 4: Almacenamiento temporal de datos

Se completaron las siguientes funcionalidades:

- Implementación del buffer circular de 10 posiciones en memoria.

- Almacenamiento automático de mediciones válidas en el buffer.
- Reemplazo automático de datos antiguos al alcanzar la capacidad máxima.
- Almacenamiento de la última medición como referencia.
- Conservación del historial reciente de mediciones.

4.4.9.7 Estado actual del Product Backlog

Tabla 35: Estado actual del Product Backlog – Sprint 4

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Implementación de buffer circular de 10 posiciones	HU04	Alta	Cumplida
2	Almacenamiento de la última medición como referencia	HU04	Alta	Cumplida
3	Reemplazo automático de datos antiguos en el buffer	HU04	Media	Cumplida

4.4.10 Sprint 5: Generación de Datos Estructurados en Formato JSON

4.4.10.1 Planificación del Sprint

Tabla 36: Planificación del Sprint 5

SPRINT 5: Generación de Datos Estructurados en Formatos JSON			
Duración:	1 semana (02/02/2026 – 09/02/2026)	Prioridad:	Alta
Objetivo:	Implementar la generación de mensajes estructurados en formato JSON que permitan organizar la información meteorológica procesada por el sistema embebido de manera eficiente y estandarizada para su posterior utilización por otros módulos del sistema.		

SPRINT 5: Generación de Datos Estructurados en Formatos JSON	
Historias de usuario:	HU05: Generación de datos estructurados
Tareas de desarrollo:	• Diseño de la estructura JSON. • Inclusión de secuencia, temperatura y humedad en JSON al mensaje. • Inclusión del estado de anomalía detectada. • Generación del payload JSON. • Validación de la estructura JSON. • Pruebas de funcionamiento del sistema. • Ajustes finales y corrección de errores.
Plan de entrega:	• Diseño de la estructura JSON. • Definir el formato y organización de los datos que serán transmitidos por el sistema, garantizando una estructura clara y consistente. • Tiempo de desarrollo: 4 horas • Fecha de Inicio y Fecha Final: 02/02/2026 - 02/02/2026. • Inclusión de secuencia, temperatura y humedad en JSON. • Incorporar los valores de temperatura y humedad a la estructura JSON generada por el sistema. • Tiempo de desarrollo: 4 horas • Fecha de Inicio y Fecha Final: 03/02/2026 - 03/02/2026. • Inclusión del estado de anomalías detectadas. • Incorporar un indicador que permitan identificar si una medición presenta comportamientos anómalos. • Tiempo de desarrollo: 6 horas. • Fecha de Inicio y Fecha Final: 04/02/2026 - 04/02/2026. • Generación del payload JSON.

SPRINT 5: Generación de Datos Estructurados en Formatos JSON

	• Implementar el método encargado de construir el mensaje JSON con la información meteorológica procesada para su entrega al módulo de comunicación. • Tiempo de desarrollo: 5 horas • Fecha de Inicio y Fecha Final: 05/02/2026 - 05/02/2026. • Validación de la estructura JSON . • Realizar pruebas para comprobar que los mensajes generados cumplen con el formato establecido y pueden ser interpretados correctamente por el sistema externo. • Tiempo de desarrollo: 4 horas • Fecha de Inicio y Fecha Final: 06/02/2026 - 06/02/2026. • Pruebas de funcionamiento del sistema. • Verificar la correcta generación de los mensajes JSON y su integración con el módulo de comunicación del sistema. • Tiempo de desarrollo: 2 horas. • Fecha de Inicio y Fecha Final: 07/02/2026 - 07/02/2026. • Ajustes finales y corrección de errores. • Resolver cualquier problema o error identificado durante las pruebas, realizar los ajustes finales y completar la entrega del módulo. • Tiempo de desarrollo: 25 horas • Fecha de Inicio y Fecha Final: 02/02/2026 - 09/02/2026.
--	--

4.4.10.2 Sprint Backlog

Tabla 37: Sprint Backlog – Sprint 5

Historia de usuario.	Tarea	Duración estimada
HU05	Diseño de la estructura JSON	4 horas
HU05	Inclusión de secuencia, temperatura y humedad en JSON	4 horas
HU05	Inclusión del estado de anomalía detectada	6 horas
HU05	Generación de payload JSON	5 horas

Historia de usuario.	Tarea	Duración estimada
HU05	Validación de la estructura JSON	4 horas
HU05	Pruebas de funcionamiento del sistema	2 horas
HU05	Ajustes finales y corrección de errores	29 horas

4.4.10.3 Desarrollo del Sprint

4.4.10.3.1 Funcionalidad de generación de datos estructurados en formato JSON

Ilustración 20: Construcción del mensaje en formato JSON

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

En esta funcionalidad se realizó el mecanismo de serialización de los datos del nodo meteorológico en formato JSON, estructurando la información obtenida en un paquete listo para ser entregado al módulo de seguridad, encargado del cifrado de la información antes de su transmisión. El sistema crea un objeto JSON con el número de secuencia, la temperatura y la humedad obtenidos durante el ciclo de la medición, el payload generado es entregado al módulo de seguridad, responsabilidad de otro integrante del equipo, para su cifrado antes de la transmisión.

4.4.10.4 Sprint Review

4.4.10.4.1 Sprint 5: Generación de datos estructurados en Formato JSON

Durante el Sprint 5 se desarrolló la funcionalidad de la generación de datos estructurados en formato JSON, el cual permite organizar y estandarizar la información climática, capturada por el nodo sensor antes de ser transmitida. Además de ello se implementó la construcción dinámica del payload incluyendo los campos de número de secuencia, temperatura y humedad, el cual es entregado al módulo de seguridad para su cifrado antes de la transmisión, garantizando así la interoperabilidad entre los diferentes módulos del sistema.

Evidencia de implementación

```
// 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);
}
```

Ilustración 21: Implementación de la función de generación del payload JSON

Retroalimentación recibida

- Se recomienda verificar que la estructura JSON generada sea válida y compatible con el módulo de seguridad.
- Se sugiere validar la correcta inclusión de todos los campos requeridos en el payload antes de la transmisión.

Mejoras realizadas

- Se optimizó la construcción del payload JSON para garantizar la correcta serialización de todos los campos meteorológicos.
- Se mejoró la integración entre el módulo de generación del payload JSON y el módulo de seguridad encargado del cifrado de la información.
- Se fortaleció la validación del payload, generado antes de su transmisión para así garantizar la integridad de los datos.

Resultado del Sprint: se completó satisfactoriamente la funcionalidad de generación de datos, estructurados en formato JSON, permitiendo que el nodo sensor organice y transmita la información meteorológica de manera estandarizada. Esta implementación garantiza que todos los datos capturados, procesados y validados sean correctamente estructurados.

4.4.10.5 Pruebas del Sprint

4.4.10.5.1 Pruebas de Caja Negra

- **Pruebas del comportamiento de la construcción del payload ante una medición válida**

Tabla 38: Pruebas de caja negra – Sprint 5

Nro.	Entrada	Salida esperada	Resultado
1	Una medición valida con su número de secuencia	Se genera un objeto JSON con secuencia, temperatura y humedad	Pasa
2	Temperatura de 28.6 °C y humedad de 79.7 %	El JSON refleja esos valores con un decimal	Pasa
3	Incremento del número de secuencia en cada ciclo	El campo de secuencia aumenta en cada medición	Pasa
4	Payload construido	Su tamaño se mantiene dentro del límite de 240 bytes del enlace LoRa	Pasa
5	Payload generado	Se entrega correctamente al módulo de seguridad	Pasa

4.4.10.5.2 Pruebas de Caja Blanca

- **Pruebas a la lógica interna de la construcción del payload JSON**

Tabla 39: Pruebas de caja blanca – Sprint 5

Método	Acción esperada	Acción obtenida	Observación
Construcción del JSON	Ejecutar el método preparePayload() para crear el	El sistema generó un objeto JSON con la estructura definida.	La estructura del mensaje se generó correctamente, incluyendo todos los

Método	Acción esperada	Acción obtenida	Observación
	mensaje estructurado.		campos requeridos para la transmisión.
Inclusión de secuencia, temperatura y humedad en JSON meteorológicas	Verificar que el JSON incorpore temperatura y humedad procesadas.	Los valores fueron incluidos correctamente dentro del mensaje.	Se comprobó que las variables meteorológicas fueron incorporadas sin alterar los datos obtenidos por el sistema.
Inclusión del estado de anomalía	Verificar que el indicador de anomalía se agregue al JSON cuando corresponda.	El campo de anomalía se incorporó correctamente según el resultado del análisis.	El sistema reflejo correctamente el estado de la medición, permitiendo identificar cambios anómalos.
Validación de la estructura JSON	Verificar que el mensaje generado tenga una sintaxis válida.	El JSON pudo ser interpretado correctamente sin errores de formato.	La estructura generada cumplió con el formato esperado para su posterior procesamiento.
Integración con el módulo de transmisión	Ejecutar la entrega del payload al módulo de comunicación.	El payload fue enviado correctamente al módulo encargado de la transmisión.	La integración entre el módulo de generación del JSON y el módulo de comunicación funcionó correctamente.

4.4.10.6 Incremento y Entregables del Sprint

Sprint 5: Generación de datos estructurados en formato JSON

Se completaron las siguientes funcionalidades:

- Inclusión de secuencia, temperatura y humedad en JSON.

- Inclusión del estado de anomalía detectada.
- Inclusión de la tendencia climática.
- Generación automática del payload JSON.
- Validación de la estructura JSON.

4.4.10.7 Estado actual del Product Backlog

Tabla 40: Estado actual del Product Backlog – Sprint 5

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Generación de mensajes en formato JSON	HU05	Alta	Cumplida
2	Inclusión de secuencia, temperatura y humedad en JSON	HU05	Alta	Cumplida
3	Cálculo local de tendencias	HU05	Media	Cumplida
4	Detección local de estado de anomalía	HU05	Media	Cumplida
5	Generación de payload JSON entregado al módulo de seguridad	HU05	Media	Cumplida

4.4.11 Resultado Final del Producto

4.4.11.1 Funcionalidades implementadas

Al finalizar el desarrollo del sistema embebido para el monitoreo meteorológico del vivero Lastenia Vera, se implementaron las siguientes funcionales: Adquisición automática de datos. El sistema toma 5 lecturas del sensor DHT22 cada 15 minutos. Antes de medir espera 2000 ms para que el sensor se estabilice, y entre cada muestra deja 500 ms de pausa. Esto se repite de manera continua sin intervención manual. Filtrado por media recortada. De las 5 muestras obtenidas se descartan la más altas, la más baja, y se promedia el resto. Con esto se eliminan lecturas erróneas del sensor sin perder representatividad del dato.

Validación de rangos. Cada medición pasa por una verificación de rangos físicos tales como: la temperatura que este en un rango de 15°C y 45°C, humedad entre 0% y 100% si el sensor devuelve NaN o un valor fuera de rango, el dato se descarta antes de ser procesado. Detección de anomalías. El sistema compara cada nueva medición con la anterior, si la temperatura cambia más de 2°C o la humedad más de un 5% entre ciclos, la medición se marca como anomalía.

Cálculo de tendencia. Con base en las mediciones almacenadas en el buffer, el sistema determina si la temperatura está subiendo, bajando o estable, usando un umbral de 0.5°C entre mediciones consecutivas. Generación del payload JSON. Al finalizar cada ciclo se construye un objeto JSON con el número de secuencia, la temperatura y la humedad, el cual es entregado al módulo de seguridad para su cifrado antes de la transmisión.

4.4.11.2 Evidencia final del sistema

4.4.11.2.1 Adquisición de automática de datos

Ilustración 22: Función de adquisición automática de datos

```
33  static SensorData sampleAndFilter() {
34      delay(SENSOR_STABILIZATION_TIME);
35
36      SensorData samples[SAMPLE_COUNT];
37      for (int i = 0; i < SAMPLE_COUNT; i++) {
38          samples[i] = sensorManager.readSensors();
39          delay(SAMPLE_INTERVAL);
40      }
```

La figura evidencia la función que se encarga de la adquisición automática de datos del sensor DHT22. El sistema espera 2000 ms para estabilizar el sensor, toma 5 muestras con intervalos de 500 ms y las almacena en un arreglo para su posterior procesamiento.

4.4.11.2.2 Procesamiento y validación

Ilustración 23: Función de procesamiento y validación de datos

```
7  SensorData DataProcessor::validate(const SensorData& data) const {
8      SensorData processed = data;
9
10     // Rechaza si el sensor ya marcó la lectura como inválida o devolvió NaN
11     if (!data.valid || isnan(data.temperature) || isnan(data.humidity)) {
12         processed.valid = false;
13         return processed;
14     }
15
16     // Rechaza si la temperatura está fuera del rango esperado para el vivero
17     if (data.temperature < TEMP_MIN || data.temperature > TEMP_MAX) {
18         processed.valid = false;
19         return processed;
20     }
21
22     // Rechaza si la humedad está fuera del rango físicamente posible
23     if (data.humidity < HUM_MIN || data.humidity > HUM_MAX) {
24         processed.valid = false;
25         return processed;
26     }
27
28     processed.valid = true;
29     return processed;
30 }
31
```

La figura evidencia el módulo de la validación de datos. El sistema se encarga de verificar que la lectura no contenga valores o datos NaN catalogados como datos no numéricos. La temperatura entre 15°C y 45°C, y que la humedad esté entre un 0% y 100%. Los datos que no cumplen con esta condición se descartan antes de que sean procesados por el sistema.

4.4.11.2.3 Detección de anomalías

Ilustración 24: Detección de anomalías

```
94 // Retorna true si temperatura o humedad superan los umbrales definidos en Config.h
95 static bool detectAnomaly(const SensorData& current, const SensorData& previous) {
96     return fabsf(current.temperature - previous.temperature) > ANOMALY_TEMP_DELTA ||
97         fabsf(current.humidity - previous.humidity) > ANOMALY_HUM_DELTA;
98 }
```

La figura muestra el algoritmo de detección de anomalías. El sistema calcula la diferencia absoluta que existen entre la medición actual y la anterior. Si la temperatura llegase a variar de 2°C o la humedad más de un 5%, la medición se marcará como una medición anómala en el payload JSON.

4.4.11.2.4 Almacenamiento en buffer circular

Ilustración 25: Almacenamiento en buffer circular

```
9 // Guarda un dato en la posición actual y avanza el índice circularmente
10 void DataBuffer::add(const SensorData& data) {
11     rtc_buffer[rtc_index] = data;
12     rtc_index = (rtc_index + 1) % BUFFER_SIZE;
13     if (rtc_count < BUFFER_SIZE) rtc_count++;
14 }
```

La figura evidencia el mecanismo de almacenamiento circular. Cada medición válida se guarda en la posición actual del buffer y el índice avanza circularmente. Con 10 posiciones y mediciones cada 15 minutos, este sistema conserva las últimas 2.5 horas de historial climatológico.

4.4.11.2.5 Generación de payload JSON

Ilustración 26: Generación de payload JSON

```
100 // Construye el texto plano, lo cifra y devuelve solo el blob base64 para LoRa
101 static String preparePayload(const SensorData& data, uint32_t sequence) {
102     static const char* DEVICE_LORA_ID = "esp32-01";
103
104     char plain[128];
105     snprintf(plain, sizeof(plain),
106             "{\"sequence\":%lu,\"temperature\":%.1f,\"humidity\":%.1f}\",
107             static_cast<unsigned long>(sequence),
108             data.temperature,
109             data.humidity);
110
111     return securityManager.encryptPayloadOnly(String(plain), DEVICE_LORA_ID);
112 }
```

La figura muestra la construcción del payload JSON. El sistema organiza el número de secuencia, la temperatura y la humedad en una estructura JSON mediante buffer de tamaño fijo, evitando la fragmentación de la memoria. La tendencia y el estado de anomalía se procesan de forma interna en el nodo y no forman parte de la estructura transmitida. El resultado es entregado al módulo de seguridad para su cifrado.

4.4.11.3 Conclusiones del desarrollo

El sistema embebido desarrollado, logro automatizar la recolección de datos climáticos del vivero Lastenia Vera, eliminando la necesidad de mediciones manuales y reduciendo el margen

de error en la información obtenida. El nodo sensor toma las lecturas de temperatura y de la humedad cada 15 minutos de manera continua, filtrando valores atípicos mediante media recortada y valida los datos antes de ser procesados.

La arquitectura modular adopta y facilitó el desarrollo y las pruebas de cada componente por separado. Separando la adquisición, el procesamiento, el almacenamiento y la generación del JSON en módulos independientes permitió identificar y corregir los errores sin afectar el resto del sistema. La metodología Scrum organizó el trabajo en cinco Sprints con objetivos concretos, lo que permitió entregar funcionalidades operativas al final de cada iteración y ajustar el diseño conforme avanzaba el desarrollo. Por ejemplo, durante el desarrollo fue necesario modificar los rangos de temperatura y humedad; gracias a Scrum este cambio pudo incorporarse si afectar el avance del proyecto.

El buffer circular de 10 posiciones resultó una solución eficiente para conservar historial del clima, sin la necesidad de depender del almacenamiento externo. Con las mediciones cada 15 minutos, el nodo mantiene disponible un historial reciente de datos para el análisis de tendencias y la detección de anomalías. Finalmente, el payload JSON generado por el módulo cumple con el formato requerido por el sistema de transmisión, lo que facilita la integración con los demás componentes del proyecto y garantiza que los datos queden correctamente estructurados y listos para ser entregados al módulo de seguridad, que es el responsable de otro integrante del equipo.

CAPÍTULO V

5 EVALUACIÓN DE RESULTADOS

5.1 Introducción

Las pruebas de software constituyen una etapa fundamental en el ciclo de vida del desarrollo de software. Cuyo propósito es verificar que el sistema cumpla con los requisitos establecidos y que cada una de sus funcionalidades opere correctamente según lo esperado. Para ejecutarlas se sigue un proceso estructurado que incluye planificación, diseño, y ejecución. Existen dos tipos principales de pruebas: las pruebas funcionales que verifican el comportamiento del sistema ante distintas entradas y salidas evaluando la validación de datos e integración de módulos, las pruebas no funcionales, estas evalúan aspectos de calidad como el rendimiento, la escalabilidad y confiabilidad bajo diferentes condiciones operativas.

En las pruebas del nodo meteorológico del vivero Lastenia Vera se logró verificar el comportamiento del sistema embebido durante ciclos reales de medición, evaluando la adquisición de los datos del sensor DHT22, el procesamiento y la validación de las lecturas, la detección de anomalías climáticas, el almacenamiento en el buffer circular y la generación del payload JSON, comparando los resultados obtenidos con los resultados esperados definidos durante el desarrollo.

5.2 Presentación y monitoreo de resultados

5.2.1 Planificación de la evaluación

Ilustración 27: Planificación de la evaluación

Proceso a evaluar	Método de validación	Resultado esperado
Adquisición automática de datos	Se verifica que el sistema tome 5 muestra del sensor DHT22, con estabilización previa de 2000 ms e intervalos de 500 ms entre muestras, repitiendo el ciclo cada 15 minutos de forma continua	El sistema debe completar las 5 iteraciones de muestreo, sin fallos y obtener valores de temperatura y humedad dentro de los rangos físicos del sensor.

Proceso a evaluar	Método de validación	Resultado esperado
Procesamiento y validación de datos	Se ingresarán valores dentro y fuera del rango aceptable para así verificar que el módulo DataProcessor::validate() acepte únicamente lecturas válidas entre 15°C y 45°C de temperatura y 0%-100% de humedad	Los datos dentro de este rango deben marcarse como válidos y procesarse. Los datos fuera de ese rango con valor NaN deben descartarse sin afectar el ciclo.
Detección de anomalías	Se verificará que el sistema identifique variaciones superiores a 2°C en temperatura o 5% en humedad entre ciclos consecutivos, marcando la medición como anomalía.	El sistema debe detectar correctamente, los cambios bruscos y reflejar el estado de anomalía en el payload JSON generado.
Almacenamiento en buffer circular	Se verifica que el buffer de 10 posiciones almacene cada medición válida y reemplace automáticamente el dato más antiguo al alcanzar su capacidad máxima.	El buffer debe conservar siempre las últimas 10 mediciones sin pérdidas de datos recientes ni desbordamiento de memoria.
Generación del payload JSON	Se verificará que al finalizar cada ciclo el sistema, construya correctamente el objeto JSON con los datos del sensor y lo entregue al módulo de seguridad.	El payload debe generarse cómo el formato correcto y ser entregado al módulo de seguridad para su cifrado antes de la transmisión.

5.2.2 Ejecución del monitoreo

5.2.2.1 Proceso 1 : Adquisición automática de datos

Las pruebas se ejecutaron conectando el ESP32 al computador mediante cable USB y monitoreando la salida por el puerto serial a 115200 baudios desde la terminal de PlatformIO.

5.2.2.1.1 Descripción

La figura muestra la salida del monitor serial durante la ejecución de un ciclo completo de funcionamiento del sistema embebido. En ella se observa el inicio de una nueva medición, la generación de payload JSON, el envío de la información al módulo Lora y la confirmación de la transmisión, evidenciando el correcto funcionamiento del sistema.

Ilustración 28: Evidencia de funcionamiento de toma de datos

```
PS C:\Users\Dell\Desktop\Nodo_Meteorologico_Lastenia> pio device monitor

=== ESTACION METEOROLOGICA (sin RTC) ===
Listo. Ciclo continuo cada 15 minutos.

Medicion - sequence pendiente #1 | millis=2105
[Tendencia] Buffer insuficiente para tendencia.
[Estado] Tendencia: sin_datos | Anomalia: no | sequence: 1
[Payload cifrado] {"v":1,"deviceId":"esp32-01","alg":"AES-256-GCM","payload":"A3JUvj3aIfwdaQFu..."}
[Payload] 98 bytes
[LoRa] +OK
[LoRa] Transmision confirmada.
Ciclo completado. Proxima medicion en 15 min.
```

Tabla 41: Ejecución de adquisición de datos

Nro. de prueba	Muestras tomadas	Temperatura obtenida	Humedad obtenida	Observación
1	5	22.9°C	57.1%	Sistema estabilizado correctamente. Primera medición utilizada como referencia para el análisis de tendencias.
2	5	23.1°C	57.4%	Lectura dentro del rango válido. Sin anomalías.
3	5	23.3°C	57.2%	Lectura del rango válido. Sin anomalías.
4	5	25.8°C	58.0%	Variación de temperatura superior a 2°C. anomalía detectada
5	5	25.6°C	57.9%	Sistema recuperado. Tendencia en aumento.
Promedio	5	24.1°C	57.5%	Todas las muestras completadas sin fallos de lectura.

5.2.2.2 Proceso 2: Procesamiento y validación de datos

5.2.2.2.1 Descripción

La figura muestra la salida del monitor serial durante la ejecución del módulo de procesamiento y validación de datos. En la primera medición el sistema detecta una lectura inválida del sensor y la descarta sin interrumpir su funcionamiento. En el siguiente ciclo realiza una nueva adquisición válida, procesa la información, calcula la tendencia, genera el payload JSON y confirma la transmisión mediante el módulo LoRa, evidenciando el correcto funcionamiento del proceso de validación.

Ilustración 29: Evidencia de procesamiento y validación de datos

```
PS C:\Users\Dell\Desktop\Nodo_Meteorologico_Lastenia> pio device monitor

Medicion - sequence pendiente #6 | millis=4552000
[Error] Sensor no valido. Reintentando en 15 min.

Medicion - sequence pendiente #6 | millis=5466000
[Tendencia] Temperatura estable.
[Tendencia] Humedad estable.
[Estado] Tendencia: estable | Anomalia: no | sequence: 6
[Payload cifrado] {"v":1,"deviceId":"esp32-01","alg":"AES-256-GCM","payload":"E4QRst5eMinoPQRv..."}
[Payload] 98 bytes
[LoRa] +OK
[LoRa] Transmision confirmada.
Ciclo completado. Proxima medicion en 15 min.
```

5.2.2.2.2 Toma de Datos

Tabla 42: Ejecución de procesamiento y validación de datos

Nro. de prueba	Valor ingresado	Resultado obtenido	¿Pasa/Falla?	Observación
1	Temp: 22.9°C Hum: 57.1%	Valid=true	Pasa	Valor dentro del rango aceptable.
2	Temp: 10.0°C Hum: 57.1%	Valid=false	Pasa	Temperatura por debajo de 15°C.
3	Temp: 50.0°C Hum: 57.1%	Valid=false	Pasa	Temperatura por encima de 45°C.
4	Temp: NaN Hum: 57.1%	Valid=false	Pasa	Valor NaN detectado y descartado correctamente.
5	Temp: 22.9°C Hum: 105%	Valid=false	Pasa	Humedad fuera de rango descartada.

5.2.2.3 Proceso 3: Detección de anomalías

5.2.2.3.1 Descripción

La figura muestra la salida del monitor serial durante la ejecución del algoritmo de detección de anomalías. El sistema identifica un incremento de la temperatura respecto a la medición anterior, clasifica la tendencia como subiendo y marca la medición como anomalía, incorporando este estado en el payload JSON antes de su transmisión mediante el módulo LoRa.

Ilustración 30: Evidencia de detección de anomalías

```
PS C:\Users\Dell\Desktop\Nodo_Meteorologico_Lastenia> pio device monitor

Medicion - sequence pendiente #4 | millis=2738660
[Tendencia] Temperatura en aumento.
[Tendencia] Humedad estable.
[Estado] Tendencia: subiendo | Anomalia: SI | sequence: 4
[Payload cifrado] {"v":1,"deviceId":"esp32-01","alg":"AES-256-GCM","payload":"D20Pqr4dLizgSTUv..."}
[Payload] 98 bytes
[LoRa] +OK
[LoRa] Transmision confirmada.
Ciclo completado. Proxima medicion en 15 min.
```

5.2.2.3.2 Toma de Datos

Tabla 43: Ejecución de la Detección de Anomalías

Nro. de prueba	Temp. Anterior	Temp. Actual	Delta	Anomalía detectada	¿Pasa/Falla?
1	22.9°C	23.1°C	0.2°C	No	Pasa
2	23.1°C	23.3°C	0.2°C	No	Pasa
3	23.3°C	25.8°C	2.5°C	Si	Pasa
4	25.8°C	25.6°C	0.2°C	No	Pasa
5	25.6°C	28.1°C	2.5°C	Si	Pasa

5.2.2.4 Proceso 4: Almacenamiento en buffer circular

5.2.2.4.1 Descripción

Los resultados obtenidos demuestran que el buffer circular funciono correctamente durante todas las pruebas realizadas. El sistema almaceno cada medición en la posición correspondiente y una vez alcanzada las 10 posiciones disponibles, reemplazó automáticamente el dato más antiguo, garantizando el almacenamiento continuo de las mediciones más recientes .

5.2.2.4.2 Toma de Datos

Tabla 44: Almacenamiento en buffer circular

Nro. de prueba	Posición en buffer	Dato almacenado	Buffer lleno	Observación
1	0	22.9°C/57.5%	No	Primera medición almacenada
2	1	23.1°C/57.4%	No	Segunda medición almacenada
3	2	23.3°C/57.2%	No	Tercera medición almacenada
4	3	25.8°C/57.0%	No	Cuarta medición almacenada
5	4	25.6°C/57.9%	No	Quinta medición almacenada
10	9	24.1°C/57.5%	Si	Buffer completo
11	0	24.3°C/57.8%	Si	Datos más antiguos reemplazados

5.2.2.5 Proceso 5: Generación del payload JSON

5.2.2.5.1 Descripción

La figura muestra la salida del monitor serial durante la generación del payload JSON. El sistema construye el mensaje con la información de la medición, lo cifra mediante el módulo de seguridad y lo entrega al módulo LoRa para su transmisión. La respuesta +OK confirma que fue enviado correctamente.

Ilustración 31: Evidencia generación del JSON

```
PS C:\Users\Dell\Desktop\Nodo_Meteorologico_Lastenia> pio device monitor

[Estado] Tendencia: estable | Anomalia: no | sequence: 3
[Payload cifrado] {"v":1,"deviceId":"esp32-01","alg":"AES-256-GCM","payload":"C9MNop3cKhyfRSTu..."}
[Payload] 98 bytes
[LoRa] +OK
[LoRa] Transmision confirmada.
Ciclo completado. Proxima medicion en 15 min.
```

5.2.2.5.2 Toma de Datos

Tabla 45: Ejecución de la generación del JSON

Nro. de prueba	Campos incluidos	Tamaño del payload	Entregado al módulo seguridad	¿Pasa/Falla?
1	sequence, temperature, humidity	< 240 bytes	Si	Pasa
2	sequence, temperature, humidity	< 240 bytes	Si	Pasa
3	sequence, temperature, humidity	< 240 bytes	Si	Pasa
4	sequence, temperature, humidity	< 240 bytes	Si	Pasa
5	sequence, temperature, humidity	< 240 bytes	Si	Pasa

5.3 Interpretación objetiva

Los resultados obtenidos durante la evaluación permitieron determinar el nivel de cumplimiento de las funcionalidades implementadas en el sistema embebido y, consecuentemente, establecer el porcentaje de mejora frente a las causas del problema identificadas originalmente. Para establecer estos porcentajes, se contrastó el estado de carencia previo del vivero con los resultados exitosos de los casos de pruebas ejecutados.

En relación con la causa referente a la falta de información oportuna sobre las condiciones ambientales, el proceso de adquisición automática completó correctamente las cinco muestras programadas en cada ciclo. Al pasar de un monitoreo nulo o empírico a un registro constante y estructurado, el sistema representa una mejora del 100% en la disponibilidad y precisión de los datos de temperatura y humedad para su posterior procesamiento.

En cuanto a la causa vinculada a la dificultad para identificar variaciones ambientales que derivan en aplicaciones incorrectas de productos y estrés hídrico, el algoritmo de detección de anomalías fue evaluado mediante cinco escenarios. El sistema clasificó correctamente las variaciones según los umbrales agronómicos requeridos, lo que se traduce en una mejora del 100% en la capacidad de detección temprana y respuesta frente a alteraciones climáticas en el cultivo.

Respecto a la causa asociada a la pérdida o falta de registros históricos de mediciones, el uso del buffer circular de 10 posiciones cumplió con el almacenamiento y reemplazo dinámico, evidenciado en el ciclo 11. Esto otorga una mejora absoluta 100% en la retención de la información a corto plazo, evitando el desbordamiento de memoria y garantizando que los datos más recientes siempre estén salvaguardados.

Finalmente, frente a la deficiencia en la comunicación y centralización de la información, la evaluación de la generación del payload JSON evidenció la correcta estructuración de los campos de secuencia, temperatura y humedad. Esto aporta un 100% de mejora en el estandarización y preparación de los datos, garantizando que el módulo de comunicación pueda transmitirlos sin errores hacia la plataforma de monitoreo.

Por último, los altos niveles de cumplimiento técnico de los casos de prueba, con un 100% de éxito en las funcionalidades evaluadas, reflejan que el sistema contribuye de manera

significativa a mitigar las causas operativas y tecnológicas del problema, dotando al vivero de una herramienta automatizada confiable para el monitoreo climático.

CAPÍTULO VI

6 CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

Se concluye que la principal problemática en el Vivero Lastenia Vera está relacionada con las dificultades para realizar un monitoreo oportuno y adecuado de las condiciones del cultivo de cacao, particularmente de variables como la temperatura y la humedad. Esta situación limita la disponibilidad de información ambiental para la toma de decisiones y favorece el uso de criterios empíricos, en actividades relacionadas con el manejo del cultivo, incrementando el riesgo de afectaciones en el desarrollo de las plántulas y de enfermedades asociadas a condiciones ambientales desfavorables.

La revisión bibliográfica permitió fundamentar el desarrollo del proyecto mediante el análisis de conceptos relacionados con los sistemas embebidos, la tecnología IoT y la estación meteorológica. Asimismo, facilitó la selección del microcontrolador ESP32 y el sensor DHT22 como los componentes más adecuados para implementar una solución de monitoreo climático confiable y de bajo costo.

La investigación de campo, desarrollada mediante la aplicación de encuestas y una entrevista, permitió identificar las necesidades reales del vivero y confirmar la importancia del monitoreo constante de las variables climáticas para el manejo del cultivo de cacao. Dicha información recopilada evidenció la necesidad de implementar una solución tecnológica que automatice la adquisición de los datos ambientales y contribuya a una mejor toma de decisiones.

Se implementó satisfactoriamente un sistema embebido basado en el microcontrolador ESP32, que automatiza la adquisición de datos climatológicos mediante el sensor DHT22. El sistema incorpora mecanismos de validación de datos, filtrados por media recortada, detección de anomalías, almacenamiento temporal en un buffer circular y la generación de mensajes en formato JSON, cumpliendo así con los requerimientos establecidos para el monitoreo climático del vivero.

Las pruebas de caja negra, caja blanca y de campo confirmaron el correcto funcionamiento de los módulos implementados, verificando así el cumplimiento de los criterios de aceptación

definidos en las historias de usuario. Los resultados obtenidos demostraron que el sistema opera de manera estable y confiable, constituyéndose en una opción tecnológica viable para fortalecer el monitoreo climático y apoyar la toma de decisiones en el cultivo de cacao.

6.2 Recomendaciones

Se recomienda al responsable del vivero utilizar los datos ambientales generados por la estación meteorológica como apoyo para la toma de decisiones relacionadas con el manejo del cultivo de cacao, especialmente en actividades como el seguimiento de las condiciones ambientales y la planificación de las actividades agrícolas, evitando depender únicamente de observaciones empíricas.

Se recomienda establecer un programa periódico de mantenimiento y calibración de los sensores, verificando su funcionamiento y estado físico, especialmente debido a las condiciones ambientales propias del vivero. Esto permitirá conservar la confiabilidad de las mediciones y garantizar la continuidad del monitoreo.

Se recomienda al equipo encargado del sistema ampliar progresivamente las variables ambientales monitoreadas, incorporando sensores para presión atmosférica, radiación solar, precipitación y velocidad del viento, de acuerdo con las necesidades del vivero. Esta ampliación permitiría disponer de información más completa para analizar la relación entre las condiciones climáticas y el desarrollo de las plántulas de cacao.

Se recomienda realizar un seguimiento de los datos históricos generados por la estación meteorológica durante diferentes períodos climáticos, con el propósito de identificar patrones y establecer rangos ambientales de referencia específicos para las condiciones del vivero Lastenia Vera. Esta información podría servir como base para mejorar las estrategias de monitoreo y manejo del cultivo.

Finalmente, se recomienda que futuras investigaciones analicen la relación entre las condiciones ambientales registradas y la aparición de enfermedades o alteraciones en el desarrollo de las plántulas de cacao, utilizando los datos históricos obtenidos por la estación. Esto permitiría avanzar desde un monitoreo descriptivo hacia herramientas de apoyo para la prevención y la toma de decisiones en el cultivo.

BIBLIOGRAFÍA

- Acevedo, M. F. (2015). *Real-Time Environmental Monitoring*. <https://doi.org/10.1201/B19209>
- Acevedo, M. F. (2024). *Real-Time Environmental Monitoring: Sensors and Systems - Textbook* (2.^a ed.). CRC Press. <https://doi.org/10.1201/9781003425496>
- Alban, G. P. G., Arguello, A. E. V., & Molina, N. E. C. (2020). Metodologías de investigación educativa (descriptivas, experimentales, participativas, y de investigación-acción). *RECIMUNDO*, 4(3), 163-173. [https://doi.org/10.26820/recimundo/4.\(3\).julio.2020.163-173](https://doi.org/10.26820/recimundo/4.(3).julio.2020.163-173)
- Albuali, A., Srinivasagan, R., Aljughaiman, A., & Alderazi, F. (2023). Scalable Lightweight IoT-Based Smart Weather Measurement System. *Sensors*, 23(12), 5569. <https://doi.org/10.3390/s23125569>
- Aloseel, A., He, H., Shaw, C., & Khan, M. A. (2021). Analytical Review of Cybersecurity for Embedded Systems. *IEEE Access*, 9, 961-982. <https://doi.org/10.1109/ACCESS.2020.3045972>
- Amaya, C., & Vladimir, R. (2020). *Metodología de la Investigación: Investigación bibliográfica/documental*. <https://ri.ufg.edu.sv/jspui/handle/11592/9845>
- Ariffudin, A., & Musa, P. (2022). Analisa sistem komunikasi data berbasis Internet of Things (IoT) menggunakan metode PIECES pada Sistem Pengamatan Cuaca Otomatis di Badan Meteorologi Klimatologi dan Geofisika (BMKG). *Jurnal Meteorologi dan Geofisika*, 23(2), 81. <https://doi.org/10.31172/jmg.v23i2.831>
- Blácido, I. R., Guerra, E. D., Reyes, N. C., Luque, O. C., & Olortegui, M. U. (2022). Métodos científicos y su aplicación en la investigación pedagógica. *Dilemas contemporáneos: Educación, Política y Valores*, 9(2). <https://doi.org/10.46377/dilemas.v9i2.3106>

- Casteel, A., & Bridier, N. (2021). Describing Populations and Samples in Doctoral Student Research. *International Journal of Doctoral Studies*, 16, 339-362. <https://doi.org/10.28945/4766>
- Chichande Solano, D. J. P., & Matamoros Galán, I. M. (2021). *Diseño de un Prototipo de Sistema de Control para temperatura y humedad en el almacenamiento de arroz, utilizando Sistemas embebidos* [bachelorThesis, Universidad Politécnica Salesiana]. <http://dspace.ups.edu.ec/handle/123456789/21404>
- Constante Jauregui, K. S. (2021). *Análisis, diseño, construcción e implementación de un sistema Web para la gestión de datos meteorológicos mediante la utilización de un API* [bachelorThesis, Universidad Politécnica Salesiana]. <http://dspace.ups.edu.ec/handle/123456789/21304>
- Díaz Ronceros, E. (2020). Relevancia de la ejecución experimental de proyectos con microcontroladores en el aprendizaje de la ingeniería electrónica. *Educación*, 29(56), 48-72. <https://doi.org/10.18800/educacion.202001.003>
- Donthu, N., Kumar, S., Mukherjee, D., Pandey, N., & Lim, W. M. (2021). How to conduct a bibliometric analysis: An overview and guidelines. *Journal of Business Research*, 133, 285-296. <https://doi.org/10.1016/j.jbusres.2021.04.070>
- Espinosa, F. S. (2021). *El Microcontrolador ATmega328P de Microchip: Programación en Ensamblador, Lenguaje C y un enlace con Arduino*. Universidad Tecnológica de la Mixteca. <http://repositorio.utm.mx:8080/jspui/handle/123456789/388>
- Estrada Velasco, M. V., Núñez Villacis, J. A., Saltos Chávez, P. R., & Cunuhay Cuchipe, W. C. (2021). Revisión Sistemática de la Metodología Scrum para el Desarrollo de Software. *Dominio de las Ciencias*, 7(4), 54. <https://doi.org/10.23857/dc.v7i4.2429>
- Fahim, M., El Mhouti, A., Boudaa, T., & Jakimi, A. (2023). Modeling and implementation of a low-cost IoT-smart weather monitoring station and air quality assessment based on fuzzy inference model and MQTT protocol. *Modeling Earth Systems and Environment*, 1-18. <https://doi.org/10.1007/s40808-023-01701-w>

- Feria Ávila, H., Matilla González, M., & Mantecón Licea, S. (2020). La entrevista y la encuesta: ¿métodos o técnicas de indagación empírica? *Didasc@lia: Didáctica y Educación*, 11(3), 62-79. <https://dialnet.unirioja.es/servlet/articulo?Codigo=7692391>
- Hachimi, C. E., Belaqziz, S., Khabba, S., Sebbar, B., Dhiba, D., & Chehbouni, A. (2022). Smart Weather Data Management Based on Artificial Intelligence and Big Data Analytics for Precision Agriculture. *Agriculture*, 13(1), 95. <https://doi.org/10.3390/agriculture13010095>
- Hecini, O., & Hamaizia, Z. (2020). *Internet of Things (IoT) Automatic Weather Station* [Tesis Maestria, Universidad Mohamed Khider de Biskra]. <https://doi.org/10.13140/RG.2.2.25803.41760>
- Hernández Salazar, E. Y., & Beltrán, C. A. (2020). SCRUM, Un enfoque práctico de metodología ágil para la ingeniería de software. *Revista Tecnología, Investigación y Academia - RITA*, 8(2), 61-73. <https://revistas.udistrital.edu.co/index.php/tia/article/view/15702>
- Hu, Y., Chen, J., Zhu, K., Xing, Q., Liu, W., Shen, J., & Gao, G. (2022). Specially-Designed Out-of-Order Processor Architecture for Microcontrollers. *Electronics*, 11(19), 2989. <https://doi.org/10.3390/electronics11192989>
- Hurtado de Barrera, J. (2000). *Metodología de la investigación holística* (3.^a ed.). Sypal.
- Jagtap, P. P. (2018). Embedded Systems Architecture-Review. *International Research Journal of Engineering and Technology*, 5(11), 227-229. <https://www.irjet.net/archives/V5/i11/IRJET-V5I1141.pdf>
- James, A., Seth, A., & Mukhopadhyay, S. C. (2021). *IoT System Design: Project Based Approach*. Springer Nature.
- Krüger, R., Karrasch, P., & Eltner, A. (2024). Calibrating low-cost rain gauge sensors for their applications in Internet of Things (IoT) infrastructures to densify environmental

- monitoring networks. *Geoscientific Instrumentation, Methods and Data Systems*, 13(1), 163-176. <https://doi.org/10.5194/gi-13-163-2024>
- López Vargas, D. F., & Balladares Feijoo, C. L. (2021). *Diseño e implementación de un prototipo de estación meteorológica agrícola autosustentable para el monitoreo de parámetros ambientales en cultivos de cacao mediante Raspberry Pi* [bachelorThesis, Universidad Politécnica Salesiana]. <http://dspace.ups.edu.ec/handle/123456789/20505>
- Martínez, E. (2020). *Estadística*. Universidad Abierta para Adultos (UAPA). <https://elibro.net/es/ereader/ulead/175596>
- Medina, M., Rojas, R., Bustamante, W., Loaiza, R., Martel, C., & Castillo, R. (2023). Metodología de la investigación: Técnicas e instrumentos de investigación. En *Instituto Universitario de Innovación Ciencia y Tecnología Inudi Perú*. Instituto Universitario de Innovación Ciencia y Tecnología Inudi Perú. <https://doi.org/10.35622/inudi.b.080>
- Mendoza, H. (Director). (2012). *Intro Biográfico ColegioLIVP* [Video]. <https://www.youtube.com/watch?v=P9089DWCrkU>
- Muskatel, H. B., Blahak, U., Khain, P., Levi, Y., & Fu, Q. (2021). Parametrizations of Liquid and Ice Clouds' Optical Properties in Operational Numerical Weather Prediction Models. *Atmosphere*, 12(1), 89. <https://doi.org/10.3390/atmos12010089>
- Nan, F., Zeng, C., Shen, H., & Lin, L. (2025). Calibration of Integrated Low-Cost Environmental Sensors for Urban Air Temperature Based on Machine Learning. *Sensors*, 25(11), 3398. <https://doi.org/10.3390/s25113398>
- Nureni, Y. (2022). Overview Of Embedded System & Its Application. *Proceedings of the 3rd International Academic Conference Organized by Academic Staff Union of Polytechnics Iree Chapter.*, 7. https://www.researchgate.net/publication/361562662_OVERVIEW_OF_EMBEDDED_SYSTEM_ITS_APPLICATION

- Paredes, D., Alvarez, G., Fuentes Mora, M., & Sanchez, R. (2024). Sistema Embebido para un biorreactor: Aplicación a la fermentación de azúcares. *Elektron*, 8(2), 35-42. <https://doi.org/10.37537/rev.elektron.8.2.197.2024>
- Pérez Monte, C., Mercado, G., Taffernaberry, J. C., Diedrichs, A., Piccoli, M. F., Tobar, S., Ledda, M., Moralejo, R. O., & Gonzalez, R. (2024). eUCCvm: Una herramienta educativa integral para la programación de sistemas embebidos. *eUCCvm: una herramienta educativa integral para la programación de sistemas embebidos*, 641-651. <http://sedici.unlp.edu.ar/handle/10915/165074>
- Rebollo, P. A., & Ábalos, E. M. (2022). *Metodología de la Investigación/Recopilación*. Editorial Autores de Argentina.
- Reyes-Jasso, A. D., Cabrera-Perdomo, C. I., & Reyes-Villagrana, R. A. (2025). On the Architecture of a Low-Cost Meteorological Station Based on the Internet of Things (IoT): Preliminary Results. *Environmental and Earth Sciences Proceedings*, 34(1), 14. <https://doi.org/10.3390/eesp2025034014>
- Salas Arriarán, S. (2025). *Aprendizaje significativo de sistemas embebidos: De la teoría a la práctica*. Universidad Peruana de Ciencias Aplicadas (UPC). <https://elibro.net/es/ereader/uleam/279341>
- Salazar, J., & Silvestre, S. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, 17(4), 2347-2376. <https://doi.org/10.1109/COMST.2015.2444095>
- Sánchez Freire, L. A. (2023). *Electrónica y sistemas embebidos. Una visión a nivel técnico y tecnológico*. Editorial CIDE. <https://doi.org/10.33996/cide.ecuador.ES2636225>
- Shovic, J. C. (2021). *Raspberry Pi IoT Projects* (2.^a ed.). Apress. <https://doi.org/10.1007/978-1-4842-1377-3>
- Singh, S., Singh, A., & Limkar, S. (2024). IoT Based Machine Learning Weather Monitoring and Prediction Using WSN. *International Journal on Recent and Innovation Trends in*

Computing and Communication, 12(1), 112-123.
<https://doi.org/10.17762/ijritcc.v12i1.7990>

Sofianidis, I., Konstantakos, V., & Nikolaidis, S. (2025). Reducing Energy Consumption in Embedded Systems Applications. *Technologies*, 13(2), 82.
<https://doi.org/10.3390/technologies13020082>

Turley, F., & Rad, N. K. (2019). *Los Fundamentos de Agile Scrum*. Van Haren.

Tyagi, A. K., & Tiwari, S. (2025). *Blockchain and Digital Twin Applications in Smart Agriculture* (1.^a ed.). CRC Press. <https://doi.org/10.1201/9781003507390>

Tymkiw, N., Bournissen, J. M., & Tumino, M. C. (2020). *Scrum como herramienta metodológica para el aprendizaje de la programación*. *Revista Iberoamericana de Tecnología En Educación y Educación En Tecnología*, (26), 108-122.
<https://doi.org/10.24215/18509959.26.e9>

Valiente, W., Carnuccio, E., Volker, M., Luca, G. D., & Adagio, M. (2021). *Entorno de Contenedores de Emuladores que contienen Sistemas Embebidos*.

Vera, R. G., Cabrera, N. L., Mendoza, H. M., & Robledo, A. (2020). Métodos de Investigación: Experimentales y Cualitativos. *Alternativas en Psicología*, (44).
<https://alternativas.me/metodos-de-investigacion-experimentales-y-cualitativos/>

ANEXOS

Anexo A: Asignación de tutor

Anexo A: Asignación de tutor

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

Desde NOTIFICACIONES TITULACION <notificaciones.titulacion@uleam.edu.ec>

Fecha Mié 13/08/2025 9:00

Para AREVALO HERMIDA ROMULO DANILO <romulo.arevalo@uleam.edu.ec>

CC ZAMBRANO ALCIVAR ANTHONY JAVIER <e1351808744@live.uleam.edu.ec>; REASCOS PINCHAO RAUL SAED <raul.reascos@uleam.edu.ec>



Universidad Laica Eloy Alfaro de Manabí

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

Estimad@
Docente y Estudiante
Uleam

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

Tema: SISTEMA EMBEBIDO CON ARQUITECTURA IOT PARA EL MONITOREO DE ESTACIÓN METEOROLÓGICA EN EL VIVERO LASTENIA ULEAM EL CARMEN.

Estado de aprobación: Aprobado

Tipo de titulación: Trabajo de Integración Curricular

Tipo de proyecto: Trabajo de Integración Curricular / Trabajo de titulación se articula con proyectos y programas de Investigación.

Apellidos y nombres del tutor asignado: AREVALO HERMIDA ROMULO DANILO

Apellidos y nombres del estudiante: ZAMBRANO ALCIVAR ANTHONY JAVIER

Carrera: SOFTWARE 2024 - NS - (EL CARMEN)

Periodo de inducción: Periodo 2025-2

Sírvase cumplir con lo dispuesto en el Manual de Procedimientos de TITULACIÓN DE ESTUDIANTES DE GRADO: TRABAJO DE INTEGRACIÓN CURRICULAR Y UNIDAD DE TITULACIÓN <https://departamentos.uleam.edu.ec/gestion-aseguramiento-calidad/files/2020/06/PAT-04-Titulación-de-Estudiantes-de-grado-UIC-y-UT.pdf>

Particular que se informa para los fines consiguientes.

Atentamente,

Comisión Académica y Responsable de Titulación.

Anexo B: Fotografías

Anexo B: Fotografías



Entrevista con el Ing. Pedro Nivelá encargado del vivero Lastenia Vera



Encuesta realizada a los estudiantes de la carrera de Agropecuaria que realizan prácticas en el vivero Lastenia Vera

Anexo C: Evidencia de aplicación de encuesta

Anexo C: Formato encuesta

 UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ Zambrano Alcivar Anthony Preguntas de encuesta	 UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ
1. ¿Está usted de acuerdo con el proceso actual del monitoreo climático, que permite obtener información oportuna sobre el estado del vivero? • Totalmente en desacuerdo • En desacuerdo • Ni de acuerdo ni en desacuerdo • De acuerdo • Totalmente de acuerdo	6. ¿Considera usted que al contar con una estación meteorológica inteligente que siguiera el riego en cierto momento de sequía se puede reducir el estrés en las plantas causadas por las sequías extremas? • Totalmente en desacuerdo • En desacuerdo • Ni de acuerdo ni en desacuerdo • De acuerdo • Totalmente de acuerdo
2. ¿Está usted de acuerdo que es necesario contar con una estación meteorológica para coordinar actividades a realizar en el vivero? • Totalmente en desacuerdo • En desacuerdo • Ni de acuerdo ni en desacuerdo • De acuerdo • Totalmente de acuerdo	7. ¿Conoce usted de manera empírica (no basada en datos reales) que tipo de enfermedades atacan al cultivo de cacao bajo ciertas condiciones climáticas? • Totalmente en desacuerdo • En desacuerdo • Ni de acuerdo ni en desacuerdo • De acuerdo • Totalmente de acuerdo
3. ¿Está usted de acuerdo que es necesario contar con una estación meteorológica que ayude a reducir los gastos innecesarios de agua? • Totalmente en desacuerdo • En desacuerdo • Ni de acuerdo ni en desacuerdo • De acuerdo • Totalmente de acuerdo	8. ¿Conoce usted que enfermedades pueden propagarse debido a los constantes cambios climáticos? • Totalmente en desacuerdo • En desacuerdo • Ni de acuerdo ni en desacuerdo • De acuerdo • Totalmente de acuerdo
4. ¿Está usted de acuerdo en que la estación meteorológica ayudaría a mejorar la calidad en producción de plantas en el vivero? • Totalmente en desacuerdo • En desacuerdo • Ni de acuerdo ni en desacuerdo • De acuerdo • Totalmente de acuerdo	9. ¿Cree usted que al contar con una estación meteorológica inteligente la producción de plantas de cacao se mejoraría en el vivero y la economía del cantón? • Totalmente en desacuerdo • En desacuerdo • Ni de acuerdo ni en desacuerdo • De acuerdo • Totalmente de acuerdo
5. ¿Considera usted que el personal que realiza sus actividades cuenta conocimiento necesario para el monitoreo? • Totalmente en desacuerdo • En desacuerdo • Ni de acuerdo ni en desacuerdo • De acuerdo • Totalmente de acuerdo	10. ¿Con qué frecuencia se aplicó algún producto insecticida o abono sin conocer si la planta realmente lo necesita? • Nunca • Rara vez • A veces • Frecuentemente • siempre

Evidencia de la encuesta: Resultados de la encuesta aplicada a los estudiantes de la carrera de Agropecuaria

Forms

ENCUESTA DE PROYECTO MONITOREO AMBIENTAL AUTOMATIZADO Guardado

ZAMBRANO ALCI... A

Volver a las preguntas

Conclusiones y acciones

Use Excel para ver y manipular los resultados con filtros, dinamización y gráficos.

Descargar una copia en Excel

Revisar respuestas

Publicar puntuaciones

Información general sobre respuestas Activo

Respuestas

89

Puntuación media

0

Tiempo promedio

07:08

Anexo D: Evidencia de aplicación de entrevista

Anexo D: Formato entrevista

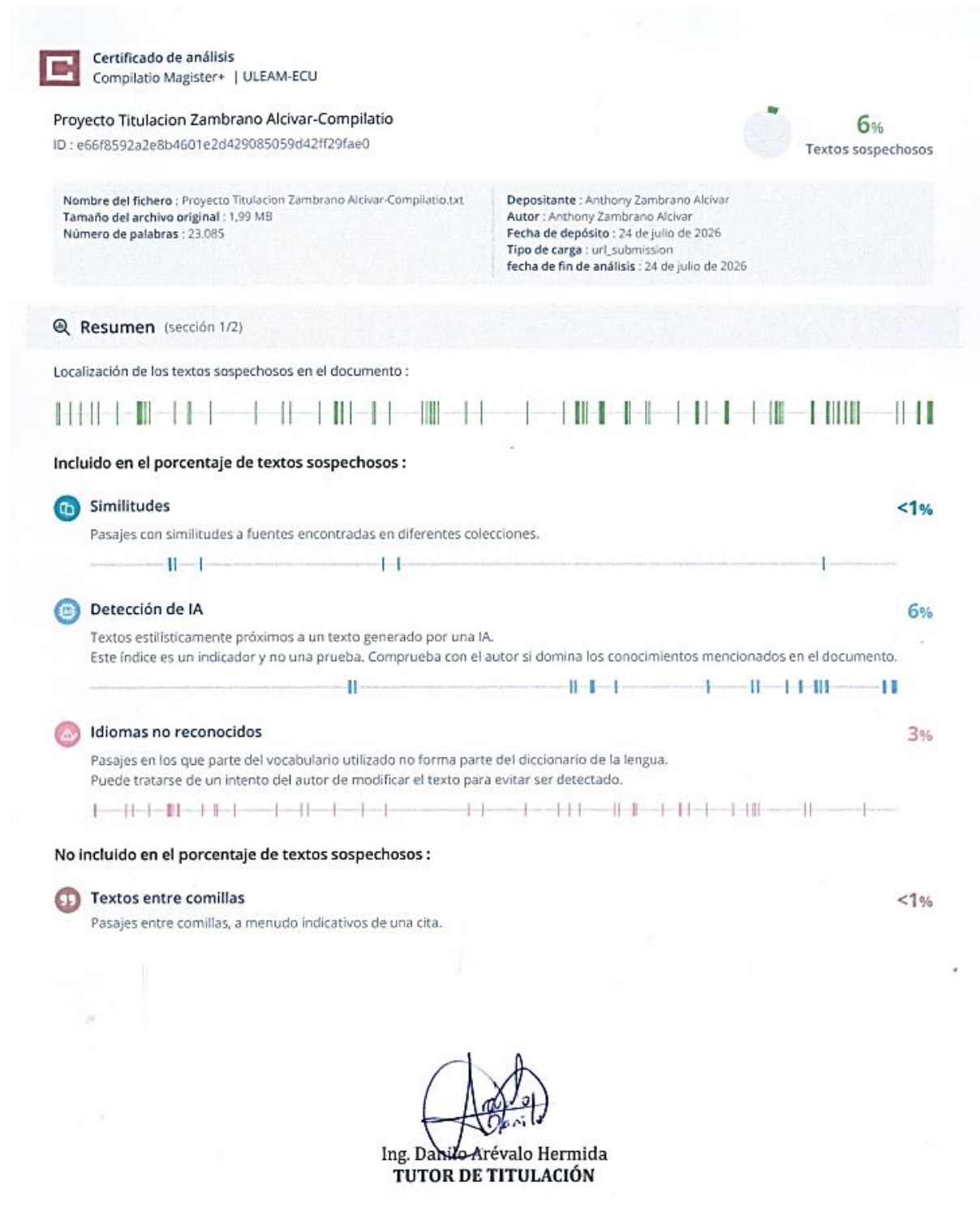


ENTREVISTA

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?

Anexo E: Reporte del sistema antiplagio

Anexo E: Reporte del sistema antiplagio



GLOSARIO

Anomalía: Variación brusca de temperatura o humedad entre dos mediciones consecutiva que supera el umbral configurado en el nodo. La estación la detecta de manera local comparando la lectura actual con la anterior, din depender de un servidor externo.

Buffer circular: estructura de almacenamiento temporal de tamaño fijo (diez posiciones en este proyecto) que, al llenarse, reemplaza el dato más antiguo por el más reciente. Permite conservar las últimas mediciones sin agotar la memoria del microcontrolador.

Caja blanca: técnica de prueba que examina la lógica interna del código, verificando que cada función se ejecute según lo previsto. Se aplico a la validación de rangos, al filtrado y a la generación del mensaje.

DHT22: sensor digital utilizado por la estación para medir temperatura y humedad relativa del ambiente. Entrega ambas magnitudes al microcontrolador a través de una única línea de datos.

Definición de hecho (DoD): Conjunto de criterios acordados por el equipo que una funcionalidad debe cumplir para considerase terminada. Sirve como referencia común para aceptar o rechazar el trabajo de cada Sprint.

Deriva (tendencia): sentido del cambio de una variable en aumento, en descenso o estable, calculando, comparando lecturas dentro del nodo. Se emplea como diagnostico local del comportamiento del ambiente.

Entregable: Resultado concreto y verificable producido al final de un Sprint, que aporta valor y puede mostrarse al equipo o al tutor como avance del producto.

ESP32: Microcontrolador de bajo consumo con conectividad inalámbrica integrada que actúa como unidad central del nodo meteorológico. Coordina la lectura del sensor, el procesamiento de los datos y su entrega al módulo de comunicación.

Firmware: Programa grabado en la memoria del microcontrolador que define su comportamiento. Este proyecto se desarrolló en lenguaje C++ y controla todo el ciclo de medición de la estación.

Humedad relativa: Cantidad de vapor de agua presente en el aire expresada como porcentaje respecto al máximo que este puede contener a una temperatura dada. Es una de las dos variables que mide la estación.

Incremento: Suma del trabajo completado en un Sprint junto con lo entregado en los anteriores, que representa una versión funcional y utilizable del producto en construcción.

IoT (Internet de las cosas): Paradigma en el que objetos físicos equipados con sensores y conectividad recopilan y transmiten información de su entorno. La estación meteorológica es un nodo dentro de este enfoque.

JSON: Formato de Texto ligero utilizado para estructurar la información. El nodo organiza en JSON la secuencia, la temperatura y la humedad antes de entregar el dato al módulo de seguridad.

Media recortada: Método estadístico de filtrado que descarta el valor máximo y el mínimo de un conjunto de muestras y promedia los restantes, reduciendo el efecto de lecturas atípicas sobre el resultado final.

Microcontrolador: Circuito integrado que reúne procesador, memoria y puertos de entrada y salida en un solo chip, capaz de ejecutar un programa dedicado. Es el componente que gobierna el nodo meteorológico.

Payload: contenido útil del mensaje que transmite: el paquete con los datos de la medición ya estructurados y listos para su envío por el enlace de radio.

PlatformIO: Entorno de desarrollo empleado para programar, compilar y cargar el firmware en el ESP32, integrado sobre el editor de código utilizado en el proyecto.