



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 INFORMÁTICO CON MQTT ORIENTADO A LA
SINCRONIZACIÓN DE DATOS EN EL VIVERO DE CACAO ULEAM
EL CARMEN

GUERRERO SALVATIERRA DAVID RICARDO
AUTOR

ARÉVALO HERMIDA RÓMULO DANILO
TUTOR

EL CARMEN, AGOSTO 2026



CERTIFICACIÓN DEL TUTOR

Ilustración 1 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 **GUERRERO SALVATIERRA DAVID RICARDO**, legalmente matriculado en la carrera de Software, período académico 2026-2027, cumpliendo el total de 384 horas, cuyo tema del proyecto es **"SISTEMA INFORMÁTICO CON MQTT ORIENTADO A LA SINCRONIZACIÓN DE DATOS EN EL VIVERO DE CACAO 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

TRIBUNAL DE SUSTENTACIÓN

Título de Trabajo de Titulación: SISTEMA INFORMÁTICO CON MQTT
ORIENTADO A LA SINCRONIZACIÓN DE DATOS EN EL VIVERO DE CACAO
ULEAM EL CARMEN

Modalidad: Proyecto integrador

Autor: Guerrero Salvatierra David Ricardo

Tutor: Ing. Árevalo Hermida Rómulo Danilo

Tribunal de sustentación:

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

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

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

Fecha de sustentación:

26 de Agosto de 2026

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



DECLARACIÓN DE AUTORÍA

La responsabilidad del contenido de este Trabajo de titulación, cuyo tema es: Sistema informático con MQTT orientado a la sincronización de datos en el vivero de cacao Uleam El Carmen corresponde exclusivamente a: Guerrero Salvatierra David Ricardo con C.I. 2300264625 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 "David Guerrero", is written over a horizontal line. The signature is stylized and fluid.

Guerrero Salvatierra David Ricardo

2300264625

DEDICATORIA

A mis padres, Patricia Salvatierra y Galo Guerrero. Ustedes son el origen de todo lo que hoy soy. Me sostuvieron cuando el camino se hizo largo ustedes me acompañaron en las noches de trabajo y en los días en que las fuerzas escaseaban que me enseñaron más con el ejemplo que con las palabras que el esfuerzo constante siempre encuentra su recompensa. Cada renuncia que hicieron para que yo pudiera estudiar está escrita en estas páginas. Este logro no es mío es nuestro.

A mis abuelos, Mariana Chávez y Ezio Salvatierra raíz y ejemplo de esta familia. Gracias por enseñarme con la sencillez de quien ha vivido mucho y con tanto que las cosas que verdaderamente valen se construyen con paciencia con honestidad y con trabajo. Llevo conmigo cada consejo suyo y yo espero que este trabajo sea motivo de orgullo para ustedes.

A Nahomy Guevara mi novia y apoyo constante. Fuiste quien estuvo cerca cuando el cansancio pesaba más que las ganas quien me devolvió el ánimo cuando lo había perdido y además quien creyó en este proyecto incluso en los días en que yo no lograba verle el final. Gracias por tu paciencia infinita por celebrar conmigo cada avance por pequeño que fuera y por no dejarme caer.

A mis mejores amigos Steve Lara, Paulette Enriquez y Adrián Becerra. A ustedes les debo mucho más de lo que cabe en una página. Gracias por el consejo sincero cuando lo necesitaba y por el regaño honesto cuando también lo necesitaba por las risas que aligeraron los días difíciles y por la calma que me dieron en los momentos de presión. Gracias por defenderme por cubrirme por permanecer a mi lado incluso cuando me equivoqué y por recordarme que un tropiezo no te define ni define a nadie. La verdadera amistad se mide en los malos momentos y ustedes nunca me dejaron solo en ninguno. Este trabajo también lleva algo de cada uno de ustedes.

David Ricardo Guerrero Salvatierra

AGRADECIMIENTO

A mi tío Antonio Salvatierra, por su apoyo gracias por aconsejarme con criterio y por contribuir su influencia está presente en cada paso que me trajo hasta aquí.

A la Doctora Debith Alcívar y a la Sra. Ximena Castañeda, por su orientación y su apoyo en momentos determinantes de mi formación. Gracias por tenderme la mano cuando el camino no estaba claro y por encaminarme hacia este objetivo sin su ayuda.

A mis amigas Andrea Parrales, Melany López, Emily Márquez, Ángela Cevallos, Dayana Muñoz, Melanie Burgos y Blanca Palma, por su compañía y su respaldo durante estos años de estudio, por el aliento en los momentos de duda y por hacer más llevadero el trayecto.

A mis jefes en BEGROUP, Melissa Mueses, por abrirme las puertas de la empresa y darme la oportunidad de ejercer profesionalmente aquello para lo que hoy me gradúo. Gracias por la confianza depositada en mí, por el aprendizaje que me ha dejado el trabajo diario y por permitirme crecer en el ámbito profesional mientras culminaba mi formación académica. De igual manera, agradezco al Ing. Marlon Bedón, gerente general, por la confianza y el respaldo brindados durante este tiempo.

A mi tutor, el Ingeniero Rómulo Danilo Arévalo Hermida por su acompañamiento académico, su exigencia y sus observaciones oportunas, que orientaron el desarrollo y la mejora continua de este proyecto.

A los ingenieros César Sinchiguano, Ángel Villarreal, Jefferson Arca, Saed Reascos y a la ingeniera Clara Pozo, por su apoyo y sus consejos en los momentos en que compaginar el trabajo con los estudios se volvió cuesta arriba. Gracias por la comprensión, la orientación oportuna y el ánimo que me brindaron cuando sentía que no podía continuar ya que su respaldo fue determinante para no abandonar este camino.

A la Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen, y a sus docentes, por la formación recibida a lo largo de la carrera.

David Ricardo Guerrero Salvatierra

ÍNDICE GENERAL

PORTADA.....	¡Error! Marcador no definido.
CERTIFICACIÓN DEL TUTOR.....	II
TRIBUNAL DE SUSTENTACIÓN.....	¡Error! Marcador no definido.
DECLARACIÓN DE AUTORÍA.....	IV
DEDICATORIA	V
AGRADECIMIENTO	VI
ÍNDICE GENERAL	VII
ÍNDICE DE TABLAS	XVI
ÍNDICE DE ILUSTRACIONES	XVIII
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	5

1.4.3	Estado actual del problema	5
1.5	Diagrama causa – efecto del problema	7
1.6	Objetivos	8
1.6.1	Objetivo general.....	8
1.6.2	Objetivos específicos	8
1.7	Justificación.....	8
1.8	Impactos esperados	9
1.8.1	Impacto tecnológico.....	9
1.8.2	Impacto social	9
1.8.3	Impacto ecológico	10
CAPÍTULO II		11
2	MARCO TEÓRICO.....	11
2.1	Antecedentes históricos.....	11
2.2	Antecedentes de investigaciones relacionadas al tema presentado	13
2.3	Definiciones conceptuales.....	14
2.3.1	Sistema informático con MQTT	14
2.3.1.1	Protocolo MQTT propósito, versiones y casos de uso en agro	15
2.3.1.2	Modelo publish/subscribe: tópicos, wildcards y estructura de mensajes ...	16
2.3.1.3	Bróker MQTT: funciones, despliegue y alta disponibilidad	16
2.3.1.4	Calidad de servicio (QoS), retención, sesiones y Last Will	17
2.3.1.5	Seguridad y control de acceso: TLS, autenticación y ACLs	18

2.3.1.6	Arquitectura IoT de la solución propuesta	19
2.3.1.7	Integración y persistencia: mapeo de mensajes a base de datos y esquema de datos	19
2.3.2	Sincronización de datos.	20
2.3.2.1	Conceptualización y alcance de la sincronización en sistemas distribuidos	21
2.3.2.2	Modos de sincronización: tiempo real, por lotes y near real-time.....	22
2.3.2.3	Consistencia y resolución de conflictos en registros concurrentes.....	22
2.3.2.4	Dimensiones e indicadores: disponibilidad, latencia, completitud y consistencia	23
2.3.2.5	Esquema offline-first y tolerancia a conectividad intermitente.....	24
2.3.2.6	Validación y control de calidad de datos en el ingreso	24
2.3.2.7	Seguridad y trazabilidad durante la sincronización.....	25
2.3.2.8	Patrones event-driven y buenas prácticas en contextos agroambientales...	26
2.3.3	Metodología de desarrollo, Prototipos.....	27
2.4	Conclusiones del marco teórico	28
CAPÍTULO III.....		29
3.	MARCO INVESTIGATIVO	29
3.1	Introducción	29
3.2	Tipos de investigación.....	29
3.2.1	Investigación de campo.....	29
3.2.2	Investigación descriptiva	30
3.2.3	Investigación cuantitativa	31

3.3	Técnicas de recolección de datos	31
3.3.1	Observación directa	31
3.3.2	Encuesta	32
3.3.3	Entrevista	32
3.4	Fuentes de información de datos	33
3.4.1	Fuentes primarias	33
3.4.1.1	Encuestas	34
3.4.1.2	Entrevista	34
3.5	Estrategia operacional para la recolección de datos.....	35
3.5.1	Población.....	35
3.5.2	Muestra	35
3.5.3	Análisis de las herramientas de recolección de datos a utilizar	36
3.5.3.1	Encuesta.....	36
3.5.3.2	Entrevista	37
3.5.3.3	Estructura de los instrumentos de recolección de datos aplicados	37
3.5.4	Plan de recolección de datos	38
3.6	Análisis y presentación de resultados.....	39
3.6.1	Presentación y descripción de los resultados obtenidos	39
3.6.1.1	Análisis de entrevista realizada a Coordinador de Vivero Lastenia Vera ..	43
3.6.2	Informe final del análisis de los datos.....	46
CAPÍTULO IV.....		48

4. MARCO PROPOSITIVO	48
4.1 Introducción	48
4.2 Descripción de la propuesta	48
4.3 Determinación de recursos	49
4.3.1 Humanos	49
4.3.2 Tecnológicos	50
4.3.3 Económicos	50
4.4 Desarrollo del sistema bajo la metodología Scrum	51
4.4.1 Descripción del Producto	52
4.4.1.1 Propósito del Producto	52
4.4.1.2 Funcionalidades Clave	52
4.4.1.3 Usuarios Objetivo:	53
4.4.1.3 Criterios de Éxito del Producto	53
4.4.2 Requerimientos del sistema	54
4.4.2.1 Historias de Usuario	54
4.4.2.2 Requerimientos funcionales	56
4.4.2.3 Requerimientos no funcionales	57
4.4.3 Diseño del Sistema	58
4.4.3.1 Casos de Uso	58
4.4.3.2 Diagramas de Secuencia	59
4.4.3.3 Diagramas de Estado	61

4.4.3.4	Diagrama de Base de Datos.....	62
Arquitectura del sistema		62
4.4.3.5	Arquitectura general	62
4.4.3.6	Tecnologías utilizadas	63
4.4.4	Roles Scrum.....	64
4.4.5	Definición de Hecho (DoD).....	65
4.4.5.1	Criterios Generales	65
4.4.5.2	Criterios Específicos del Proyecto.....	65
4.4.6	Product Backlog.....	66
4.4.7	Sprint 1: Infraestructura, base de datos y API	67
4.4.7.1	Planificación del Sprint	67
4.4.7.2	Sprint Backlog Sprint 1	68
4.4.7.3	Desarrollo del Sprint 1.....	68
4.4.7.4	Sprint Review Sprint 1	70
4.4.7.5	Pruebas del Sprint 1, Caja Blanca	72
4.4.7.6	Incremento y entregable del Sprint 1.....	72
4.4.7.7	Estado actual del Produc Backlog	73
4.4.8	Sprint 2: Ingesta MQTT, validación y auditoría.....	75
4.4.8.1	Planificación del Sprint 2	75
4.4.8.2	Sprint Backlog Sprint 2	76
4.4.8.3	Desarrollo del Sprint 2.....	76

4.4.8.4	Sprint Review Sprint 2	79
4.4.8.5	Pruebas del Sprint 2, Caja Blanca	80
4.4.8.6	Incremento y Entregable del Sprint 2	81
4.4.8.7	Estado actual del Product Backlog	82
4.4.9	Sprint 3: Seguridad – Cifrado extremo a extremo y transporte seguro.....	83
4.4.9.1	Planificación del Sprint 3	83
4.4.9.2	Sprint Backlog Sprint 3	84
4.4.9.3	Desarrollo del Sprint 3.....	84
4.4.9.4	Sprint Review Sprint 3	87
4.4.9.5	Pruebas del Sprint 3, Caja Blanca	88
4.4.9.6	Incremento y Entregable del Sprint 3	89
4.4.9.7	Estado del Sprint Backlog 3	90
4.4.10	Sprint 4: Pruebas automatizadas y documentación.....	91
4.4.10.1	Planificación del Sprint	91
4.4.10.2	Sprint Backlog Sprint 4	91
4.4.10.3	Desarrollo del Sprint 4.....	92
4.4.10.4	Sprint Review – Sprint 4	95
4.4.10.5	Pruebas del Sprint, Caja blanca	96
4.4.10.6	Incremento y Entregable del Sprint.....	96
4.4.10.7	Estado del Sprint Backlog 4	97
4.4.11	Sprint 5: Integración extremo a extremo y despliegue.....	98

4.4.11.1	Planificación del Sprint 5	98
4.4.11.2	Sprint Backlog Sprint 5S	99
4.4.11.3	Desarrollo del Sprint 5.....	99
4.4.11.4	Sprint Review Sprint 5	103
4.4.11.5	Pruebas del Sprint 5	105
4.4.11.6	Incremento y Entregables del Sprint 5	106
4.4.11.7	Estado actual del Product Backlog 5	106
4.4.12	Resultado Final del Producto	108
4.4.12.1	Funcionalidades implementadas.....	108
4.4.12.2	Evidencia final del sistema.....	109
4.4.12.3	Conclusiones del desarrollo.....	109
CAPÍTULO V		111
5.	EVALUACIÓN DE RESULTADOS	111
5.1	Introducción	111
5.2	Presentación y monitoreo de resultados	112
5.2.1	Planificación de la evaluación	112
5.2.2	Ejecución del monitoreo	113
5.2.2.1	Proceso 1 Recepción y almacenamiento de telemetría.....	113
5.2.2.2	Proceso 2 Deduplicación de lecturas	115
5.2.2.3	Proceso 3 Detección de manipulación de datos.....	117
5.3	Interpretación objetiva.....	119

CAPÍTULO VI.....	121
6. CONCLUSIONES Y RECOMENDACIONES	121
6.1 Conclusiones	121
6.2 Recomendaciones.....	123
BIBLIOGRAFÍA	125
ANEXOS	128
GLOSARIO	142

ÍNDICE DE TABLAS

Tabla 1 Plan de recolección de datos	38
Tabla 2 Resultados de la encuesta dirigida a los estudiantes de Ingeniería en Agropecuaria .	39
Tabla 3 Análisis de la entrevista realizada al coordinador del Vivero Lastenia Vera	44
Tabla 4 Recursos humanos del proyecto	49
Tabla 5 Recursos Tecnológicos	50
Tabla 6 Recursos Económicos	50
Tabla 7 Usuarios Objetivos.....	53
Tabla 8 Historia de usuario HU01: Autenticación y navegación básica	54
Tabla 9 Historia de usuario HU02: Registro de mediciones en campo	54
Tabla 10 . Historia de usuario HU03: Sincronización y almacenamiento centralizado	55
Tabla 11 Historia de usuario HU04: Visualización de históricos y monitoreo	55
Tabla 12 Historia de usuario HU05: Gestión de alertas por condiciones	56
Tabla 13 Tecnologías usadas	63
Tabla 14 Roles de Scrum	64
Tabla 15 Estado del Product Backlog.....	66
Tabla 16 Infraestructura del backend, base de datos y API de consulta.....	67
Tabla 17 Sprint Backlog del Sprint 1	68
Tabla 18 Pruebas de caja blanca del Sprint 1 – Infraestructura, base de datos y API	72
Tabla 19 Estado del Product Backlog al cierre del Sprint 1	73
Tabla 20 Recepción de telemetría MQTT validación deduplicacion y auditoria	75

Tabla 21 Sprint Backlog del Sprint 2.....	76
Tabla 22 Estado del Product Backlog al cierre del Sprint 2	82
Tabla 23 Cifrado de extremo a extremo, detección de manipulación y transporte seguro	83
Tabla 24 Estado del Product Backlog al cierre del Sprint 3	84
Tabla 25 Estado del Product Backlog al cierre del Sprint 3	90
Tabla 26 Cifrado de extremo a extremo, detección de manipulación y transporte seguro	91
Tabla 27 Estado del Product Backlog al cierre del Sprint 4	91
Tabla 28 Pruebas de caja blanca del Sprint 4 – Pruebas automatizadas y documentación	96
Tabla 29 Estado del Product Backlog al cierre del Sprint 4	97
Tabla 30 Integración con el hardware, seguridad de acceso y despliegue en producción.....	98
Tabla 31 Sprint Backlog del Sprint 5.....	99
Tabla 32 Ingestión segura por telemetría por HTTP	105
Tabla 33 Procesamiento de la ingestión y control de acceso.....	105
Tabla 34 Estado del Product Backlog al cierre del Sprint 5	106
Tabla 35 Funcionalidades implementadas	108
Tabla 36 Planificación de la evaluación del componente de sincronización.....	112
Tabla 37 Resultados del Proceso 1 – Recepción y almacenamiento de telemetría	114
Tabla 38 Resultados del Proceso 2 - Deduplicación de lecturas	116
Tabla 39 Resultados del Proceso 3 - Detección de manipulación de datos	118
Tabla 40 Mitigación de las causas del problema	119

ÍNDICE DE ILUSTRACIONES

Ilustración 1 Certificación del tutor	II
Ilustración 2 Árbol de problemas sobre la inconsistencia de registros en el vivero	7
Ilustración 3 Distribución de respuestas de la Pregunta 1 – Oportunidad de la información del monitoreo actual.....	39
Ilustración 4 Distribución de respuestas de la Pregunta 8 – Acceso a información en tiempo real para decisiones fitosanitarias	40
Ilustración 5 Distribución de respuestas de la Pregunta 12 – Riesgo de errores por registro manual.....	40
Ilustración 6 Distribución de respuestas de la Pregunta 16 – Tiempo dedicado a la transcripción manual.....	41
Ilustración 7 Distribución de respuestas de la Pregunta 17 Efectividad del canal de comunicación actual.....	42
Ilustración 8 Distribución de respuestas de la Pregunta 18 – Efecto de la falta de un repositorio centralizado	43
Ilustración 9 Distribución de respuestas de la Pregunta 22 – Eficiencia y eficacia del registro manual de datos.....	43
Ilustración 10 Ingesta de datos.....	58
Ilustración 11 Validación de datos.....	58
Ilustración 12 Consulta de telemetría	59
Ilustración 13 Diagrama de secuencia del proceso de captura y sincronización de una lectura	59
Ilustración 14 Diagrama de secuencia del procesamiento de un lote de telemetría.....	60

Ilustración 15 Diagrama de secuencia de la consulta de lecturas	60
Ilustración 16 Diagrama de estados del ciclo de vida de un lote de telemetría	61
Ilustración 17 Diagrama de estados de un dispositivo registrado en el servidor	61
Ilustración 18 Diagrama de la base de datos.....	62
Ilustración 19 Arquitectura del programa	62
Ilustración 20 Servicios del backend en ejecución contenedores Docker	69
Ilustración 21 Respuesta de la API con el contrato de datos	69
Ilustración 22 Servicios del backend en ejecución (contenedores Docker).....	71
Ilustración 23 Respuesta de la API con el contrato de datos	71
Ilustración 24 Ingesta correcta de un lote de telemetría	77
Ilustración 25 Registro de la auditoria de los lotes procesados	77
Ilustración 26 Ingesta correcta de un lote de telemetría	79
Ilustración 27 Rechazo de un lote duplicado	79
Ilustración 28 Registro de auditoría de los lotes procesados	79
Ilustración 29 Pruebas de caja blanca del Sprint 2 – Ingesta MQTT, validación y auditoría	¡Error! Marcador no definido.
Ilustración 30 Descifrado correcto de los lotes cifrados	84
Ilustración 31 Rechazo de un lote manipulado	85
Ilustración 32 Auditoria del intento de manipulación.....	85
Ilustración 33 Conexión al bróker MQTT mediante TLS 1.3	85
Ilustración 34 Descifrado correcto de los lotes cifrados	87
Ilustración 35 Rechazo de un lote manipulado	87

Ilustración 36 Auditoria del intento de manipulación.....	87
Ilustración 37 Conexión al bróker MQTT mediante TLS 1.3	87
Ilustración 38 Pruebas de caja blanca del Sprint 3 – Cifrado extremo a extremo y transporte seguro	¡Error! Marcador no definido.
Ilustración 39 Ejecución de la batería de pruebas automatizadas.....	92
Ilustración 40 Ejecución de la batería de pruebas automatizadas.....	93
Ilustración 41 Ejecución de la batería de pruebas automatizadas.....	93
Ilustración 42 Ejecución de la batería de pruebas automatizadas (52 pruebas aprobadas)	95
Ilustración 43 Servidor accesible desde Internet mediante dominio propio	99
Ilustración 44 Control de acceso a la API mediante llave de aplicación sin llave de acceso	100
Ilustración 45 Recepción y envío del dato cifrado por parte del gateway	100
Ilustración 46 dato descifrado y almacenado.....	101
Ilustración 47 Recepción y reenvío del dato cifrado por parte del gateway	103
Ilustración 48 El mismo dato ya descifrado y almacenado.....	103
Ilustración 49 Dato ingresado y descartado por intento simulado de inserción de datos ajeno a la estación.....	109
Ilustración 50 Dato ingresado y aceptado publicado por la estación y el gateway.....	109
Ilustración 51 Publicación de un lote cifrado de telemetría mediante MQTT sobre TLS	113
Ilustración 52 Lecturas almacenadas en la base de datos tras la ingesta	114
Ilustración 53 Registro de auditoría de los lotes procesados por ambas vías de ingesta	114
Ilustración 54 Envío de un lote cifrado manipulado para la prueba de detección de integridad	116
Ilustración 55 Verificación de ausencia de registros duplicados en la base de datos	116

Ilustración 56 Auditoría del lote original y de su reenvío	116
Ilustración 57 Envío de un lote cifrado manipulado para la prueba de detección de integridad	118
Ilustración 58 Registro de auditoría de los intentos de manipulación detectados	118
Ilustración 59 Verificación de que los lotes manipulados no fueron almacenados en la base de datos	118
Ilustración 60 Creación del servidor	129
Ilustración 61 Cantidad de estudiantes entrevistadas y tiempo promedio	130
Ilustración 62 Encuesta realizada a los estudiantes de agropecuaria	131

ÍNDICE DE ANEXOS

Anexo A: Asignación de tutor	128
Anexo B Fotografías	129
Anexo C Evidencia de aplicación de encuestas y entrevistas.....	130
Anexo D Estructura de la entrevista aplicada	132
Anexo E Estructura de la encuesta aplicada	133
Anexo F Certificado Sistema Antiplagio	141

RESUMEN

El presente proyecto tuvo como objetivo general implementar un sistema de sincronización eficiente de datos distribuidos utilizando el protocolo MQTT, para mejorar la confiabilidad operativa del monitoreo automatizado en el vivero de cacao de la ULEAM Extensión El Carmen. La problemática identificada se radicaba en que el registro de las variables ambientales se realizaba de forma manual en cuadernos y hojas de cálculo individuales lo que ocasionaba demoras en la consolidación de la información, errores de transcripción, ausencia de un repositorio centralizado y falta de garantías sobre la integridad de los datos, afectando la planificación y evaluación de los procesos formativos.

La población estuvo conformada por 89 personas vinculadas a los procesos operativos y académicos del vivero, de la cual se obtuvo una respuesta en su totalidad de 89 encuestas validas lo que otorga un carácter censal al levantamiento. Las técnicas de recolección empleadas fueron la observación directa, una encuesta estructurada de treinta preguntas dirigida a los estudiantes y una entrevista que esta semiestructurada de trece preguntas aplicada al coordinador técnico del vivero.

La propuesta consistió en un sistema informático con arquitectura IoT que recibe la telemetría por dos vías convergentes: el protocolo MQTT y un endpoint HTTP autenticado. La información se descifra mediante cifrado autenticado AES-256-GCM, se valida, se deduplica, se audita y se almacena en una base de datos centralizada, quedando disponible para su consulta a través de una API REST.

Los resultados de las pruebas evidenciaron que la totalidad de las lecturas enviadas se almacenó sin pérdida, con un tiempo promedio de disponibilidad de 0,232 segundos que la deduplicación evitó el 100 % de los registros repetidos ante reenvíos; y que el 100 % de los lotes manipulados fue detectado y rechazado sin ingresar a la base de datos.

Palabras clave: MQTT, sincronización de datos, cifrado de extremo a extremo, monitoreo ambiental, vivero de cacao.

ABSTRACT

The general objective of this project was to implement an efficient distributed data synchronization system using the MQTT protocol to improve the operational reliability of automated monitoring at the ULEAM El Carmen Extension cocoa nursery. The identified problem stemmed from the fact that environmental variables were recorded manually in notebooks and individual spreadsheets; this caused delays in data consolidation, transcription errors, the absence of a centralized repository, and a lack of guarantees regarding data integrity, thereby affecting the planning and evaluation of educational processes.

The population was made up of 89 people connected to the operational and academic processes of the nursery, from which a total response of 89 valid surveys was obtained, giving the survey a census-like character. The data collection techniques used were direct observation, a structured survey of thirteen questions directed at the students, and a semi-structured interview of thirteen questions applied to the nursery's technical coordinator.

The proposal involved an IoT-architecture computer system that receives telemetry via two converging channels—the MQTT protocol and an authenticated HTTP endpoint. It decrypts the data using AES-256-GCM authenticated encryption, validates, deduplicates, audits, and stores it in a centralized database, making it available for querying via a REST API.

Test results demonstrated that all transmitted readings were stored without loss, with an average availability time of 0.232 seconds; deduplication prevented 100% of duplicate records resulting from retransmissions; and 100% of manipulated batches were detected and rejected without entering the database.

Keywords: MQTT, data synchronization, end-to-end encryption, environmental monitoring, cocoa nursery.

CAPÍTULO I

1 INTRODUCCIÓN

1.1 Introducción

Para la planificación, ejecución y evaluación de las actividades en viveros universitarios, es imprescindible manejar la información de manera confiable y oportuna. En el contexto del cantón El Carmen en la Universidad Laica Eloy Alfaro de Manabí (ULEAM), los métodos de gestión que se emplean dentro del vivero de cacao, se llevan a cabo principalmente de forma manual en un soporte disperso, lo cual, ha causado retrasos en la consolidación, omisión y duplicación, que complican el proceso de intervención y coordinación entre los estudiantes y el personal técnico. Dichas circunstancias han impedido un correcto monitoreo de los datos, provocando falencias en la toma de decisiones y las oportunidades de evaluar los resultados de forma constante.

Por ende, se considera necesario que la institución cuente con acceso a datos estandarizados y accesibles que respalden la labor operativa y académica. Debido a su modelo de publicación/suscripción y a que es ligero, MQTT hace posible el transporte eficaz de mensajes desde dispositivos o formularios móviles hasta un repositorio central, aun con una conectividad limitada. Esta arquitectura proyectada incluye un bróker MQTT, clientes que publican y suscriben, una estructura de tópicos con niveles de calidad de servicio apropiados y una base de datos incorporada a un panel de consulta para su análisis y visualización.

Las variables de investigación se definieron de la siguiente manera. La variable independiente fue el sistema informático basado en MQTT, entendido como un conjunto coordinado de procedimientos y componentes tecnológicos destinados a capturar, validar, transportar y almacenar datos mediante este protocolo, con el objetivo de estandarizar el flujo de información. La variable dependiente fue la sincronización de datos del vivero, definida como el nivel en que los registros se mantienen actualizados, accesibles, completos y consistentes. La relación entre ambas variables se estableció a partir de la implementación del sistema informático basado en MQTT, el cual actúa sobre los procesos de captura y transporte de información para disminuir la latencia e incrementar la disponibilidad y consistencia de los registros en el repositorio central. Esto favorece una coordinación más eficiente entre las actividades académicas y operativas, facilita la participación de los estudiantes en procesos

reales de registro y análisis y reduce la carga administrativa asociada con la consolidación manual de la información.

1.2 Presentación del tema

El tema de este documento es un sistema informático con MQTT enfocado en la sincronización de datos. Como su nombre indica, el objetivo es lograr una sincronización óptima de datos utilizando el protocolo MQTT. Esto se debe a que el protocolo MQTT cuenta con mecanismos de cifrado de mensajes y autenticación de dispositivos, lo que permite establecer una comunicación confiable y ligera en redes con recursos limitados. Esta característica resulta adecuada para el entorno rural en el que se aplica el proyecto, debido a que presenta conectividad intermitente y recursos tecnológicos restringidos.

El estudio realizado representa un aporte considerable como guía práctica para el desarrollo y la optimización de futuros proyectos tecnológicos relacionados con el flujo constante de información. De esta manera, el proyecto promueve el desarrollo de soluciones tecnológicas robustas y escalables que pueden aplicarse posteriormente en contextos similares.

1.3 Ubicación y contextualización de la problemática

La Universidad Laica Eloy Alfaro de Manabí, extensión El Carmen, está situada en el mismo cantón que la provincia de Manabí. Esta institución académica opera desde 1988 y su ubicación es crucial para facilitar el transporte de los alumnos del área y de las comunidades rurales aledañas, lo cual es esencial para acceder a la educación superior. La institución ha creado y fomentado carreras técnicas y profesionales durante más de 30 años, las cuales están constantemente enfocadas en satisfacer las necesidades educativas y de desarrollo regional.

Durante ese periodo de tiempo, ha conseguido poner en marcha programas de pregrado, proyectos de investigación y acciones de vinculación con la comunidad, logrando así el desarrollo social del cantón y profesional con los alumnos. Aunque implica desafíos en cuanto a la planificación de recursos y los proyectos de gestión de programas académicos, se hace evidente la necesidad de mejorar la eficiencia de sus procedimientos administrativos y reforzar la organización interna.

El proyecto de cultivo de cacao se ubicó en el sector sur de las parcelas experimentales del campus Lastenia Vera (en la Extensión El Carmen), concretamente en la zona que está detrás del bloque antiguo, al lado del vivero y cerca del pequeño cerro protector y del tanque de agua. La selección de ese lugar se fundamentó en la calidad del drenaje y la textura del terreno, en el hecho de que se puede acceder directamente a la red de riego (lo que reduce la necesidad de desplazamientos constantes) y en su microclima favorable (sombra parcial apropiada para el cacao y consuelo al sol matutino).

También se aprecia la cercanía con el taller de electromecánica y el laboratorio de alimentos, ya que esto posibilita la creación de prototipos, el procesamiento de muestras y la intervención técnica sin demoras a nivel logístico. Debido A la falta de recursos tecnológicos y educativos, la selección también se fundamenta en criterios de optimización. El propósito es reunir actividades prácticas. Dé experimentación, educativas y demostrativas en un único espacio que disminuye los tiempos, logre una mejor coordinación entre las materias y posibilite el uso simultáneo de instalaciones y equipos.

Por lo tanto, la zona sugerida es adecuada para poner en marcha parcelas demostrativas, protocolos de manejo y actividades formativas que promuevan la incorporación práctica de los alumnos y el desarrollo eficaz de las prácticas cacaoteras...

1.4 Planteamiento del problema

1.4.1 Problematicación

La dificultad en la creación de procedimientos esenciales para asegurar la producción, como la calidad de las plantas de cacao, tales como: Control de riego, registro y seguimiento de las condiciones del medioambiente; esto contribuirá a la gestión administrativa de datos relacionados con la producción. Estos procesos se necesitan con un flujo de información fiable y constante necesarios para tomar decisiones de ayuda prevención y oportuna.

En el vivero del campus Lastenia Vera existen limitaciones que afectan directamente con la eficiencia y la confiabilidad de las actividades que están atadas con la producción y con la enseñanza practica del cultivo de cacao. La principal de ellas surge del registro manual ya que una buena parte de las mediciones se anota en cuadernos o hojas de cálculo lo que puede

ocasionar errores en la transcripción las omisiones y la pérdida de datos esto ocasiona un retraso en la consolidación de la información que se necesita para tomar acciones oportunas

En segundo lugar, las herramientas y los soportes tecnológicos existentes no están articulados ni centralizados. Esto hace que sea difícil coordinar entre los departamentos académicos y operativos y que no se tenga una visión completa de la operación. Se verificó que los registros seguían conservándose en hojas de cálculos y cuadernos durante el diagnóstico del 1 de agosto y del 15 de septiembre del 2025. La falta de un repositorio enfocado en sus métodos de entrega periódica provoca que el tiempo para consolidar datos se alargue, dificultando así la trazabilidad de las intervenciones.

En tercer lugar, debido a la falta de un procedimiento uniforme para asegurar la actualización frecuente de los registros administrativos y de campo, se producen bases de datos inconsistentes y obsoletas. La escasez de regularidad y coherencia necesita trabajos manuales para necesita ajustar los datos antes de cualquier tipo de análisis , lo que reduce la seguridad en las evaluaciones experimentales y en la capacidad de reproducir un resultado exitoso alcanzado. Trabajos manuales para limpiar y ajustar los datos antes de cualquier tipo de análisis, lo que reduce la seguridad en las evaluaciones experimentales y en la capacidad de reproducir un resultado exitoso alcanzado

En cuarto lugar, estas deficiencias en la actualización y el registro constante limitan la capacidad de los responsables para tomar decisiones apropiadas ante cambios significativos en las variables de gestión (por ejemplo, la temperatura, la humedad del suelo o el estado fitosanitario de las plantas), lo que repercute negativamente en la planificación y control de las prácticas agrícolas. Las situaciones de diagnóstico registradas en las que la demora en la verificación de datos impidió la detección precoz de estrés hídrico, lo que provocó reacciones reactivas. El hecho de no contar con datos de manera oportuna también dificulta la evaluación exhaustiva de la efectividad y eficiencia de los tratamientos que se aplican en parcelas experimentales...

Finalmente, los métodos actuales no hacen posible la incorporación progresiva de nuevas herramientas o módulos para la gestión, lo que obstaculiza el desarrollo y la actualización del vivero. La investigación tiene como objetivo responder las preguntas que se presentan a continuación, teniendo en cuenta el punto de problematización mencionado anteriormente:

- ¿De qué manera afecta la ausencia de centralización de datos y el manual a la planificación, en el vivero de la Extensión El Carmen (Lastenia Vera) a la planificación, control y decisiones que se toman en esas prácticas cacaoteras?
- ¿De qué manera la falta de un mecanismo automatizado de sincronización influye en el tiempo de las verificaciones y en la omisión de los registros operativos del vivero?
- ¿De qué manera afecta la puesta en marcha del sistema a las habilidades y la integración práctica de los alumnos, así como a la carga administrativa del personal y al uso eficaz de insumos (como fertilizantes y agua) en las parcelas experimentales?

1.4.2 Génesis del problema

El vivero de cacao de la Extensión El Carmen (Lastenia Vera) funcionaba con un sistema manual de registros que se estableció al principio. Sin embargo, desde diciembre de 2024, a medida que las actividades y la infraestructura aumentaban, comenzaron a notarse vacíos en el control operativo, retrasos en la comprobación de los datos y omisiones en las bitácoras. No obstante, en el pasado, cuando la cantidad de información era bastante baja, parecía que el manejo manual era suficiente; no obstante, con el incremento de actividades y la necesidad de un seguimiento más riguroso y detallado, empezaron a surgir pequeños vacíos en la administración, como en la manera de registrar los datos.

Sin embargo, no fue un solo evento lo que reveló la situación, sino una combinación de varios factores: el número elevado de plántulas en proceso, la demanda de informes más exactos y la concurrencia simultánea de varios grupos y carreras en las mismas parcelas. Por lo general, la presión para responder de manera eficaz y rápida a las condiciones de cultivo. Todo esto demostró que los métodos empleados, tanto tradicionales como no, ya no son suficientes para preservar la misma calidad de gestión.

1.4.3 Estado actual del problema

Aparentemente, el panorama que atraviesa el Campus Lastenia Vera - Ext. El Carmen se origina a partir de una serie de circunstancias que, por sobrecarga, han expuesto las restricciones en la gestión de información. Los registros manuales y la comunicación directa sirvieron durante cierto tiempo, pero a medida que el vivero crecía y surgía la necesidad de atender más plantas y reportes, se hizo evidente que este esquema ya no era suficiente. Con el

tiempo sostener la vigencia y la coherencia de la información se volvió una dificultad creciente. No se trata de un problema puntual es la suma de varias situaciones concretas que percuten en el manejo diario del vivero.

Actualmente, gran parte de la información se registra de forma manual en cuadernos y hojas de cálculo, lo que provoca errores, omisiones y retrasos en la consolidación de datos. Además, la comunicación interna es tardía y los registros administrativos no están centralizados, lo que obstaculiza la coordinación entre las actividades de producción y los procedimientos formativos. Estas condiciones crean una carga adicional para el equipo a cargo y restringen la oportunidad y la coherencia de los datos requeridos para tomar decisiones. Para sugerir medidas de mejora, es preciso caracterizar estas fallas con exactitud por todo lo anterior.

1.5 Diagrama causa – efecto del problema

Ilustración 2 Árbol de problemas sobre la inconsistencia de registros en el vivero



1.6 Objetivos

1.6.1 Objetivo general

Implementar un sistema de sincronización eficiente de datos distribuidos utilizando el protocolo MQTT para mejorar la confiabilidad operativa del monitoreo automatizado.

1.6.2 Objetivos específicos

- Identificar las falencias que existen en la gestión de registros y el acceso a la información del vivero de cacao, esto mediante la observación directa en encuestas y entrevistas al personal para conocer las principales dificultades que se presentan durante el manejo de la información.
- Revisar información bibliográfica relacionada con protocolos de comunicación livianos, especialmente MQTT, sistemas de gestión de datos en viveros agrícolas y procesos de digitalización, mediante la consulta de fuentes relacionadas con el tema, para conocer alternativas que puedan adaptarse a las condiciones del vivero.
- Recolectar y organizar la información relacionada con las actividades y registros del vivero mediante bitácoras entrevistas al personal y observaciones realizadas en las parcelas para conocer cómo se manejan actualmente los datos y que dificultades se presentan en este proceso.
- Diseñar un modelo de sincronización de datos basados en el modelo y protocolo MQTT, integrando los medios de recolección de información, el broker y una base de datos centralizada de acuerdo con los recursos disponibles esto para facilitar el intercambio y la actualización de datos del vivero.
- Evaluar el funcionamiento del sistema de sincronización implementado, mediante métodos cualitativos y cuantitativos que permitan analizar aspectos como la pérdida de datos, el tiempo de consolidación y la opinión del personal para poder determinar los resultados obtenidos y su aporte al manejo de la información del vivero.

1.7 Justificación

La puesta en marcha de este sistema informático, que utiliza el protocolo MQTT para sincronizar datos en la plantación de cacao de la Extensión El Carmen (Lastenia Vera), se debe

a las carencias registradas en la gestión de información, como son el registro manual en hojas de cálculo y cuadernos, la ausencia de centralización y los retrasos en la consolidación de datos. Se usa el sistema MQTT que se usa para mandar mensajes de forma ligera desde registros de campo o móviles a un servidor central. Con esto se logra que la información sea coherente y esté a la mano para decidir en temas operativos o académicos.

Además, la propuesta permite a los alumnos agilizar el procesamiento y la trazabilidad de datos frente al manejo del cacao, generando una utilización de fertilizantes y riego más acordes a los insumos. Esta visión concuerda con el deber de formación de la Universidad Uleam, lo que favorece la modernización y prospera sostenibilidad del vivero de cacao con costos más reducidos que generen mayor posibilidad de crecimiento.

1.8 Impactos esperados

1.8.1 Impacto tecnológico

La puesta en marcha de un sistema informático basado en MQTT posibilitará centralizar y a su vez sincronizar oportunamente los registros operativos del vivero, lo cual reducirá tanto los errores en las bitácoras como en los tiempos de consolidación de la información. Al tratarse de un protocolo ligero y capaz de operar sobre redes con conectividad limitada, resulta viable sumar de forma progresiva formularios en dispositivos móviles como también dispositivos de captura y paneles para el seguimiento de indicadores. Así de este modo se agilizan las intervenciones y la generación automática de informes y se conforma de un repositorio único y confiable que sirve de base para el análisis posteriores y mejoras que puedan crecer o no con el tiempo.

1.8.2 Impacto social

Al Estandarizar los formatos y los canales de registro, se facilitará el acceso compartido a la información actualizada para planear actividades académicas y prácticas. Esto ayudará a que el personal docente, estudiante y técnico se coordine mejor. Como resultado, se anticipa que la capacitación relacionada con la gestión del cacao mejorará, que las obligaciones administrativas de los empleados del vivero disminuirán y que las competencias digitales de

los estudiantes aumentarán. Estos últimos tendrán la posibilidad de participar en procesos reales de recopilación, análisis y uso de datos relacionados con la producción.

1.8.3 Impacto ecológico

La disponibilidad de los datos de riego, estado fitosanitario y manejo de insumos permitirá optimizar aplicaciones y programaciones, reduciendo el uso innecesario de agua y agroquímicos así también minimizando riesgos de estrés o pérdidas de las plantas de cacao. Al optimizar el registro y monitoreo de prácticas de manejo, se toman decisiones más exactas y sostenibles en las parcelas experimentales. De esta manera, se logra una gestión más eficaz de los recursos y se mantiene la productividad y la salud del suelo a mediano plazo.

CAPÍTULO II

2 MARCO TEÓRICO

2.1 Antecedentes históricos

Los sistemas informáticos han progresado desde mediados del siglo XX, pasando de entornos computacionales centralizados en mainframes a arquitecturas distribuidas (Ceruzzi, 2003). Los sistemas de gestión de bases de datos relacionales y el modelo transaccional se afianzaron en los años setenta para garantizar la integridad (Codd, 1970). Se produjo un paradigma cliente-servidor en la década de los noventa, y la expansión de las redes locales hizo posible que aplicaciones multiusuario intercambiaran información en tiempo casi real. Con la expansión del internet, lograron surgir aplicaciones web de servicios orientados a la integración entre sistemas heterogéneos (Tanenbaum & van Steen, 2017).

Las arquitecturas más escalables fueron desarrolladas en 2010 gracias a la computación en la nube y los dispositivos móviles. Al mismo tiempo, la captura de datos se expande a formularios y sensores móviles en el campo gracias a Internet. En este contexto, los autobuses de eventos y las estructuras. De mensajería asíncrona se volvieron relevantes para desacoplar a los productores y consumidores de información, lo cual según los mismos autores posibilitó una sincronización efectiva en circunstancias con conectividad variable.

Historia de variable independiente: Sistema informático con el protocolo MQTT (Message Queuing telemetry Transport) fue creado a fines de 1990 para telemetría en enlaces con ancho de banda limitado y alta latencia. Su modelo publish/suscribe, mediado por un bróker. Permite que distintos clientes publiquen o se suscriban a “Tópicos” sin conectarse entre sí, lo que reduce el acoplamiento y favorece la escalabilidad. MQTT se estableció como una opción liviana en comparación con HTTP en situaciones con conectividad escasa o intermitente, así como recursos limitados, desde que OASIS lo estandarizó en 2014 y lanzó la versión 5.0 en 2018 (Flores Cruz & López Ortiz , 2023)

La sólida infraestructura de brókers y herramientas de código abierto (como EMQZ, Eclipse Mosquitto o HiveMQ Community) SDKS para varios idiomas y conexiones a bases de datos y paneles de visualización. Se propagaron, al mismo tiempo, modelos arquitectónicos que fusionan este protocolo con almacenamiento central (ya sea relacional o de series temporales),

los autores citados señalan además que se propagaron modelos arquitectónicos que fisieron este protocolo con almacenamiento central ya sea relacional o de series temporales con validaciones de esquemas en el borde y tableros para supervisión.

Historia de la variable dependiente: sincronización de información. La sincronización de datos ha sido un tema de preocupación histórica desde que surgieron varios puntos de captura o copias. Durante las Décadas de los años ochenta y noventa, se popularizó el uso de sistemas multimaestro y la replicación maestro-esclavo en bases de datos distribuidas. Con la movilidad, se presentan estrategias "offline-first" para conciliar modificaciones registradas en dispositivos con conectividad inestable, incluyendo directrices de resolución de conflictos y control de versiones. (Chugá Rosero, 2022).

La sincronización en los sistemas distribuidos actuales se basa en flujos de eventos, colas de mensajes y almacenamiento eventual o fuertemente consistente, dependiendo de la importancia del dominio. En contextos operativos como un vivero, la sincronización no se limita a "subir los datos", sino que también incluye garantizar que sean accesibles, tengan una latencia aceptable y sean completos y consistentes entre los participantes que registran simultáneamente. La madurez de estos métodos permite medir con indicadores como la tasa de duplicados y omisiones, el porcentaje de registros sincronizados, de acuerdo con la misma fuente, la madurez de estos métodos permite medir con indicadores como la tasa de duplicados y omisiones, el porcentaje de registros sincronizados, el tiempo de publicación a disponibilidad y la eficacia de las validaciones.

Historia de sistemas informáticos para sincronización de datos medioambientales. La vigilancia del medioambiente tiene sus orígenes en redes de estaciones con dataloggers y en la transmisión periódica por radio o telefonía desde los años setenta del siglo XX. En los noventa y dos mil se incorporaron tecnologías GSM/GPRS y satelitales, mientras que las redes de sensores inalámbricos (IEEE 802.15.4/Zigbee) permitieron capturas locales de variables como temperatura, humedad y precipitación (Ortega, 2022).

Desde que el Internet de las Cosas (IoT) comenzó a expandirse en la década del 2010, se ha vuelto común utilizar arquitecturas que integran pasarelas, servicios en la nube y nodos de campo con bajo consumo. Al mismo tiempo, se establecieron plataformas "offline-first" y formularios móviles para equipos humanos en campo, los cuales sincronizan cuando la

conectividad está disponible. Actualmente, soluciones de código abierto y servicios en la nube permiten desplegar pipelines de ingestión, almacenamiento de series temporales, validación y alertamiento a costos decrecientes, el mismo autor destaca que actualmente las soluciones de código abierto y los servicios en la nube permiten desplegar pipelines de ingestión, almacenamiento de series temporales, validación y alertamiento a costos decrecientes, lo que favorece proyectos de monitoreo en instituciones educativas y productivas.

2.2 Antecedentes de investigaciones relacionadas al tema presentado

Universidad Miguel Hernández (ubicada en España). "Internet de las cosas (IoT) y agricultura 4.0: Vigilancia de la producción agrícola en base a la humedad del terreno, por medio de sensores y actuadores que se rigen por el protocolo MQTT". Este TFG aplica una estructura de sensores de humedad, dispositivos para riego y comunicación MQTT a un servidor central. Su propósito principal es regular el riego a partir de lecturas en tiempo real y normas automáticas. Informa una latencia de comunicación apropiada para decisiones de riego, así como una disminución en el consumo de agua y en los tiempos de respuesta. Los hallazgos respaldan lo pertinente de MQTT en contextos agrícolas con conectividad variable (Ortega, 2022)

Universidad SEK Internacional (Ecuador). "Desarrollo y aplicación de un sistema para el riego inteligente y la supervisión mediante IoT". El trabajo combina nodos con sensores, un bróker MQTT, una base de datos y un panel de visualización para una granja agrícola. La meta es mejorar el riego y el monitoreo de las variables ambientales a través de notificaciones y control remoto. Las pruebas de campo muestran que se ha reducido el consumo de agua y que la disponibilidad de datos para los técnicos ha mejorado. Se deduce que MQTT permite una operación en tiempo casi real y la escalabilidad (Chugá Rosero, 2022)

Universidad Nacional Autónoma de México (México) La solución que se presenta en la tesis "Plataforma web para recolección y visualización de datos en redes de sensores ambientales usando MQTT" combina dispositivos de campo con un bróker MQTT y un repositorio central al cual se puede acceder. Su propósito principal es recoger y sincronizar datos ambientales de manera eficaz, facilitando su análisis y visualización en un tiempo casi real. El diseño incluye la autenticación básica, el almacenamiento persistente y los temas estructurados, así como también los resultados que muestran un aumento de la oportunidad y de la coherencia de los registros en comparación con las capturas manuales. Debido a la semejanza entre los flujos de

datos y las exigencias de sincronización, la propuesta puede aplicarse también en contextos agrícolas y de viveros (Flores Cruz & López Ortiz , 2023).

Los documentos que se han investigado y mencionado: En los antecedentes estudiados (Flores Cruz y López Ortiz, 2023; Ortega, 2022; Rosero, 2022) se observa un intervalo de coincidencia: todos confirman que el protocolo MQTT es la columna vertebral más adecuada para la telemetría en el sector agroambiental. Estos estudios concuerdan en que la arquitectura de publicación/suscripción es eficaz para vincular sensores y concentrar el proceso de toma de decisiones, lo cual permite sobrepasar la lentitud de los registros manuales.

No obstante, al analizar el aspecto técnico más a fondo, se observa una falta de validación de robustez. Los antecedentes se centran en la funcionalidad operativa, pero no ofrecen una cuantificación precisa del rendimiento de la red; específicamente, carecen de métricas sobre latencia de extremo a extremo, tasa de paquetes perdidos o la actuación del bróker en situaciones de estrés prolongado. No obstante, esta seguridad es frecuentemente abordada de manera superficial, sin profundizar en aspectos cruciales como la Lista de Control de Acceso (ACL) o el cifrado TLS.

Por otro lado, la siguiente propuesta no se limita a establecer la comunicación, sino que también incluye la auditoría de esta. El proyecto se caracteriza por establecer políticas de calidad del servicio (QoS) y de retención de mensajes que se adaptan a la importancia del cultivo, además de incorporar métricas precisas sobre la disponibilidad y el tiempo para volver a conectarse. La estrategia emplea una capa de seguridad en base a una autenticación robusta con almacenamiento de datos temporales, que garanticen la continuidad de procesos operativos en áreas rurales con escasa conectividad, logrando modernización al establecer una base que pueda superar el alcance de investigación antiguas.

2.3 Definiciones conceptuales

2.3.1 Sistema informático con MQTT

Se define como una infraestructura tecnológica estructurada bajo el modelo de publicación-suscripción, con el objetivo de mejorar el envío de datos en redes que tienen un ancho de banda limitado. Esta estructura introduce un agente intermediario o corredor entre los dispositivos de

campo y el almacén central, lo que posibilita la separación de los procedimientos de lectura y de almacenamiento. Con el fin de esta investigación, el sistema utiliza los niveles de calidad de servicio (QoS) del protocolo para garantizar la sincronización completa de las variables ambientales. Esto asegura que la información se mantenga y transmita adecuadamente, incluso en caso de que haya inestabilidad en la conexión, como suele ocurrir en áreas rurales (Flores Cruz & López Ortiz , 2023)

2.3.1.1 Protocolo MQTT propósito, versiones y casos de uso en agro

El protocolo MQTT es una norma de mensajería liviana que se fundamenta en la pila TCP/IP, concebida especialmente para redes con un ancho de banda escaso o con latencia elevada. Su estructura asíncrona permite reducir el intercambio de datos a través de una cabecera binaria de únicamente 2 bytes, lo cual disminuye significativamente el consumo energético en comparación con HTTP. Esta eficacia lo transforma en la solución perfecta para la telemetría M2M, posibilitando que se conserven abiertos los canales de comunicación en infraestructuras inestables (Tanenbaum & van Steen, 2017).

Es crucial su desarrollo normativo para que el estándar sea ampliamente implementado en el sector industrial. La edición 3.1.1 se enfocó en realizarse con facilidad en hardware económico, mientras que la versión 5.0 incorporó funciones más complejas para gestionar sesiones y códigos de diagnóstico. Estas optimizaciones permiten que los sistemas actuales sean capaces de detectar desconexiones con precisión y gestionar metadatos de usuarios sin sobrecargar la lógica del equipo final, lo cual mejora la escalabilidad (Banks & Gupta, 2014).

Para la supervisión a distancia de variables climáticas dispersas en la agricultura de precisión, este protocolo es esencial. Al desacoplar la lectura de los sensores del procesamiento, MQTT garantiza que la falta de continuidad en la red celular en áreas rurales no cause que se pierdan datos históricos. La literatura técnica respalda su empleo en sistemas de riego automatizados, mostrando que asegura una respuesta a tiempo frente a situaciones críticas y mejora la utilización de recursos hídricos (Ortega, 2022).

2.3.1.2 Modelo publish/subscribe: tópicos, wildcards y estructura de mensajes

El modelo de publicación/suscripción establece una separación en términos de tiempo y espacio entre los componentes, lo que lo distingue del modelo clásico cliente-servidor. En este paradigma, los productores de datos y los consumidores no se comunican directamente, sino que lo hacen mediante "tópicos", que son canales lógicos. Esta abstracción favorece que la escalabilidad sea horizontal, ya que permite agregar nuevos servicios de visualización sin necesidad de cambiar la configuración de los sensores ya existentes en el campo (HiveMQ, 2025).

Se utiliza un orden jerárquico de temas separados por barras para que estos flujos de datos no se mezclen este esquema funciona con caracteres comodín donde el signo numeral sirve para ver todas las ramas del árbol mientras que el signo más permite entrar a niveles concretos de la jerarquía. Saber usar esta sintaxis es fundamental para mejorar el tráfico en la red ya que segmenta el acceso por zonas y evita recibir mensajes idos en los sistemas de monitoreo (Eclipse Foundation, 2025).

Respecto a la unidad de información el payload que ayuda al transporte de cada mensaje MQTT es independiente del protocolo, esto lo deja en manos de la aplicación la definición de su formato. En los desarrollos actuales de IoT suele optarse por JSON los cuales ayudan con la estructura de la información de forma estandarizada lo cual admite junto al valor medido metadatos de contexto como la marca temporal. Contar con esos datos adicionales resulta determinante cuando el mensaje no llega a la plataforma central, pues son los que hacen posible su trazabilidad y la verificación de la integridad de cada lectura (Flores Cruz & López Ortiz , 2023).

2.3.1.3 Bróker MQTT: funciones, despliegue y alta disponibilidad

El Bróker desempeña el papel de ser el centro de coordinación de la red, encargado de recibir, filtrar y redistribuir los mensajes entre todos los clientes conectados. Su función no se limita al enrutamiento; actúa como un punto de control de seguridad que autentica los dispositivos y comprueba los permisos para acceder. En términos de arquitectura agrícola, es habitual implementar instancias locales en el borde para garantizar que las decisiones de control tengan

la menor latencia posible y que la operatividad no se vea afectada si se pierde la conexión a internet. (Chugá Rosero, 2022).

El Bróker, debido a la importancia crítica de su función, debe implementar estrategias de redundancia en su despliegue para prevenir puntos únicos de falla. Los sistemas sólidos establecen clústeres de alta disponibilidad en los que varios nodos comparten el estado de las sesiones en tiempo real. En caso de que el servidor principal se caiga, los balanceadores de carga dirigen el tráfico hacia los nodos secundarios, lo cual asegura que el servicio de monitoreo esté disponible las veinticuatro horas del día (Ceruzzi, 2003).

La administración centralizada de estas conexiones también posibilita la gestión del ciclo de vida de los dispositivos IoT que están distribuidos. El servidor supervisa los tiempos de inactividad (keep alive) y desconecta con eficacia las conexiones que están muertas para liberar recursos. En contextos en los que miles de sensores pueden tratar de reconectarse al mismo tiempo después de un corte de energía, esta habilidad para una gestión eficaz es esencial, ya que previene la congestión de la red interna (Elmasri & Navathe, 2016).

2.3.1.4 Calidad de servicio (QoS), retención, sesiones y Last Will

La confiabilidad con la que se entrega información y se maneja a través de los niveles de calidad de servicio (QoS), un contrato operativo entre el servidor y el cliente. El protocolo brinda tres posibilidades: QoS 0 para envíos rápidos que no tienen garantía, QoS 1 que garantiza la entrega por medio de confirmaciones y QoS 2, que otorga una entrega única y precisa. La selección apropiada de este parámetro posibilita balancear el uso de ancho de banda con la importancia crucial que tienen las alertas de seguridad (Banks et al., 2019).

Los mecanismos complementarios, como los mensajes retenidos, garantizan que los recién suscritos obtengan de inmediato el valor más reciente de un tópico. Por otro lado, la función de Última Voluntad (Last Will) posibilita que el Bróker avise automáticamente si un cliente se desconecta de manera irregular. Esto funciona como un sistema de alerta temprana en un vivero, notificando a los operadores sobre problemas en los sensores antes de que impacten el cultivo. (HiveMQ, 2025).

El principio de sesión persistente es la base de la estrategia offline-first en zonas con conectividad intermitente. Cuando está activo, el servidor almacena en una cola todos los mensajes destinados a un cliente desconectado y los envía cuando se establece nuevamente la conexión. Esta funcionalidad garantiza que los registros históricos permanezcan en la base de datos sin ser eliminados durante las ventanas de desconexión que son habituales en la infraestructura rural. (Ortega, 2022).

2.3.1.5 Seguridad y control de acceso: TLS, autenticación y ACLs

El uso de túneles cifrados TLS (Transport Layer Security) es la base de la seguridad en el transporte de telemetría. Como MQTT funciona sobre TCP, encapsular la comunicación resguarda las credenciales y la carga útil ante los ataques de interceptación. El cifrado extremo a extremo es la principal barrera necesaria para prevenir la manipulación maliciosa de los sistemas de control en contextos agrícolas abiertos, donde las redes inalámbricas son susceptibles (Eclipse Foundation, 2025).

En la capa de aplicación del control de acceso se maneja con las ACL lo que son las listas de control de acceso del broker. Con ellas se define que puede hacer cada cliente a que tópicos se pueden suscribir y en cuales puede publicar. En este proyecto cada estación queda amarrada solo a los tópicos que le tocan. La idea es aplicar el mínimo privilegio lo que quiere decir que da a cada dispositivo apenas los permisos que necesita y nada más. Así si alguien llegara o lograra comprometer la estación no podría publicar lecturas haciéndose pasar por otra ni meterse en los datos del resto ya que el broker le rechaza cualquier acción fuera de lo que tiene permitido (Banks & Gupta, 2014)

La protección contra la saturación de recursos a través de límites de tasa (rate limiting) también es una de las mejores prácticas operativas. Los brókers actuales posibilitan la configuración de cuotas máximas para el envío de mensajes por segundo con el objetivo de prevenir ataques DoS (Denegación de Servicio). Estas acciones de defensa, junto con la división de la red en puertos seguros, robustecen la infraestructura tecnológica y aseguran que el servicio esté disponible frente a incidentes externos (Flores Cruz & López Ortiz , 2023).

2.3.1.6 Arquitectura IoT de la solución propuesta

La arquitectura de este proyecto se compone de niveles jerárquicos, desde la percepción física hasta la visualización. Los nodos sensores, ubicados en el borde, digitalizan las variables ambientales y solo funcionan como editores de datos optimizados. Estos flujos se unen en una pasarela que maneja la conectividad con el núcleo del sistema, funcionando como un límite seguro entre el contexto operativo del vivero y los servicios de procesamiento en la nube (Tanenbaum & van Steen, 2017).

El Bróker MQTT se combina con servicios de backend que están desacoplados en el elemento central de procesamiento. Esta estructura separa el plano de datos del plano de control, lo que posibilita que un servicio "trabajador" procese la información asíncronamente hacia la base de datos. Esta separación hace más fácil la escalabilidad y el mantenimiento, pues posibilita que se actualice la lógica del negocio o que la capacidad de almacenamiento se expanda sin que el flujo de mensajería en tiempo real se corte (Chugá Rosero, 2022).

Por último, la capa de aplicación muestra a los usuarios la información validada mediante tableros de control interactivos. La interfaz elimina la incertidumbre visual al procesar datos y estados de conectividad que provienen de los mensajes retenidos. Esta arquitectura justifica la conveniencia de concentrar la inteligencia del sistema para simplificar la toma de decisiones agronómicas inmediatas, fundamentadas en información coherente y actualizada (Ceruzzi, 2003).

2.3.1.7 Integración y persistencia: mapeo de mensajes a base de datos y esquema de datos

Para integrar datos telemétricos, es necesaria una normalización que convierta mensajes volátiles en registros estables. Un servicio de ingestión se suscribe a los flujos de datos, descompone el contenido JSON y comprueba si este cumple con un esquema establecido previamente. En teoría, esta fase funciona como un tamiz de calidad que elimina paquetes corruptos y uniformiza las marcas de tiempo, garantizando que la base de datos únicamente Para el almacenamiento eficaz, se prefiere el empleo de bases de datos de series temporales (TSDB) en lugar de los modelos relacionales tradicionales. Estas tecnologías están diseñadas para manejar cargas masivas de escritura secuencial y consultas que se basan en rangos de

tiempo. Su arquitectura posibilita la implementación de políticas nativas para resumir y retener datos, asegurando que el desempeño del sistema no disminuya conforme aumenta el volumen histórico a lo largo de los años de funcionamiento. (Elmasri & Navathe, 2016).

El esquema de persistencia diseñado enriquece el valor de la medición con metadatos técnicos del protocolo MQTT. Al almacenar parámetros como la calidad de señal y el identificador del dispositivo junto con la variable ambiental, se habilita una observación completa del sistema. Esto permite que los administradores vinculen las anomalías en los datos con fallos en la infraestructura de comunicación, completando así el ciclo de inteligencia para el mantenimiento predictivo. (Ortega, 2022).

2.3.2 Sincronización de datos.

Desde la mirada de los sistemas distribuidos, sincronizar significa lograr que varias copias de un mismo grupo de datos se mantengan coherentes entre ellas e incluso cuando se actualizan al mismo tiempo en lugares distintos y aparece el riesgo de que se desincronicen estos datos. No se trata de solo copiar o mover archivos de un lado a otro lo que implica definir reglas que mantengan datos consistentes entre todos los nodos, tanto en el servidor central como en los dispositivos de campo que están en los extremos de la red (Özsu & Valduriez, 2011).

Para enfrentar este reto existen dos caminos los cuales son la replicación síncrona y la asíncrona lo que la asíncrona busca que todos vean el mismo dato en el mismo instante lo que es una consistencia fuerte pero lo paga con más latencia porque hay que esperar la confirmación de todos. La asíncrona en cambio es la más usada en aplicaciones móviles y de internet de las cosas lo que prioriza que el sistema responda de inmediato, aunque durante un breve lapso el dato del dispositivo y el del servidor no coincidan. A este modelo se lo llama “consistencia eventual” y esa pequeña diferencia demora no solo se tolera si no que resulta ser indispensable cuando la conexión es intermitente como suele ocurrir en lugares como el vivero (Kleppmann, 2017).

En las arquitecturas actuales, esta sincronización suele apoyarse con patrones orientados a eventos (Event-Driven Architecture). En lugar de consultar la base de datos directamente los cambios se envían como eventos a través de un bus de mensajería. Esto desacopla a quien produce el dato de quien lo va a consumir de este modo que una estación de campo puede

mandar sus lecturas sin quedarse esperando a que el servidor confirme que las guardo. Justamente esta independencia es la base del enfoque offline-first donde lo primero que se requiere es capturar el dato localmente y dejar la sincronización con el servidor para cuando haya un canal disponible (Michelson, 2006).

2.3.2.1 Conceptualización y alcance de la sincronización en sistemas distribuidos

La sincronización de datos en arquitecturas distribuidas se compone de protocolos lógicos que están creados para preservar la consistencia del estado entre varios nodos de almacenamiento separados, este proceso va más allá de la mera duplicación de archivos; supone la coordinación de algoritmos que son capaces de propagar actualizaciones, identificar divergencias entradas en picas y solucionar disputas de concurrencia. El propósito final es garantizar que todos los participantes del sistema, sin importar su localización física o la latencia de la red, tengan una percepción conjunta y válida de los datos operativos (Özsu & Valduriez, 2011).

El Teorema CAP, que establece las limitaciones intrínsecas entre la consistencia, la disponibilidad y la tolerancia a las particiones, define el alcance de la sincronización desde un enfoque teórico... En los sistemas de monitoreo ambiental, en los cuales la red tiene una alta probabilidad de fallos, a menudo se le da prioridad a la disponibilidad antes que a una consistencia fuerte e inmediata, esto requiere la implementación de modelos de consistencia eventual, en los que se admite una ventana temporal de desacuerdo entre los datos del borde (vivero) y el servidor central, asegurando que, después de restaurada la comunicación, el sistema llegue a un estado idéntico (Tanenbaum & van Steen, 2017).

Definir la topología de la red y el sentido del flujo de datos es necesario para implementar técnicamente estos conceptos. En el marco de esta investigación, se utiliza un esquema ascendente y unidireccional para la telemetría (del sensor a la nube) y uno descendente para las órdenes de control. Como la fuente de verdad para las mediciones es siempre el dispositivo de campo y la autoridad para las configuraciones está en el servidor, esta división facilita la administración de conflictos al disminuir la complejidad de los algoritmos necesarios para alcanzar un consenso (Kleppmann, 2017).

2.3.2.2 Modos de sincronización: tiempo real, por lotes y near real-time

La transmisión instantánea de eventos a medida que se producen, con una latencia mínima extremo a extremo, es típica de la sincronización en "tiempo real". Este modo es característico de las arquitecturas enfocadas en eventos (Event-Driven), en las que cada lectura de un sensor activa un mensaje único dirigido al bróker a pesar de que proporciona la información más fresca, su inconveniente es el alto sobrecargado de red debido al envío ininterrumpido de cabeceras, lo cual puede colapsar los canales de comunicación inestables o estrechos en áreas rurales (Michelson, 2006).

Por el contrario, el procesamiento por lotes, (procesamiento por lotes) compila una serie de registros en la unidad local durante un período de tiempo establecido previamente y los envía posteriormente en un único paquete comprimido. registros en la unidad local durante un período de tiempo establecido previamente y los envía posteriormente en un único paquete comprimido. Esta estrategia mejora notable el ancho de banda y el consumo de energía del transmisor, aunque introduzca un retraso intencionado en la visibilidad de la información. Es el método perfecto para informes analíticos históricos en los que la inmediata no es esencial, como el resumen diario del consumo de agua del sistema de riego (Bass et al., 2013).

El enfoque híbrido, conocido como “casi tiempo real”, busca un equilibrio pragmático al combinar ventanas breves de acumulación (por ejemplo de 15 segundos) con notificaciones prioritarias. En este proyecto se aplica ese esquema para la telemetría rutinaria con el fin de suavizar los picos de tráfico que pueden ovacionar en la red mientras que las alertas críticas como un fallo de bomba se manejan como eventos de tiempo real puro. Esta separación permite optimizar los recursos de la infraestructura sin comprometer la capacidad de respuesta ante índices (Chugá Rosero, 2022).

2.3.2.3 Consistencia y resolución de conflictos en registros concurrentes

Cuando Existen múltiples nodos que pueden escribir datos al mismo tiempo, es imposible evitar que surjan conflictos de actualización. Para resolver estas discrepancias, se necesitan tácticas deterministas; Una de las más frecuentes es la política Last-Write-Wins (LWW), que emplea marcas de tiempo sumamente precisas para que el dato más nuevo reemplace al

anterior. No sin embargo, esta técnica presupone que los relojes de todos los dispositivos se encuentran sincronizados a la perfección (NTP), lo cual representa un reto técnico en hardware de bajo costo extendido en el campo (Shapiro et al., 2011).

Los Tipos de Datos Replicados Libres de Conflictos (CRDT) son utilizados para situaciones más complejas; Estas son estructuras matemáticas que aseguran la convergencia automática sin requerir un consenso central, a pesar de que su implementación es cara en términos de computación, garantizan que las operaciones conmutativas (por ejemplo, aumentar el contador de litros de agua) alcancen el mismo resultado final independientemente del orden en que lleguen los mensajes. Esto es crucial para preservar la integridad de los contadores acumulados en el sistema de riego (Kleppmann, 2017).

La inmutabilidad de los datos de telemetría permite simplificar la resolución de conflictos dentro del proyecto. Como cada lectura de sensor es un acontecimiento temporal único y no una actualización de algo anterior, no Hay conflictos de escritura tradicional. El reto se centra en manejar los duplicados generados por retransmisiones de red, que se filtran en la capa de ingestión mediante el uso de identificadores únicos de mensaje (UUID) para asegurar que la operación de almacenamiento sea idempotencia (Elmasri & Navathe, 2016).

2.3.2.4 Dimensiones e indicadores: disponibilidad, latencia, completitud y consistencia

La calidad del proceso de sincronización no puede evaluarse subjetivamente; requiere métricas cuantificables como la "Disponibilidad del Servicio" (Uptime). Este indicador mide el porcentaje de tiempo que el bróker de mensajería está accesible para recibir datos. En sistemas críticos de agro-tecnología, se busca alcanzar al menos un 99.9% de disponibilidad, lo que implica diseñar arquitecturas tolerantes a fallos que minimicen las ventanas de mantenimiento o caída inesperada del servidor (Bass et al., 2013).

Otra dimensión relevante es la latencia de extremo a extremo que mide el tiempo que mide el tiempo transcurrido desde que el sensor detecta el fenómeno físico hasta que el dato aparece en el tablero de control. Ese tiempo abarca el procesamiento en el borde, el envío por la red y el almacenamiento en la base de datos. Medir esta latencia ayuda que se detecten cuellos de botella en la infraestructura y ayuda a ajustar los parámetros de QoS de modo que las decisiones de riego se tomen sobre información vigente (Banks et al., 2019).

Por último, "Compleitud" y "Consistencia" examinan la integridad de los datos. La completitud, al comparar la cantidad de paquetes que los sensores envían con la que se almacena realmente, muestra el índice de pérdida de la red. Por su parte, la consistencia verifica la ausencia de duplicados y el orden cronológico correcto de las series temporales. Estos indicadores son esenciales para auditar la efectividad de las políticas de almacenamiento en caché y retransmisión que se han aplicado en los nudos del vivero (Ortega, 2022).

2.3.2.5 Esquema offline-first y tolerancia a conectividad intermitente

El paradigma offline-first considera que la falta de red es un estado normal y no una excepción, lo cual revierte la dependencia convencional de la conexión. Siguiendo este modelo, el diseño de la aplicación cliente (sensor o puerta de enlace) se hace para funcionar completamente con una base de datos local embebida. Las operaciones de lectura y escritura se llevan a cabo en primer lugar con este almacenamiento local, lo que asegura una respuesta instantánea y un uso fluido sin interrupciones debidas a la espera de la red. (Richardson & Amundsen, 2013).

Un administrador de sincronización en segundo plano permite que se logre la tolerancia a la intermitencia. Cuando el dispositivo detecta conectividad, inicia un proceso de replicación que vacía la cola de mensajes acumulados hacia el servidor central. Este procedimiento tiene que ser inteligente, utilizando tácticas de "retroceso exponencial" (exponential back off) para los reintentos de conexión; de este modo, se evita agotar la batería del aparato o saturar la red tras repetidos intentos fallidos durante una larga interrupción del servicio (HiveMQ, 2025).

En el contexto agrícola este patrón resulta ser necesario porque la cobertura celular en zonas rurales es irregular. La solución propuesta emplea colas persistentes en el gateway lo que es capaz de guardar hasta una hora de telemetría detallada. Así aunque una tormenta degrade la señal o un corte de energía afecte la torre de comunicaciones, el seguimiento de las variables ambientales no se interrumpe, los vacíos en los gráficos históricos se completan de forma automática apenas se restablece el enlace (Flores Cruz & López Ortiz, 2023).

2.3.2.6 Validación y control de calidad de datos en el ingreso

La validación de datos en el punto de ingreso actúa como unos cortafuegos semánticos para el sistema, el backend, antes de continuar con cualquier mensaje, verifica que el esquema de datos

se cumpla rigurosamente, garantizando que campos obligatorios como la marca de tiempo y el ID del dispositivo estén presentes y correctamente estructurados. Los registros que no se ajustan a la estructura JSON requerida son rechazados de inmediato, lo cual protege la base de datos contra la contaminación con información maliciosa o corrupta. (Timescale Inc., 2025).

La validación de los datos en el punto de ingreso funciona como un "firewall" semántico para el sistema. El backend valida que se cumpla a cabalidad el esquema de datos antes de almacenar cualquier mensaje, garantizando que existan y estén adecuadamente formados campos obligatorios como la marca de tiempo y el ID del dispositivo. Se descartan al instante los registros que no se adaptan a la estructura JSON requerida, lo que protege la base de datos de ser contaminada con información maliciosa o corrupta. (InfluxData Inc., 2025).

La normalización de las marcas de tiempo es otro aspecto crítico del control de calidad, como los dispositivos IoT pueden tener relojes que se desvían, es necesario que el servidor verifique la consistencia temporal de los mensajes recibidos. Con frecuencia se permite una tolerancia temporal (por ejemplo, de +/- cinco minutos) en relación con la hora del servidor; los datos que tienen fechas demasiado antiguas o futuras se marcan para ser revisados manualmente o se corrigen, asegurando la consistencia cronológica necesaria para el estudio de series temporales (Tanenbaum & van Steen, 2017).

2.3.2.7 Seguridad y trazabilidad durante la sincronización

La sincronización de datos a través de redes públicas expone la información a vulnerabilidades que deben ser mitigadas con un cifrado sólido. El uso obligatorio de TLS (Transport Layer Security) durante el tránsito asegura que los paquetes de sincronización sean ilegibles para terceros, protegiendo la confidencialidad de la telemetría y, crucialmente, las credenciales de autenticación, esta capa de seguridad es innegociable en sistemas modernos para prevenir ataques de intermediario (Man-in-the-Middle) que podrían inyectar datos falsos en el flujo de sincronización (Eclipse Foundation, 2025).

La Implementación de logs de La auditoría inmutable permite que las tareas de sincronización sean trazables. Los metadatos de origen, que incluyen el hash de integridad del contenido, la dirección IP y el certificado de cliente usado, se registran para cada lote de información

recibido, esto posibilita la reconstrucción de la sucesión precisa de los acontecimientos si hay una discrepancia de datos, lo que simplifica el diagnóstico técnico forense y la rendición de cuentas operativas acerca de quién cambió qué variable y cuándo (EMQ Technologies Co., Ltd., 2025).

Además, se Utilizan procedimientos de no repudio a través de firmas digitales en mensajes críticos de control. Cuando el servidor transmite una instrucción para activar el riego, esta es firmada criptográficamente. Esto posibilita que el controlador de campo confirme la autenticidad del mandato antes de llevarlo a cabo. Asimismo, el aparato confirma la ejecución, completando así el ciclo de sincronización de estado mediante una prueba comprobable. De la acción ejecutada en el entorno físico (Banks & Gupta, 2014).

2.3.2.8 Patrones event-driven y buenas prácticas en contextos agroambientales

Las arquitecturas orientadas a eventos (Event-Driven Architecture) son el estándar de facto para sistemas de monitoreo agroambiental debido a su capacidad de desacoplamiento y escalabilidad. En este patrón, los sensores funcionan como generadores de eventos puros que no tienen conocimiento de los consumidores finales. Conectar varios servicios y alertas, así como el almacenamiento y análisis, a un flujo de datos común mediante un bus de eventos asíncrono no afecta la eficacia de la red de sensores y simplifica el desarrollo natural del sistema (Michelson, 2006).

Separar lógicamente los temas de eventos es una práctica fundamental y efectiva para mostrar cómo está estructurado básicamente el cultivo. Crear una taxonomía de temas que sea clara (ej. finca/zona/tipo_sensor) permite aplicar políticas de Calidad de Servicio (QoS) diferenciadas. En consecuencia los eventos de latido (heartbeat) del sistema pueden enviarse con prioridad de tipo baja mientras que los de alarma de sequía reciben trato preferente y se asegura su persistencia en el broker (Date, 2019).

Por ultimo vigilar de forma proactiva el propio flujo de eventos (throughput) es la clave para para ajustar la operación. Detectar anomalías como un sensor ruidos que satura la red o un silencio inesperado que delata fallos a gran escala lo cual se logra analizando la tasa de eventos por segundo. Integrar estas métricas de salud del sistema en el mismo tablero de control

agronómico proporciona a los operadores una visión holística tanto del cultivo como de la infraestructura tecnológica que lo sustenta (Chugá Rosero, 2022).

2.3.3 Metodología de desarrollo, Prototipos

El propósito principal del prototipo en la ingeniería de software contemporánea es navegar por la incertidumbre técnica. Esta táctica evolutiva elude las definiciones estrictas al principio, ya que supone que en sistemas complejos los requerimientos reales surgen a medida que se va construyendo. Al materializar una versión preliminar funcional, el equipo puede contrastar sus hipótesis con la realidad física del hardware, reduciendo el riesgo de entregar un producto final que no soporta las restricciones operativas del entorno (Pressman & Maxim, 2019).

En términos operativos, la estrategia se divide en dos vías: construir modelos "desechables" para aclarar preguntas específicas o desarrollar prototipos "evolutivos" que se perfeccionan hasta llegar a ser de calidad de producción, este ciclo de iteración veloz posibilita la identificación de errores arquitectónicos antes de que resulten demasiado costosos. En telemetría, por ejemplo, esto implica validar la estabilidad del gateway antes de invertir horas en interfaces gráficas, garantizando que los cimientos de la comunicación sean sólidos desde el inicio (Sommerville, 2016).

Sin embargo, el uso de modelos tempranos conlleva riesgos que deben gestionarse, como la acumulación de "deuda técnica". Un prototipo exitoso permite identificar cuellos de botella como una latencia excesiva, pero requiere una reingeniería rigurosa antes del despliegue final es crucial resistir la tentación de poner en producción un código "parcheado", garantizando que la solución satisfaga los estándares necesarios de mantenibilidad y seguridad para una operación continua (Boehm, 1988).

Se utiliza un enfoque incremental validado en campo para este estudio. La primera iteración se concentra únicamente en establecer un canal confiable entre los sensores y el bróker MQTT, sin considerar configuraciones que no toleran fallas. Los pasos siguientes comprenden el almacenamiento histórico y la visualización, lo que produce evidencia empírica en cada etapa, comprenden el almacenamiento histórico y la visualización, lo que produce evidencia empírica en cada etapa de este modo, los parámetros de sincronización se adaptan en función de datos reales del vivero y no de conjeturas teóricas sobre diseño (Larman & Basili, 2003).

2.4 Conclusiones del marco teórico

El protocolo MQTT se valida como la mejor arquitectura para ambientes agroambientales con recursos escasos, según el análisis teórico. Su construcción liviana y la aplicación de sistemas como la Calidad de Servicio (QoS) posibilitan que los impactos de la intermitencia en la red se reduzcan, garantizando así que la telemetría esencial llegue a su destino, asimismo, la integridad del canal se asegura a través de la implementación de capas de seguridad apoyadas en el cifrado TLS y en las listas de control de acceso (ACL) lo que permite superar las debilidades propias de los esquemas tradicionales de transmisión abierta.

Se concluye que sincronizar de forma efectiva en sistemas distribuidos exige algo más de replicar datos lo que requiere un control riguroso sustentado en métricas de completitud consistencia y latencia. Para preservar la trazabilidad operativa resulta clave adoptar estrategia “offline-first” basadas en colas persistentes y en el manejo de conflictos. La intermitencia propia del campo causaría, sin estos mecanismos, brechas de información inaceptables, lo que pondría en riesgo la confiabilidad de las series históricas requeridas para decidir, la integración del estándar MQTT con políticas de sincronización orientadas a eventos genera una sinergia técnica capaz de transformar lecturas dispersas en información agronómica confiable.

Esta fusión tecnológica permite auditar el flujo de datos en tiempo real, estableciendo garantías medibles de entrega y orden cronológico. Al blindar la comunicación contra fallos de red, el sistema asegura que las alertas y los análisis históricos reflejen fielmente la realidad física del vivero, cerrando la brecha entre la captura del sensor y la acción del operador.

Por último, la selección de la metodología de prototipos se basa en la exigencia de comprobar los requerimientos empíricamente en un contexto físico incierto, el desarrollo de manera iterativa posibilita comprobar la estructura de tópicos y los esquemas de seguridad en situaciones reales antes de que se produzca el despliegue a gran escala, está la estrategia disminuye de manera importante el riesgo técnico, porque cada mejora en la funcionalidad ofrece pruebas concretas para modificar la arquitectura, lo que garantiza que la solución final sea sólida y esté en consonancia con las limitaciones operativas del proyecto

CAPÍTULO III

3. MARCO INVESTIGATIVO

3.1 Introducción

En este proyecto el marco investigativo integra ordena y proyecta las decisiones metodológicas necesarias para analizar una problemática real del registro manual y la sincronización de datos en el vivero de cacao de la ULEAM Extensión El Carmen. En este capítulo se expone el tipo de estudio realizado como el enfoque adoptado la definición de la población y muestra, así como las herramientas empleadas para recolectar y procesar la información.

La investigación se plantea con carácter aplicado y de campo ya que se desarrolla directamente en un contexto operativo del vivero y combina técnicas cuantitativas como encuestas estructuradas, con técnicas cualitativas, como entrevistas abiertas con el fin de obtener una visión amplia y profunda de la situación actual. Se describe quienes conforman la población de interés, los criterios utilizados para seleccionar la muestra y el procedimiento estadístico que sustenta dicho tamaño muestral garantizando que los resultados sean representativos del conjunto de actores que intervienen en los procesos de monitoreo y registros de los datos.

Además, se detallan los instrumentos diseñados para la recolección de información como la forma en que se aplican y las estrategias utilizadas para el análisis de los datos tanto numéricos como narrativos, de modo que los hallazgos obtenidos permitan evaluar de manera objetiva las limitaciones del sistema manual y fundamentar la pertinencia del sistema informático con MQTT orientado a la sincronización automatizada de registros den el vivero.

3.2 Tipos de investigación

3.2.1 Investigación de campo

La investigación de campo se define como aquella que se realiza recopilando información primaria directamente en el contexto natural donde ocurren los fenómenos objeto de estudio. De acuerdo con Tarrillo Saldaña et al. (2024), en su obra *Metodología de la investigación una mirada global*, la investigación de campo constituye un proceso sistemático de obtención de

conocimiento mediante la recolección de datos en el ambiente real, sin manipulación artificial de variables por parte del investigador.

Como el estudio buscaba obtener la información directamente donde ocurren los hechos, se optó por una investigación de campo en el vivero de cacao de la Extensión El Carmen de la ULEAM. Estos datos se recogieron en el mismo lugar donde se llevaban a cabo los registros de forma manual, lo que permitió observar de primera mano las fallas al consolidar la información, cuanto tardaba en responder el personal técnico y en qué condiciones de conectividad se trabaja en ese entorno rural. En la etapa de diagnóstico se realizaron visitas al vivero, empleando observación directa, registros fotográficos y diarios de campo para documentar la situación existente. Posteriormente durante esta implementación del prototipo basado en el protocolo MQTT se continuó recolectando datos en el sitio mediante mediciones de latencia y pruebas de sincronización esto permitió evaluar el comportamiento real de la comunicación bajo las condiciones operativas del vivero.

3.2.2 Investigación descriptiva

El propósito de la investigación descriptiva es exponer las propiedades, atributos y rasgos esenciales del fenómeno que se estudia, describiendo su funcionamiento, composición y estructura sin pretender detallar las causas que subyacen a su existencia. Hernández-Florez (2025) sostiene que este tipo de investigación tiene como objetivo responder preguntas tales como qué sucede, cómo ocurre y cuáles son sus características, mediante el empleo de técnicas sistemáticas como la observación, las entrevistas y la revisión documental, que permiten recopilar información exhaustiva sobre el objeto de estudio.

Para este proyecto, se utiliza la investigación descriptiva en el período de diagnóstico con el objetivo de describir minuciosamente la situación presente de los procesos de registro y administración de información en el vivero. A través de observaciones directas, la manera en que funcionan hoy en día las bitácoras manuales, el tiempo que toma consolidar los datos hojas de cálculo, la periodicidad de errores y omisiones en los registros y formatos empleados para almacenar información.

3.2.3 Investigación cuantitativa

La investigación cuantitativa utiliza la recopilación y el análisis de datos numéricos para responder preguntas de investigación y validar hipótesis mediante procedimientos estadísticos y matemáticos. Hernández-Florez (2025) señala que la ruta cuantitativa se caracteriza por un enfoque deductivo, en el que el investigador parte de teorías generales y las somete a prueba midiendo variables específicas en poblaciones o muestras determinadas. Este tipo de estudio destaca por su objetividad, su capacidad de replicación y la posibilidad de extender los resultados a contextos similares, reduciendo la interpretación subjetiva mediante la aplicación rigurosa de métodos estadísticos que permiten realizar análisis descriptivos e inferenciales sobre las relaciones entre variables.

Se empleó la investigación cuantitativa para evaluar el sistema informático con MQTT y calcular índices clave de rendimiento que reflejaran la eficacia y operatividad de la solución propuesta. Entre estos datos numéricos que se recogieron estuvieron la cantidad de los registros que se sincronizaron sin errores las tasas de duplicación y de omisión y el tiempo que tardaba un mensaje desde que llegaba hasta que quedaba guardado en la base de datos central. Estos valores se obtuvieron consultando la propia base de datos y la bitácora de auditoría del sistema lo que hizo posible comparar como se comportaba el sistema antes y después de la implementación. Para el análisis se calcularon procesos y porcentajes y con estos se verificó de manera cuantitativa si el prototipo cumplía las metas técnicas de confiabilidad, disponibilidad y mejora en la sincronización de la información del vivero.

3.3 Técnicas de recolección de datos

3.3.1 Observación directa

La observación directa es una técnica de recolección de datos que implica la supervisión atenta y sistemática de un fenómeno o comportamiento en su ambiente natural, permitiendo al investigador obtener información de primera mano sin manipulación artificial del contexto. Hernández Sampieri et al. (2014) señalan que esta técnica consiste en el registro sistemático, válido y confiable de comportamientos y situaciones observables, a través de un conjunto de categorías y subcategorías definidas previamente.

En este proyecto integrador, en la fase de diagnóstico se utilizó la observación directa para registrar de manera sistemática el funcionamiento actual de los procesos manuales de registro en el vivero de cacao. Los investigadores fueron al vivero en momentos oportunos con el propósito de observar las actividades de recolección de datos sin intervenir, usarán diarios de campo y documentación fotográfica para registrar los métodos manuales, el tiempo que toman, la interacción entre el personal técnico y los alumnos, así como las dificultades que surgen durante la recopilación de información.

3.3.2 Encuesta

La encuesta es una técnica de recolección de datos que permite obtener de manera rápida y efectiva información cualitativa y cuantitativa, posibilitando la comparación sistemática de respuestas entre un gran número de personas; resulta particularmente útil cuando se necesita medir variables como la percepción, el cambio de conductas o la satisfacción en muestras amplias. Martínez (2024) señala que la encuesta se caracteriza por el uso de cuestionarios estructurados que estandarizan la recolección de la información y facilitan su posterior análisis estadístico.

Se aplicó una encuesta en la fase de diagnóstico a los estudiantes de la carrera de Ingeniería en Agropecuaria que colaboran en las actividades de monitoreo y registro del vivero, con el propósito de conocer su percepción sobre los procesos manuales vigentes. El cuestionario estuvo formado por treinta preguntas estructuradas agrupadas en cinco dimensiones lo cual aumenta la eficacia del monitoreo actual, las dificultades que trae la gestión manual de los datos, su impacto en el trabajo diario y la carga administrativa el estado actual de la infraestructura disponible y el perfil tecnológico de quienes respondieron. En total se recogieron ochenta y nueve encuestas y sus datos se analizaron con estadística descriptiva, calculando frecuencias y porcentajes. Así se logró mostrar con cifras las deficiencias del registro manual y respaldar la necesidad de la solución que se propone.

3.3.3 Entrevista

La entrevista es una técnica cualitativa que se desarrolla mediante un diálogo interpersonal directo entre el investigador y los sujetos de estudio, y permite profundizar en las experiencias,

percepciones y vivencias de los participantes, facilitando la obtención de información detallada y contextualizada que no puede captarse únicamente mediante instrumentos cuantitativos. Hernández Sampieri et al. (2014) señalan que la entrevista es una reunión para conversar e intercambiar información entre el entrevistador y el entrevistado, mediante preguntas y respuestas que permiten construir significados respecto de un tema determinado.

En la fase de diagnóstico se recurrió a una entrevista semiestructurada para obtener información cualitativa más detallada sobre cómo funciona el vivero. Se entrevistó al magister Pedro Nivelá, coordinador y responsable del Vivero Lastenia vera, elegido como un informante clave por su experiencia directa en la gestión del cultivo. El guion incluyó trece preguntas abiertas, enfocadas en conocer los umbrales críticos de temperatura y humedad para las plántulas de cacao pues las dificultades que genera el registro manual de la información, la falta de los datos históricos organizados y las limitaciones operáticas del vivero. Las respuestas se grabaron y transcribieron para su análisis cualitativo, mediante la identificación de temas recurrentes y patrones de opinión que aportaron evidencia narrativa complementaria a los resultados obtenidos en la encuesta.

3.4 Fuentes de información de datos

3.4.1 Fuentes primarias

Las fuentes primarias son aquellas de las que se obtiene información directamente de las personas o del lugar donde se realiza el estudio. En este proyecto integrador, las principales fuentes primarias fueron los 89 estudiantes de Ingeniería Agropecuaria que participan en las actividades de monitoreo y registro del vivero, quienes brindaron información a través de una encuesta. También se contó con la participación del magister Pedro Nivelá quien es el coordinador del vivero “Lastenia Vera” a quien se le realizó una entrevista para poder conocer de primera mano cómo se llevan a cabo estas actividades. Además se realizó una observación directa en el vivero, lo que permitió conocer cómo se llevan a cabo actualmente los registros y con qué dificultades se presentan durante este proceso.

3.4.1.1 Encuestas

La encuesta es una técnica de recolección de datos que emplea procedimientos bien estructurados y estandarizados para la recolección de información lo que bien permite analizar de manera sistemática las características y variables del fenómeno objeto de estudio. De acuerdo con Sánchez y Sánchez y Hernández Olvera (2024) estas encuestas se distinguen por incluir cuestionarios estructurados con preguntas elaboradas de forma cautelosa y sistemática posibilitando una recopilación estandarizada de datos

En este proyecto integrador, la encuesta se aplicó a los estudiantes de la carrera de Ingeniería en Agropecuaria que colaboran directamente en las tareas de monitoreo y el mantenimiento como registro de las plantas en el vivero de cacao, por ser quienes interactúan de forma directa con los procedimientos de captura manual de datos y conocen sus fallos y limitaciones operativas. En la fase de diagnóstico se buscó conocer su percepción sobre las dificultades del registro manual, el tiempo invertido en consolidar los datos y la frecuencia de errores luego posteriormente tras la puesta en marcha del prototipo basado en MQTT se aplicó nuevamente una encuesta con escala de Likert para medir el impacto en la facilidad de uso, la reducción de errores y tiempo como la disponibilidad de información para la toma de decisiones. Los datos se analizaron mediante estadística descriptiva (frecuencias, porcentajes y medidas de tendencia central), aportando evidencia numérica del impacto de la solución.

3.4.1.2 Entrevista

La entrevista es una técnica cualitativa de investigación que emplea un diálogo guiado entre el investigador y uno o más sujetos para recopilar información detallada sobre sus vivencias, puntos de vista y perspectivas respecto a un asunto concreto. De acuerdo con ATLAS.ti (2025) la entrevista posibilita recoger datos ricos en detalles, expresados con palabras, ideas y significados, lo que permite examinar las perspectivas individuales de manera sensible al contexto; según el grado de rigidez de las preguntas, puede ser estructurada, semiestructurada o no estructurada.

En este proyecto integrador, se realizó una entrevista semiestructurada al magister Pedro Nivelá, coordinador y responsable del vivero “Lastenia Vera” de la ULEAM Extensión El Carmen, por su entendimiento de los requisitos técnicos y académicos de este tipo de proyectos

y de los retos operativos del vivero de cacao. La entrevista se orientó a conocer el aporte académico y operativo de la arquitectura basada en MQTT y los criterios de diseño de un proyecto integrador. Las respuestas se grabaron y transcribieron para su análisis cualitativo, identificando temas recurrentes y patrones de opinión que aportaron evidencia narrativa a la solidez académica y técnica de la propuesta.

3.5 Estrategia operacional para la recolección de datos

3.5.1 Población

La población, también denominada universo, es el conjunto finito o infinito de elementos que comparten características comunes y sobre los cuales se pretende realizar la investigación. De acuerdo con Arias (2012), la población está constituida por elementos o unidades que poseen características comunes y que se encuentran delimitados de acuerdo con el problema y los objetivos del estudio.

La población de este proyecto integrador estuvo conformada por 89 estudiantes de la carrera de Ingeniería en Agropecuaria de la Extensión El Carmen de la ULEAM que colaboraban en las tareas de monitoreo, mantenimiento y registro de las plantas de cacao. Estos estudiantes fueron considerados porque participaban directamente en las actividades del vivero y conocían los procedimientos actuales de captura manual de datos, así como las dificultades relacionadas con la consolidación de la información. La población fue delimitada con base en la relación directa de los estudiantes con las actividades operativas del vivero y con el problema objeto de estudio.

3.5.2 Muestra

En el campo de la investigación, la muestra se entiende como un subconjunto de la población seleccionado para participar en el estudio. Arias (2012) explica que la muestra representa una parte de la población y se determina de acuerdo con las características y los objetivos de la investigación. En este proyecto, aunque el cálculo determinó un tamaño mínimo de 72 participantes, se encuestó a los 89 estudiantes que conformaban la población total, por esta razón, se encuestó a toda la población y el levantamiento tuvo carácter censal.

Para el cálculo de la muestra se consideró una población de 89 estudiantes, un nivel de confianza del 95%, un margen de error del 5% y una proporción esperada de $p = 0.5$. Debido a que $q = 1 - p$, se obtuvo $q = 0.5$. Se utilizaron estos valores porque, al no contar con una proporción previa conocida, representan el escenario de máxima variabilidad.

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

$$n = \frac{89 * 1.96^2 * 0.5 * 0.5}{0.05^2 * (89 - 1) + 1.96^2 * 0.5 * 0.5}$$

$$n = \frac{89 * 3.8416 * 0.5 * 0.5}{0.0025 * 88 + 3.8416 * 0.5 * 0.5}$$

$$n = \frac{341.9024 * 0.25}{0.22 + 0.9604}$$

$$n = \frac{85,4756}{1.1804}$$

$$n = 72$$

El cálculo determinó un tamaño mínimo de muestra de 72 participantes. Sin embargo, se obtuvieron 89 respuestas, correspondientes a la totalidad de los estudiantes que conformaban la población. Por lo tanto, el levantamiento fue censal, ya que se encuestó al total de la población definida.

3.5.3 Análisis de las herramientas de recolección de datos a utilizar

3.5.3.1 Encuesta

La encuesta aplicada en este proyecto consta de 30 preguntas estructuradas y diseñadas para evaluar la percepción y experiencia del personal del vivero de cacao respecto a los procesos actuales de monitoreo ambiental, registro de información y gestión operativa de las plántulas de *Theobroma* (el cacao). El cuestionario se encuentra organizado en cinco dimensiones

temáticas principales como: la eficacia del proceso actual de monitoreo y los conocimientos del personal (preguntas 1 al 6) como los problemas derivados de la gestión manual de los datos (preguntas de la 7 a 15), el impacto operativo y la carga administrativa (preguntas 16 al 21), la condición actual de la infraestructura y sus recursos (preguntas 22 a 27) y el perfil tecnológico y las fuentes de información para decisiones (preguntas 28 a 30).

3.5.3.2 Entrevista

La entrevista que se estructuró y aplicó constó de 13 preguntas abiertas dirigidas al magíster Pedro Nivelá, coordinador y responsable técnico del vivero de cacao del campus Lastenia Vera. El objetivo principal de este instrumento fue recopilar información especializada sobre los procesos críticos de gestión del vivero, los umbrales de decisión para las intervenciones fitosanitarias y de riego, las restricciones operativas derivadas del registro manual de información y las expectativas respecto a la implementación de soluciones tecnológicas basadas en la sincronización automatizada de datos mediante el protocolo MQTT.

3.5.3.3 Estructura de los instrumentos de recolección de datos aplicados

3.5.3.3.1 Estructura de la entrevista

La entrevista se realizó de manera presencial e individual y estuvo dirigida al magíster Pedro Nivelá, coordinador y responsable técnico del vivero de cacao. Para su aplicación se utilizó una guía semiestructurada formada por trece preguntas abiertas. Las preguntas estuvieron relacionadas con temas como la temperatura y humedad necesarias para las plántulas de cacao, las dificultades que se presentan al registrar la información de forma manual, la falta de datos históricos organizados y algunas de las limitaciones que existen en las actividades del vivero.

Durante la entrevista se tuvo cierta flexibilidad en el orden de las preguntas, permitiendo realizar preguntas adicionales cuando fue necesario para obtener mayor información sobre algún tema. Las respuestas fueron registradas mediante una grabación de audio, previo consentimiento del entrevistado, y posteriormente fueron transcritas para facilitar su revisión y análisis. La guía utilizada para la entrevista se encuentra en el Anexo D.

3.5.3.3.2 Estructura de la encuesta

La encuesta se aplicó de forma digital mediante un formulario en línea, que fue respondido directamente por los participantes. El cuestionario estuvo compuesto por treinta preguntas relacionadas con diferentes aspectos del monitoreo y registro de información en el vivero. Entre los temas considerados estuvieron el funcionamiento del proceso de monitoreo actual, las dificultades que genera el manejo manual de los datos, el impacto que esto tiene en las actividades y en la carga de trabajo, las condiciones de la infraestructura disponible y el nivel de conocimiento y uso de herramientas tecnológicas por parte de los participantes.

El cuestionario incluyó preguntas de opción única y otras afirmaciones que fueron valoradas mediante una escala de Likert de cinco niveles. Esto permitió conocer y comparar las opiniones de los participantes sobre los diferentes aspectos evaluados. La encuesta fue aplicada de manera anónima y la participación fue voluntaria. El cuestionario completo se encuentra en el Anexo E.

3.5.4 Plan de recolección de datos

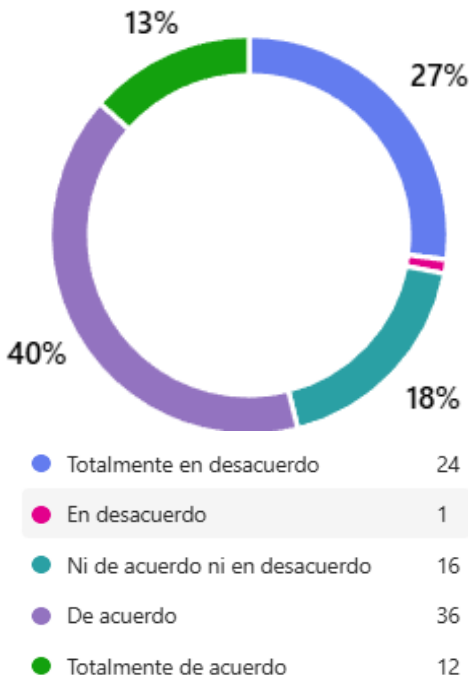
Tabla 1 Plan de recolección de datos

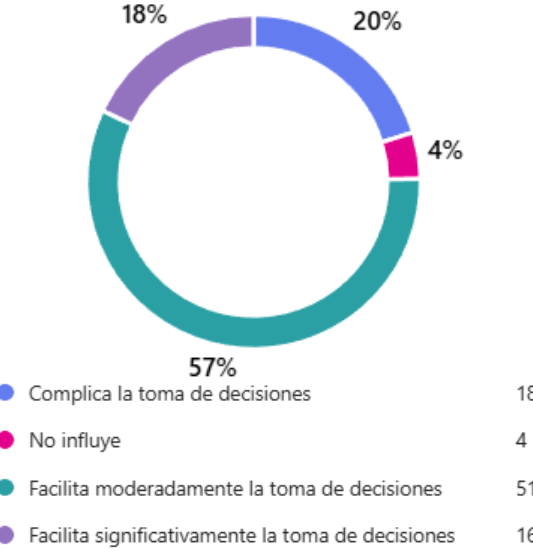
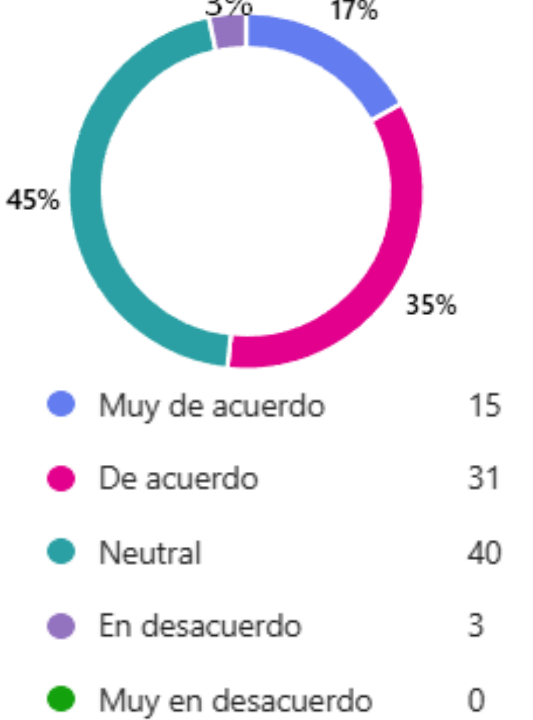
ACTIVIDAD	RESPONSABLES	FECHA
Ejecución de encuestas	Dereck Castro, David Guerrero, Bryan López, Anderson Plua, Kevin Rodríguez, Antony Zambrano.	22/02/2026 – 15/04/2026
Realización de entrevista	Dereck Castro, David Guerrero, Bryan López, Anderson Plua, Kevin Rodríguez, Antony Zambrano.	15/04/2026 – 11:30

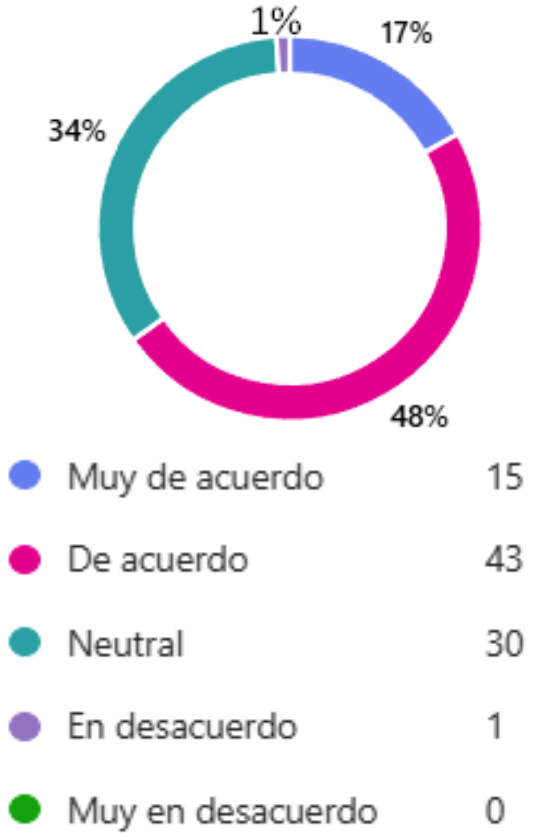
3.6 Análisis y presentación de resultados

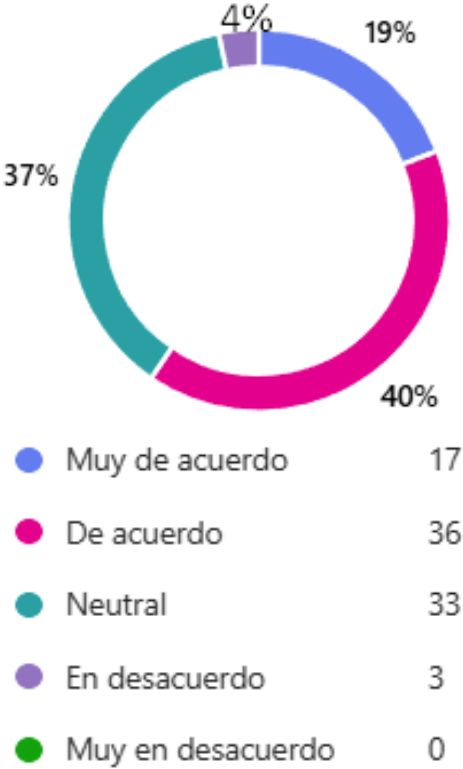
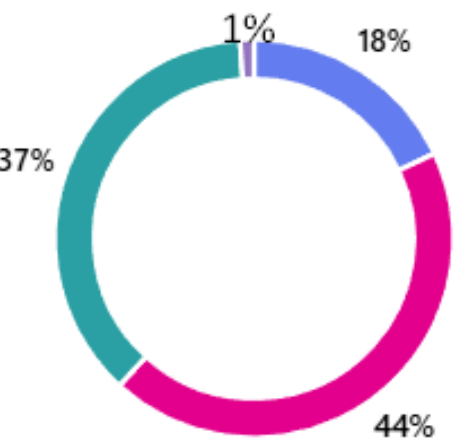
3.6.1 Presentación y descripción de los resultados obtenidos

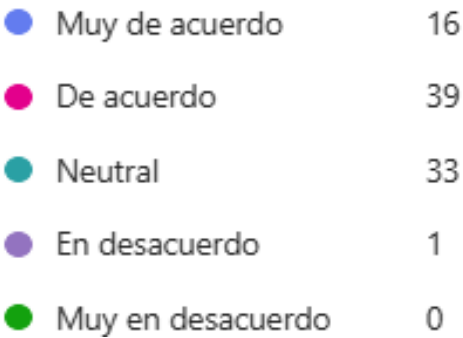
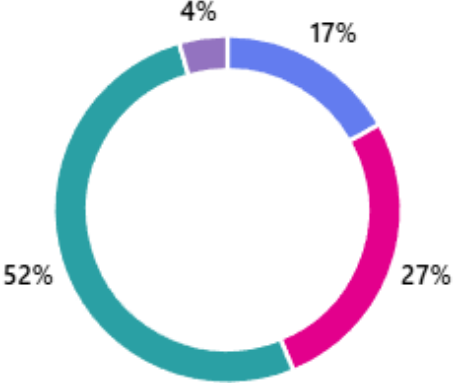
Tabla 2 Resultados de la encuesta dirigida a los estudiantes de Ingeniería en Agropecuaria

PREGUNTA	GRAFICO	INTERPRETACIÓN
Pregunta 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?	 Ilustración 3 Distribución de respuestas de la Pregunta 1 – Oportunidad de la información del monitoreo actual	Los resultados muestran que algo más de la mitad de los encuestados (54%) considera que el proceso actual entrega información oportuna, mientras que el 28% está en desacuerdo y con un 18% se mantiene neutral. Aunque predomina la valoración positiva, casi la mitad del grupo no afirma que la información llegue a tiempo, lo que revela que la percepción de oportunidad no es homogénea. Este dato cobra sentido al contrastarlo con otras preguntas del mismo instrumento el cual el 65% reconoce que invertir tiempo considerable transcribiendo datos y el 62% afirma que la falta de un repositorio centralizado genera inconsistencias la información llega pero a costa de un esfuerzo manual que el sistema propuesto elimina

PREGUNTA	GRAFICO	INTERPRETACIÓN																		
Pregunta 8: ¿De qué manera considera que influye el acceso a información ambiental en tiempo real sobre la toma de decisiones fitosanitarias en el vivero?	 <table border="1"> <thead> <tr> <th>Respuesta</th> <th>Porcentaje</th> <th>Cantidad</th> </tr> </thead> <tbody> <tr> <td>Complica la toma de decisiones</td> <td>20%</td> <td>18</td> </tr> <tr> <td>No influye</td> <td>4%</td> <td>4</td> </tr> <tr> <td>Facilita moderadamente la toma de decisiones</td> <td>57%</td> <td>51</td> </tr> <tr> <td>Facilita significativamente la toma de decisiones</td> <td>18%</td> <td>16</td> </tr> </tbody> </table> <i>Ilustración 4 Distribución de respuestas de la Pregunta 8 – Acceso a información en tiempo real para decisiones fitosanitarias</i>	Respuesta	Porcentaje	Cantidad	Complica la toma de decisiones	20%	18	No influye	4%	4	Facilita moderadamente la toma de decisiones	57%	51	Facilita significativamente la toma de decisiones	18%	16	La mayoría de los encuestados reconoce que contar con información ambiental en tiempo real facilita las decisiones fitosanitarias y lo ve como algo que mejora bastante la capacidad de respuesta cuando las condiciones son adversas. Esa percepción respalda la justificación del proyecto ya que confirma que los usuarios finales valoran tener los datos disponibles de inmediato para actual a tiempo.			
Respuesta	Porcentaje	Cantidad																		
Complica la toma de decisiones	20%	18																		
No influye	4%	4																		
Facilita moderadamente la toma de decisiones	57%	51																		
Facilita significativamente la toma de decisiones	18%	16																		
Pregunta 12: ¿Cree usted que el registro manual de datos de riego y clima aumenta el riesgo de cometer errores en la información recolectada?	 <table border="1"> <thead> <tr> <th>Respuesta</th> <th>Porcentaje</th> <th>Cantidad</th> </tr> </thead> <tbody> <tr> <td>Muy de acuerdo</td> <td>17%</td> <td>15</td> </tr> <tr> <td>De acuerdo</td> <td>35%</td> <td>31</td> </tr> <tr> <td>Neutral</td> <td>45%</td> <td>40</td> </tr> <tr> <td>En desacuerdo</td> <td>3%</td> <td>3</td> </tr> <tr> <td>Muy en desacuerdo</td> <td>0%</td> <td>0</td> </tr> </tbody> </table> <i>Ilustración 5 Distribución de respuestas de la Pregunta 12 – Riesgo de errores por registro manual</i>	Respuesta	Porcentaje	Cantidad	Muy de acuerdo	17%	15	De acuerdo	35%	31	Neutral	45%	40	En desacuerdo	3%	3	Muy en desacuerdo	0%	0	A los encuestados les preocupa que tan confiable es el registro manual ellos reconocen que hacerlo a mano aumenta las posibilidades de que haya errores al transcribir datos que se omiten o registros que pueden llegar a duplicarse. Este resultado respalda uno de los problemas principales que se detectaron en el diagnóstico inicial y sustenta la necesidad de un sistema automatizado que quite esos pasos intermedios de transcripción a mano. La arquitectura propuesta con sensores IoT conectados mediante MQTT al repositorio central garantiza la integridad de los datos desde su origen, eliminando la intervención manual que actualmente compromete la calidad de la información.
Respuesta	Porcentaje	Cantidad																		
Muy de acuerdo	17%	15																		
De acuerdo	35%	31																		
Neutral	45%	40																		
En desacuerdo	3%	3																		
Muy en desacuerdo	0%	0																		

PREGUNTA	GRAFICO	INTERPRETACIÓN
Pregunta 16: Invierto una cantidad considerable de tiempo pasando datos de campo "a limpio" en formatos digitales para mis reportes.	 <i>Ilustración 6 Distribución de respuestas de la Pregunta 16 – Tiempo dedicado a la transcripción manual</i>	Estos resultados revelan que los estudiantes gastan bastante tiempo pasando a limpio los datos que recogen a mano en el campo una tarea que no aporta nada a su formación y que les quita horas que podrían dedicar a aprender de forma práctica sobre el cultivo. Este trabajo administrativo de más deja ver la ineficiencia que tiene el proceso actual y sustenta la idea de automatizar la captura y la sincronización de los datos con el protocolo MQTT. Al eliminar la necesidad de transcripción manual, el sistema propuesto libera tiempo del personal para actividades de mayor valor técnico y mejora la experiencia formativa de los estudiantes.

PREGUNTA	GRAFICO	INTERPRETACIÓN
Pregunta 17: El canal de comunicación actual para reportar emergencias en las parcelas es inmediato y efectivo.	 <i>Ilustración 7 Distribución de respuestas de la Pregunta 17 Efectividad del canal de comunicación actual</i>	Las respuestas muestran opiniones divididas sobre qué tan efectivos son los canales de comunicación que usan hoy para reportar situaciones críticas ya que buena parte de los encuestados se mantiene neutral o no está de acuerdo con lo que el sistema actual avise a tiempo. Esta debilidad al comunicar eventos urgentes es un riesgo operativo ya que puede provocar reacciones tardías frente a condiciones que ponen en peligro las plántulas. El sistema propuesto ya incluye un módulo de alertas automáticas mediante suscripciones MQTT a tópicos específicos avisa de inmediato al personal responsable cuando se detectan valores fuera de los rangos críticos, con lo que se acortan bastante los tiempos de respuesta.
Pregunta 18: La falta de un repositorio centralizado de datos genera confusión e inconsistencia en los informes finales míos o de mi grupo.		Los encuestados confirman que tener la información repartida en varios cuadernos o hojas de cálculo por separado genera inconsistencias y compila armar informes de grupo coherentes. Este resultado apunta directo al problema central de este proyecto hace falta un repositorio centralizado que funcione como única fuente confiable para todos los registros del vivero. La arquitectura propuesta con bróker MQTT centralizado y base de datos única resuelve este problema al garantizar que todos los datos capturados, independientemente del nodo de origen, se almacenen de forma estructurada en un repositorio único accesible para consulta por todos los usuarios autorizados.

PREGUNTA	GRAFICO	INTERPRETACIÓN
	 <i>Ilustración 8 Distribución de respuestas de la Pregunta 18 – Efecto de la falta de un repositorio centralizado</i>	
Pregunta 22: ¿Cuál es su opinión sobre la eficiencia y eficacia (sin errores) en el registro manual de los datos ambientales meteorológicos?	 <i>Ilustración 9 Distribución de respuestas de la Pregunta 22 – Eficiencia y eficacia del registro manual de datos</i>	En la pregunta 22 se consultó a los estudiantes su opinión sobre la eficiencia y eficacia del registro manual de los datos ambientales. Los resultados muestran una postura mayoritariamente neutral: el 52 % se ubicó en el punto medio, mientras que un 27 % y un 17 % se inclinaron hacia los extremos y apenas un 4% marcó la opción restante. Que la mitad del grupo no se decida ni por "eficiente" ni por "ineficiente" indica que los estudiantes conocen el método manual y lo dan por funcional, pero al mismo tiempo no lo respaldan del todo. Esa duda deja ver que perciben sus límites aunque no los expresen abiertamente. Este hallazgo respalda la necesidad de digitalizar la captura de datos con sensores IoT y transmisión por MQTT, un cambio que apunta a reducir los errores del proceso manual y a asegurar que ningún registro quede incompleto.

3.6.1.1 Análisis de entrevista realizada a Coordinador de Vivero Lastenia Vera

La entrevista fue aplicada al Ing. Pedro Nivelá, coordinador técnico del Vivero Lastenia Vera, con el objetivo de validar desde la perspectiva del experto de dominio las problemáticas identificadas en el Capítulo I respecto a la gestión manual de variables ambientales y

determinar los requerimientos técnicos para la implementación del sistema informático con protocolo MQTT propuesto.

Tabla 3 Análisis de la entrevista realizada al coordinador del Vivero Lastenia Vera

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?	"En la fase inicial desde la siembra, debemos controlar la temperatura; debe ser un poco alta para que pueda germinar. Hay un balance entre humedad y temperatura en el sustrato para que la semilla germine. Una vez que el cacao, que es tropical y subtropical, germina, debemos tener un rango de entre 28°C a 30°C y una humedad relativa no mayor al 75% para evitar problemas de plagas y enfermedades principalmente."	Hoy esos rangos los 28 a 30 °C y la humedad por debajo del 75 % que menciona el coordinador se controlan a puro criterio y observación, sin un registro que los respalde en el tiempo. Al no medirse de forma continua además de tampoco haber una manera de comprobar cada cuánto se salen de esos límites ni de sustentar con datos por qué esos y no otros. Justamente ahí encaja el sistema propuesto: al llevar la temperatura y la humedad al servidor en tiempo real, quedaría el registro que hoy falta y se podría avisar cuando los valores pasen del umbral que el experto señaló antes de que aparezcan las plagas y enfermedades que él mismo advierte.
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?	"La planta de cacao tiene mecanismos para regular procesos metabólicos, pero en un estado avanzado de falta de agua, se produce estrés hídrico. A simple vista podemos observar cuándo la planta necesita agua, pero sería ideal tener herramientas como sensores que lo determinen con precisión."	La carencia de instrumentos de medición precisos impide la detección temprana de condiciones que generan estrés hídrico en las plántulas. El mismo coordinador lo admite cuando dice que "a simple vista" se nota cuando la planta necesita agua pero que "sería ideal tener herramientas como sensores" que lo midan ya se ve, la planta lleva rato sufriendo así que por el riesgo termina llegando tarde. Con sensores conectados por MQTT enviando los datos al servidor de manera continua, esas señales podrían detectarse antes de que el daño sea visible y actuar a tiempo en lugar de reaccionar cuando ya hay plantas en mal estado.
3. ¿Cree que la falta de monitoreo automatizado de humedad y temperatura en tiempo real contribuye al estrés hídrico de las plantas de cacao? ¿Por qué?	"Claro. Tenemos muchas fluctuaciones en el clima de esta zona por la nubosidad, el viento y la heliofanía (horas de sol). Estas fluctuaciones producen estrés. Es necesario un monitoreo constante con equipos para saber cuándo aumenta la temperatura y aplicar estrategias como el micro-riego para evitar cambios bruscos que afectan el metabolismo."	Nivela lo confirma sin dudar y explica por qué que en esta zona el clima cambia mucho por la nubosidad el viento y la heliofanía y esas variaciones son las que estresan a la planta. Revisando a mano y de vez en cuando no se alcanza a captar esos cambios a tiempo, así que cuando se reacciona el daño ya empezó. Esto respalda directamente la razón de ser del proyecto: si los sensores envían la temperatura y la humedad al servidor de forma continua por MQTT, se puede saber en el momento cuándo sube la temperatura y aplicar a tiempo estrategias como el micro-riego que él mismo menciona, en lugar de esperar a que el cambio brusco afecte a las plántulas.
5. ¿Cuáles son los principales problemas que enfrenta el Vivero	"El problema más grave es la disponibilidad hídrica. No tenemos una fuente primaria	Nivela apunta al agua como el mayor problema: sin un río cercano, el vivero improvisa con pozos artesanales y dos cisternas, y bombear esa

PREGUNTA	RESPUESTA	INTERPRETACIÓN
Lastenia relacionados con la gestión del riego y demás variables ambientales en el cacao?	como un río, así que improvisamos con pozos artesanales y dos cisternas. Esto eleva los costos por el uso de combustible para el bombeo, el cual ha subido de precio últimamente."	agua cuesta cada vez más porque el combustible ha subido. Ese peso económico es justamente lo que le da sentido a monitorear las variables: si se sabe con datos reales cuándo la planta necesita agua y cuándo no, se riega solo lo necesario y se deja de gastar combustible bombeando de más.
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?	"Tenemos una mini estación meteorológica en la carrera agropecuaria. En base a esos datos, se hace una proyección y análisis para el tema ambiental."	Lo que responde Nivelá es corto y revela algo: menciona que en la carrera agropecuaria hay una mini estación meteorológica con la que se hacen proyecciones, pero no describe ningún paso concreto de cómo los estudiantes toman el dato en la parcela ni cómo llega al repositorio. Esa misma vaguedad confirma el problema de fondo: no existe un procedimiento claro y ordenado para capturar y consolidar la información. El sistema propuesto llenaría ese vacío, porque los sensores tomarían el dato, lo enviarían al bróker y quedaría guardado de una vez en el servidor, sin pasos manuales de por medio y sabiendo siempre de qué estación vino cada medición.
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?	"Los datos históricos sirven para análisis estadísticos de probabilidad. Las fluctuaciones de lluvias permiten prever la incidencia de plagas. Aunque con el cambio climático los datos ya no son tan constantes, siempre debemos estar actualizando la información."	El experto reconoce explícitamente el valor de los datos históricos para análisis predictivos de incidencia de plagas y la necesidad de actualización continua de la información ante la variabilidad climática creciente. Esta declaración valida directamente uno de los objetivos del proyecto contar con un histórico ordenado de las mediciones. Él mismo explica que las fluctuaciones de lluvias permiten prever cuándo van a aparecer las plagas, pero que con el cambio climático los datos ya no son tan constantes y hay que actualizarlos siempre. Hoy no existe ese registro acumulado, así que no se puede cruzar el clima de fechas pasadas con los brotes que hubo. Al guardar cada lectura de forma continua y estructurada, el sistema propuesto permitiría revisar esos patrones más adelante y anticipar las épocas de mayor riesgo, que es justo el análisis que el coordinador señala como útil.
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?	"Para evitar la mortalidad, seleccionamos bien la semilla y hacemos controles fitosanitarios preventivos. Manejamos una bitácora manual donde registramos los tratamientos. No tenemos un software o sensores actualmente por el tema de la inversión que requiere el sistema público; nos adaptamos a lo que tenemos."	Nivelá es claro: el vivero no lleva registros digitales, solo una bitácora manual donde anotan los tratamientos, y explica que no tienen software ni sensores por lo que cuesta ese sistema en lo público, así que "nos adaptamos a lo que tenemos". Esa respuesta confirma el problema central del proyecto lo cual no hay una memoria ordenada que permita mirar hacia atrás y ver qué funcionó o qué falló, y deja ver lo frágil que es depender de un cuaderno que se puede perder o quedar incompleto. Aquí encaja el sistema propuesto, pensado justamente con herramientas de bajo costo: sería una opción

PREGUNTA	RESPUESTA	INTERPRETACIÓN
		accesible para el presupuesto del vivero que guardaría los datos de forma ordenada y permanente, para poder consultarlos y analizarlos después.

3.6.2 Informe final del análisis de los datos

Con respecto al registro de variables ambientales (temperatura y humedad) en el vivero de cacao, se observa que los encuestados en las preguntas 16 y 18 hacen referencia a que invierten una cantidad considerable de tiempo transcribiendo datos de campo a formatos digitales y que la falta de un repositorio centralizado genera confusión e inconsistencia en los informes finales. En esa línea, al responder la pregunta 8, Nivela comenta que en la carrera agropecuaria hay una mini estación meteorológica, pero no explica ningún paso concreto sobre cómo los estudiantes registran los datos en el vivero ni cómo se juntan ambos lados. Eso deja ver que la captura y consolidación de datos va por partes, sin un orden claro, y por eso es fácil que se cometan errores al transcribir, se pierda información o se dupliquen registros, justo los datos que después sirven para decidir en el cultivo. Y como todo se hace a mano, se genera un trabajo administrativo de más que les quita tiempo a los estudiantes de lo que de verdad importa: su formación.

Por su parte, el entrevistado en las preguntas 2 y 3 confirma explícitamente que la falta de instrumentos de medición impide la detección temprana del estrés hídrico y que las fluctuaciones climáticas de la zona requieren un monitoreo constante con equipos especializados para aplicar estrategias correctivas oportunas. Esta coincidencia entre la percepción de los usuarios finales y la evaluación del experto técnico confirma que el proceso actual de monitoreo manual discontinuo es insuficiente para garantizar condiciones óptimas de crecimiento, generando intervenciones reactivas tardías cuando los síntomas de deterioro ya son visibles en las plantas.

En cuanto a la comunicación de eventos que necesitan atención inmediata, las respuestas a la pregunta 17 están divididas: buena parte de los encuestados se muestra neutral o en desacuerdo con que el canal actual sirva para reportar emergencias en las parcelas de forma rápida y efectiva. Nivela, al responder la pregunta 10, habla de que especialistas y estudiantes colaboran para resolver novedades, pero no menciona ninguna forma establecida de avisar cuando nadie

está físicamente en el vivero. Esto muestra que avisar de un problema depende de la buena voluntad de cada quien y no de un procedimiento fijo, lo que retrasa la respuesta ante situaciones que ponen en riesgo a las plántulas, sobre todo fuera de horario, cuando no hay personal en el campus Lastenia Vera.

Sobre la disponibilidad de datos históricos para revisar el pasado y decidir mejor, quienes respondieron la pregunta 18 confirman que la falta de un repositorio central genera confusión e inconsistencias en los informes, y en la pregunta 22 se ven opiniones repartidas sobre qué tan eficiente es el registro manual. Nivela, en las preguntas 9 y 13, reconoce que los datos históricos sirven para prever la incidencia de plagas, pero aclara que el vivero no tiene ni software ni sensores para llevar esos registros por lo que cuesta el sistema. Sin esa memoria guardada de forma digital, no se puede acumular conocimiento sobre cómo se comportan las variables ambientales en la zona, y todo termina dependiendo de cuadernos que se pueden perder o dañar.

CAPÍTULO IV

4. MARCO PROPOSITIVO

4.1 Introducción

En este capítulo expone el marco propositivo del proyecto en el que se detalla la solución tecnológica desarrollada para responder a la problemática identificada en el diagnóstico la falta de información climática oportuna y centralizada en el Vivero Lastenia de la ULEAM Extensión El Carmen ya que a partir de las faltantes evidenciadas en el registro manual de datos la latencia en la consolidación de los registros se plantea un sistema informático con arquitectura IoT orientado a la sincronización de datos ambientales mediante el protocolo MQTT.

En las secciones siguientes se describe la propuesta de manera integral partiendo de una visión general del producto y sus funcionalidades clave para así luego profundizar en los requerimientos del sistema el diseño como la arquitectura y las tecnologías que se empleó. Así mismo, se documenta el proceso de construcción del software bajo la metodología SCRUM organizado en sprints que reflejan la implementación incremental de cada componente, desde la infraestructura y la ingesta de datos hasta la seguridad de la información y su despliegue.

El objetivo de este capítulo es presentar de forma ordenada y verificable como la solución propuesta transforma el registro manual y disperso de las variables ambientales en un flujo de datos automatizado, integro y disponible de manera oportuna sentando las bases para la correcta evaluación de los resultados que se aborda en el capítulo anterior.

4.2 Descripción de la propuesta

La propuesta consiste en un sistema informático con arquitectura IoT orientado a la sincronización de datos del vivero de cacao mediante el protocolo MQTT. Este sistema articula varios componentes: nodos de captura de información (formularios móviles y/o sensores en campo), una parcela o gateway, un bróker MQTT el cual está encargado de la mensajería, un servicio del backend para ingestión y validación de datos, una base de datos central y un panel de visualización web para consulta y análisis por parte de docentes, técnicos y estudiantes.

Este sistema está dirigido a personal técnico responsable del vivero, a docentes que coordinan y conforman practicas formativas y a los estudiantes de Ingeniería en agropecuaria que participan en actividades de monitoreo y registro de variables ambientales y operativas. A través de una interfaz web y/o móvil, el cual los usuarios pueden ver y registrar lecturas de campo, consultar historiales, generar reportes y recibir alertas cuando se detectan condiciones críticas, reduciendo la dependencia de cuadernos y hojas de cálculos dispersas .

Con esta solución se busca disminuir la latencia entre la captura del dato y su disponibilidad para la toma de decisiones, reducir la tasa de omisiones y duplicidades en los registros, con esto se busca mejorar la trazabilidad de las intervenciones realizadas en las parcelas y facilitar la integración de nuevas fuentes de datos en el futuro, manteniendo una arquitectura escalable y tolerante a condiciones de conectividad intermitente propias del entorno rural y en el que esta.

4.3 Determinación de recursos

4.3.1 Humanos

Tabla 4 Recursos humanos del proyecto

Recurso	Cargo	Función
Guerrero Salvatierra David Ricardo	Desarrollador principal	Levantamiento de requerimientos, diseño de arquitectura MQTT, desarrollo de backend y panel de consulta, pruebas de integración y documentación técnica.
Ing. Rómulo Arévalo	Tutor académico	Asesorar en la formulación metodológica y técnica del proyecto, revisar entregables y validar la propuesta.
Ing. Saed Reascos	Experto de dominio Usuario experto / tester	Aportar los lineamientos de los proyectos integradores, validar la pertinencia académica y operativa del sistema en el contexto del vivero. Proporcionar información sobre los procesos operativos, colaborar en pruebas de campo y validar la utilidad del sistema en las actividades diarias.

4.3.2 Tecnológicos

Tabla 5 Recursos Tecnológicos

Recurso	Característica	Descripción
Laptop de desarrollo	Procesador AMD Ryzen 7 6800H, GPU RTX 3050, 24 GB de RAM, sistema operativo Windows	Equipo principal para el diseño, desarrollo, pruebas y documentación del sistema.
Servidor del backend	Equipo con sistema operativo Ubuntu Server, 4 servicios en contenedores Docker	Aloja la base de datos, el bróker MQTT, el worker de ingesta y la API REST.
Microcontrolador (estación)	ESP32 NodeMCU-32S, 240 MHz, WiFi integrado, 4 MB de memoria flash	Lee el sensor, cifra la medición con AES-256-GCM y la transmite por LoRa.
Microcontrolador (gateway)	ESP32 NodeMCU-32S, 240 MHz, WiFi integrado	Recibe los datos cifrados por LoRa y los reenvía al servidor por HTTPS.
Sensor de temperatura y humedad	DHT22 (AM2302): -40 a 80 °C ($\pm 0,5$ °C); 0 a 100 % HR (± 2 %)	Mide las condiciones ambientales del vivero.
Módulos de radiofrecuencia	Reyax RYLR (LoRa), comandos AT, antena externa, banda ISM	Enlace de largo alcance entre la estación y el gateway, sin depender de red celular.
Smartphone Android	Android 12, conectividad WiFi y datos móviles	Ejecuta la aplicación móvil de monitoreo y las pruebas de sincronización en campo.
Motor de base de datos	PostgreSQL 16	Almacena de forma centralizada las estaciones, las lecturas y el registro de auditoría.
Bróker de mensajería	Eclipse Mosquitto 2, con autenticación, ACL y TLS 1.3	Recibe la telemetría de los dispositivos que operan en la red del servidor.
Dominio y túnel seguro	Dominio propio con certificado HTTPS; túnel Cloudflare como servicio del sistema	Expone la API a Internet con una dirección fija, sin abrir puertos del servidor.
IDE de desarrollo	Cursor 3.9, Visual Studio Code con PlatformIO	Desarrollo del backend, de la aplicación móvil y del firmware de los microcontroladores.
Control de versiones	Git / GitHub (repositorio privado)	Versionado del código y respaldo del proyecto.
Conexión a Internet	Enlace de banda ancha con WiFi de 2,4 GHz	Comunica el gateway con el servidor y permite el acceso remoto al sistema.

4.3.3 Económicos

Tabla 6 Recursos Económicos

Cantidad	Recurso	Precio unitario	Subtotal
1	Laptop de desarrollo (equipo propio, valor referencial)	800,00	800,00

Cantidad	Recurso	Precio unitario	Subtotal
1	Smartphone Android de pruebas (equipo propio, valor referencial)	200,00	200,00
1	Servidor local (equipo reutilizado, valor referencial)	150,00	150,00
2	Microcontrolador ESP32 NodeMCU-32S	13,00	26,00
1	Sensor de temperatura y humedad DHT22	7,00	7,00
2	Módulo LoRa Reyax RYLR998	23,65	47,30
1	Cables, protoboard y fuente de alimentación	15,00	15,00
1	Dominio con certificado HTTPS (anual)	10,46	10,46
4 meses	Servicio de Internet	25,00 / mes	100,00
150 horas	Horas de programación (valor referencial)	10,00 / hora	1 500,00
	TOTAL		2 855,76 USD

4.4 Desarrollo del sistema bajo la metodología Scrum

Scrum es un marco de trabajo ágil utilizado para gestionar proyectos complejos, especialmente en desarrollo de software, que en si se basa en ciclos iterativos e incrementales denominados sprints. En lugar de intentar definir todos los requisitos desde inicio y desarrollar el sistema en un solo bloque, Scrum propone construir el producto por partes pequeñas y funcionales, priorizadas según el valor que aportan al usuario.

Dentro de este marco intervienen tres roles: el Product Owner, encargado de ordenar el backlog según la prioridad de cada requerimiento; el Scrum Master, que acompaña el proceso y resuelve los obstáculos que aparecen; y el equipo de desarrollo, que construye las funcionalidades acordadas en cada sprint. El trabajo se organiza mediante eventos la planificación del sprint, las reuniones diarias de seguimiento, la revisión y la retrospectiva y se apoya en tres artefactos: el product backlog, el sprint backlog y el incremento entregado al cierre de cada ciclo.

En este proyecto integrador se optó por Scrum porque hacía posible probar de forma temprana la arquitectura basada en MQTT y las funciones centrales del sistema en un escenario real como el vivero de cacao, lo que reducía el riesgo frente a la conectividad intermitente y a los cambios

que pudieran surgir en los requerimientos. Al trabajar por ciclos cortos, además, se pudo recoger la opinión del personal técnico y de los estudiantes e ir ajustando el diseño antes de llegar a las etapas más costosas.

4.4.1 Descripción del Producto

4.4.1.1 Propósito del Producto

Desarrollar un sistema informático prototipo basado en el protocolo MQTT que sincronice de forma eficiente y centralizada los datos operativos del vivero de cacao de la ULEAM Ext. El Carmen, reduciendo errores y tiempos de consolidación de los registros manuales y mejorando la confiabilidad del monitoreo automatizado para la toma de decisiones académicas y productivas.

4.4.1.2 Funcionalidades Clave

Autenticación y gestión de usuarios: Permitir el acceso al sistema mediante usuario y contraseña, diferenciando perfiles como administrador, técnico del vivero, docente y estudiante para controlar los permisos de cada rol

- **Registro de mediciones y actividades en el vivero:** Facilitar que los técnicos y estudiantes ingresen desde la app móvil o web datos como ejemplo la humedad la temperatura, riego aplicado y observaciones de las parcelas de cacao.
- **Envío de datos al bróker MQTT:** Publicar automáticamente las mediciones registradas en los tópicos MQTT estructurados por parcela, tipo de variable y el turno garantizando un flujo de datos ligero y eficiente.
- **Ingestión y validación de mensajes MQTT:** Suscribirse a los tópicos definidos, validar el formato de los mensajes recibidos, descartar registros corruptos y preparar los datos de su almacenamiento en la base de datos central.
- **Sincronización y almacenamiento centralizado de registros:** Guardar en un repositorio único los datos operativos del vivero (variables ambientales, riesgos y

actividades) esto asegura que estén y se estén actualizando, sean accesibles y consistentes para consultas posteriores.

- **Visualización de históricos en un panel de control:** Mostrar en un Dashboard móvil gráficos y tablas de las variables del vivero filtradas por fecha, la parcela y también el tipo del registro así facilitando el análisis de tendencia y el monitoreo continuo.
- **Generación de reportes operativos y académicos:** Generar reportes descargables (por ejemplo, en PDF o CSV) sobre los riesgos realizados, condiciones ambientales y las actividades de los estudiantes en un periodo determinado.
- **Gestiona de catálogos del vivero:** Administrar catálogos de parcelas, cultivos, tipos de variables, turnos de prácticas y equipos utilizados para mantener ordenada la información de referencia del sistema.
- **Configuración y gestión:** Definir umbrales para variables críticas (podría ser cómo la humedad del suelo) y emitir alerta cuando se detecten valores fuera de rango, apoyando la dirección temprana de estrés hídrico u otros problemas.
- **Monitoreo del estado de nodos y trazabilidad del sistema:** Registrar eventos del sistema, estado de conexión de nodos de captura y errores de comunicación para facilitar el diagnóstico, la auditoría y el mantenimiento del sistema en el tiempo

4.4.1.3 Usuarios Objetivo:

Tabla 7 Usuarios Objetivos

Tipo de usuario	Principales funcionalidades
Administrador del sistema	Gestión de usuarios y roles, la configuración de parámetros generales, administración de catálogos, supervisión de logs y monitoreo del estado del sistema.
Técnico del vivero	Registro de actividades y mediciones, consulta de indicadores en tiempo casi real, generación de reportes operativos, atención de alertas
Docente coordinador	Consulta de históricos, evaluación de desempeño de prácticas esto ayuda a la revisión de indicaciones de producción y de insumos
Estudiantes de agropecuaria	Registros de observaciones y mediciones durante prácticas, revisión de información básica del estado de las parcelas asignadas

4.4.1.4 Criterios de Éxito del Producto

Reducir en al menos un 50% el tiempo requerido para consolidar los registros de campo un informe administrativo respecto al proceso manual previo.

- Disminuir la tasa de omisiones y duplicidades en los registros operativos en por lo menos al menos un 40% tras la implementación del sistema.
- Lograr que la latencia promedio entre el envío de un dato desde el nodo de captura y su disponibilidad en el panel de consulta sea menor a 5 segundos en condiciones normales de conectividad.
- Asegurar la disponibilidad del servicio de sincronización de datos (bróker y backend) igual o superior al 99% durante el periodo de evaluación.
- Conseguir que al menos el 80% de los usuarios (técnicos y estudiantes encuestados) tengan que estar “de acuerdo” o “totalmente de acuerdo” con que el sistema facilita la gestión de información del vivero

4.4.2 Requerimientos del sistema

4.4.2.1 Historias de Usuario

4.4.2.1.1 Historias de Usuario: Autenticación y navegación básica

Tabla 8 Historia de usuario HU01: Autenticación y navegación básica

HISTORIA DE USUARIO			
Id de historia	HU01	Título:	Autenticación y navegación básica
Prioridad:	Alta	Usuario:	Administrador
Riesgo de desarrollo:	Medio	Tamaño de la tarea:	Grande
Como:	Administrador		
Quiero:	Acceder a la plataforma mediante usuario y contraseña válidos		
Para:	Gestionar los módulos disponibles en el menú		
Criterios de aceptación:	Gestionar los módulos disponibles en el menú según mi perfil		
Dependencias:	Base de datos con usuarios registrados		

4.4.2.1.2 Historias de Usuario: Registro de mediciones en campo

Tabla 9 Historia de usuario HU02: Registro de mediciones en campo

HISTORIA DE USUARIO			
Id de historia	HU02	Título:	Registro de mediciones en campo
Prioridad:	Alta	Usuario:	Estudiante / Técnico del vivero
Riesgo de desarrollo:	Medio	Tamaño de la tarea:	Mediana
Como:	Estudiante o técnico del vivero		

HISTORIA DE USUARIO	
Quiero:	Registrar desde mi dispositivo las mediciones de variables ambientales y actividades, ingresando parcela, tipo de variable, valor medido, fecha y hora mediante un formulario sencillo
Para:	Que la información quede disponible para su sincronización y posterior análisis en el sistema central
Criterios de aceptación:	Los campos obligatorios no pueden quedar vacíos; al momento de guardar el sistema confirma el registro y envía el dato hacia el bróker MQTT para su procesamiento.
Dependencias:	Base de datos con usuarios registrados

4.4.2.1.3 Historias de Usuario: Sincronización y almacenamiento centralizado

Tabla 10 . Historia de usuario HU03: Sincronización y almacenamiento centralizado

HISTORIA DE USUARIO			
Id de historia	HU03	Título:	Sincronización y almacenamiento centralizado
Prioridad:	Alta	Usuario:	Sistema / Administrador
Riesgo de desarrollo:	Medio	Tamaño de la tarea:	Grande
Como:	Servicio de sincronización del servidor		
Quiero:	Recibir los mensajes publicados en los tópicos MQTT, validar el contenido de cada uno y almacenarlo en la base de datos central con su marca de tiempo y metadatos		
Para:	Mantener un repositorio único actualizado y consistente de los registros operativos del vivero		
Criterios de aceptación:	Los mensajes validos se guardan correctamente en la base de datos los mensajes corruptos o incompletos se descartan y registran en un log de errores; no se pierden registros cuando hay reconexión del servicio		
Dependencias:	Bróker MQTT, servicio de ingestión suscrito a los tópicos, base de datos configurada y accesible		

4.4.2.1.4 Historias de Usuario: Visualización de históricos y monitoreo

Tabla 11 Historia de usuario HU04: Visualización de históricos y monitoreo

HISTORIA DE USUARIO			
Id de historia	HU04	Título:	Visualización de históricos y monitoreo
Prioridad:	Media	Usuario:	Docente / técnico del vivero
Riesgo de desarrollo:	Medio	Tamaño de la tarea:	Mediana
Como:	Docente coordinador o técnico del vivero		
Quiero:	Consultar gráficos y tablas históricas de las variables registradas por parcela y rango de fechas, filtrando por tipo de variable, fecha y parcela		

HISTORIA DE USUARIO	
Para:	Analizar la evolución de las condiciones del cultivo y tomar decisiones oportunas sobre el riego, el manejo fitosanitario y las actividades de los estudiantes
Criterios de aceptación:	El sistema permite seleccionar filtros y muestra los datos correspondientes sin errores los gráficos mismos se actualizan con nuevos registros esto ayuda a exportar la información consultada a CSV o PDF
Dependencias:	Base de datos con registros históricos, módulos y componente de visualización grafica

4.4.2.1.5 Historias de Usuario: Gestión de alertas por condiciones

Tabla 12 Historia de usuario HU05: Gestión de alertas por condiciones

HISTORIA DE USUARIO			
Id de historia	HU05	Título:	Gestión de alertas por condiciones
Prioridad:	Alta	Usuario:	Técnico del vivero
Riesgo de desarrollo:	Medio	Tamaño de la tarea:	Mediana
Como:	Técnico del vivero		
Quiero:	Configurar umbrales de alerta para variables críticas (como la humedad del suelo o la temperatura) y que el sistema genere una alerta cuando los valores registrados superen o estén por debajo de los rangos establecidos		
Para:	Detectar de forma temprana situaciones de estrés hídrico u otras condiciones que puedan afectar la salud de las plantas.		
Criterios de aceptación:	El técnico puede definir y actualizar los rangos por parcela y variable cuando hay un dato excedente este umbral se registra y una alerta visible en el panel de control mismas a letras incluyen fecha hora parcela variable y el valor que la origino		
Dependencias:	Módulo de configuración de umbrales, servicio de procesamiento de datos y panel de visualización de alertas		

4.4.2.2 Requerimientos funcionales

- **RF01:** El sistema debe suscribirse al bróker MQTT y rescribir los lotes de telemetría de todas las estaciones.
- **RF02:** El sistema debe descifrar el contenido de cada lote mediante AES-256-GCM.
- **RF03:** El sistema debe verificar la autenticidad e integridad del mensaje mediante el tag de autenticación.
- **NF04:** El sistema debe validar que cada lectura esté dentro de rangos físicos válidos.
- **RF05:** El sistema debe evitar el almacenamiento de lecturas duplicadas.
- **RF06:** El sistema debe registrar en auditoría el resultado de cada lote procesado.

- **RF07:** El sistema debe exponer una API REST para la consulta de lecturas y dispositivos.
- **RF08:** El sistema debe crear automáticamente el registro de una estación nueva al recibir su primer lote.

4.4.2.3 Requerimientos no funcionales

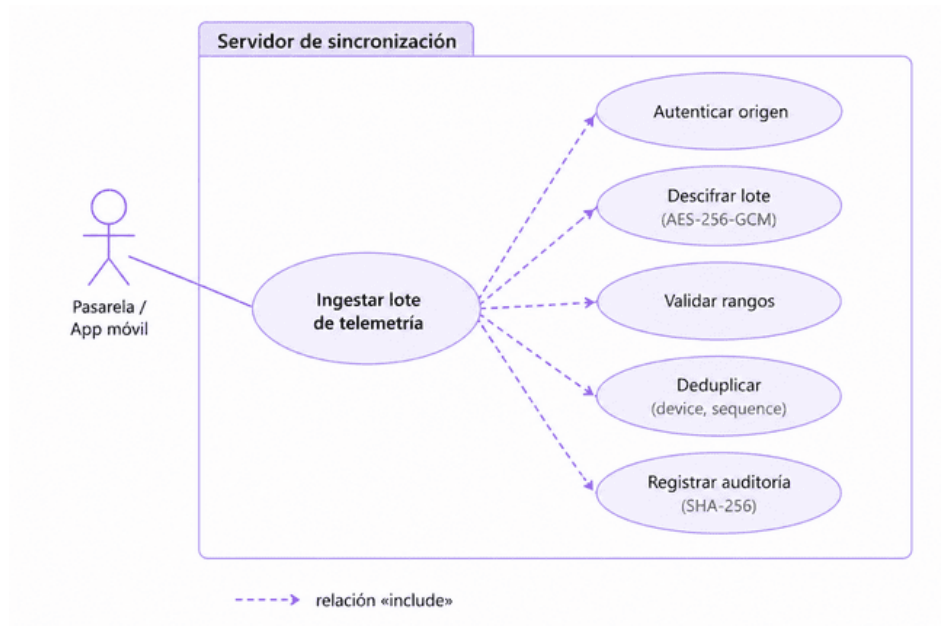
- **RNF01 (Seguridad):** La comunicación debe protegerse con cifrado de extremo a extremo (AES-256-GCM) y transporte cifrado (TLS 1.3)
- **RNF02 (Confiabilidad):** El sistema debe tolerar desconexiones y reanudar la operación automáticamente
- **RNF03 (Escalabilidad):** El sistema debe soportar múltiples estaciones sin cambios en el código
- **RNF04 (Mantenibilidad):** El sistema debe organizarse en componentes desacoplados y desplegarse mediante contenedores
- **RNF05 (Privacidad):** El sistema no debe exponer detalles internos ni datos sensibles ante errores
- **RNF06 (Reproducibilidad):** El despliegue debe ser reproducible mediante un único archivo de orquestación (Docker Compose) y migraciones de base de datos versionadas

4.4.3 Diseño del Sistema

4.4.3.1 Casos de Uso

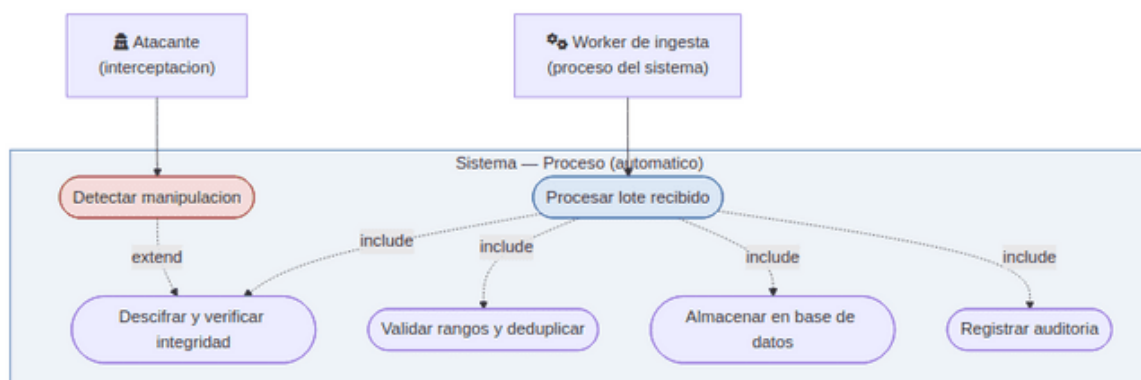
4.4.3.1.1 Caso de uso: Entrada de datos

Ilustración 10 Ingesta de datos



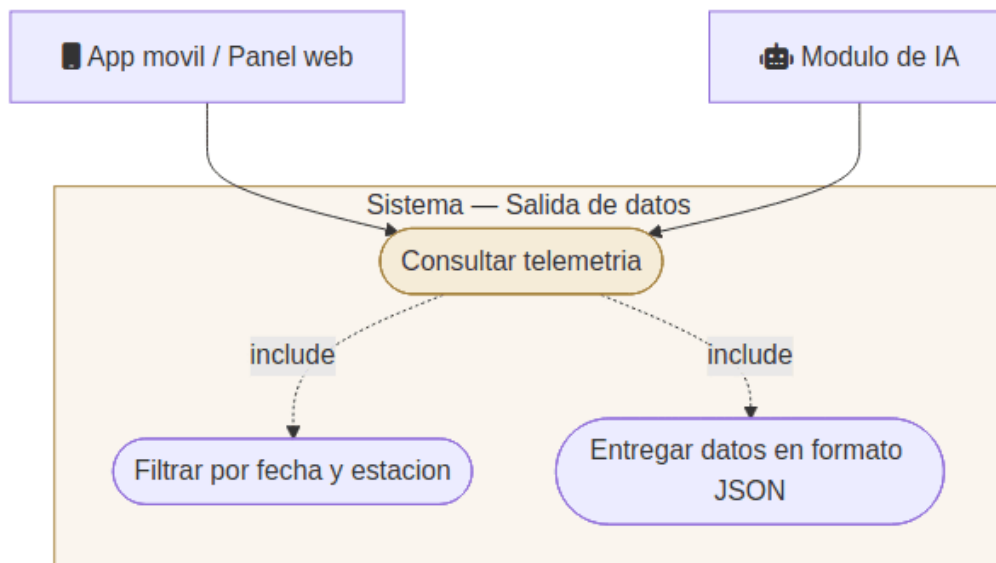
4.4.3.1.2 Caso de uso: Procesamiento y validación de datos

Ilustración 11 Validación de datos



4.4.3.1.3 Caso de uso: Consulta de telemetría

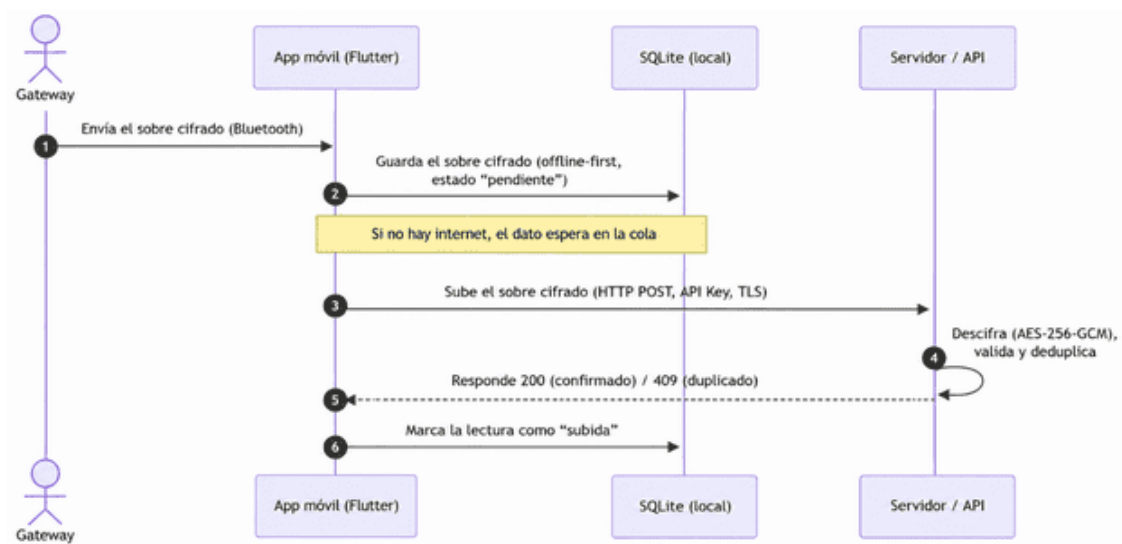
Ilustración 12 Consulta de telemetría



4.4.3.2 Diagramas de Secuencia

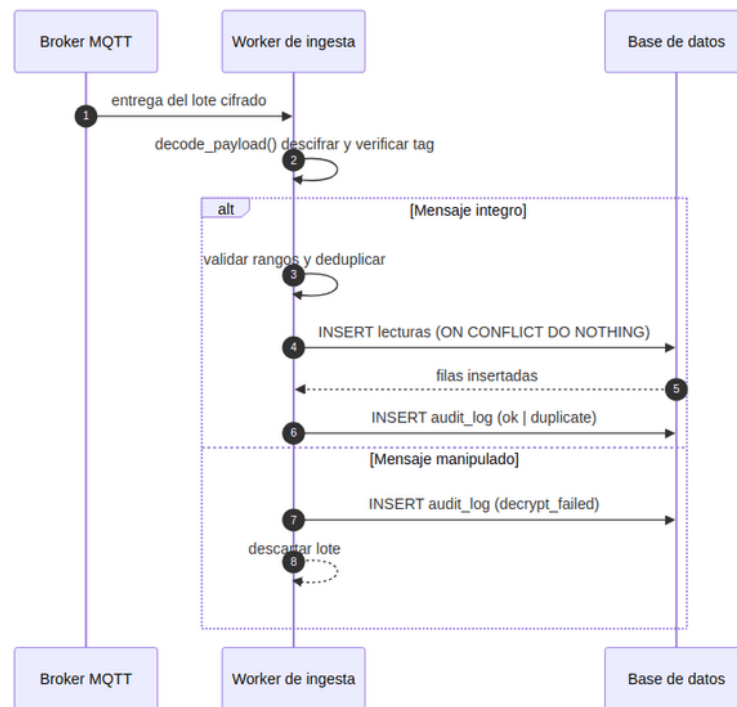
4.4.3.2.1 Diagrama de secuencia: Captura y sincronización de una lectura

Ilustración 13 Diagrama de secuencia del proceso de captura y sincronización de una lectura



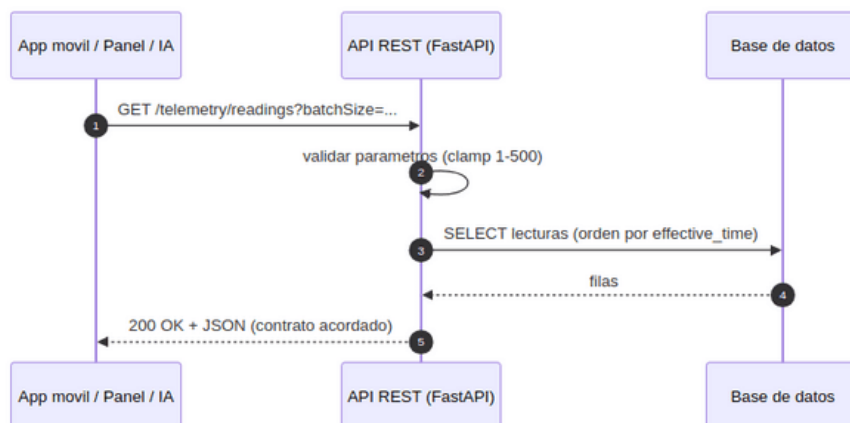
4.4.3.2 Diagrama de secuencia: Procesamiento del lote

Ilustración 14 Diagrama de secuencia del procesamiento de un lote de telemetría



4.4.3.2.3 Diagrama de secuencia: Consulta de lecturas

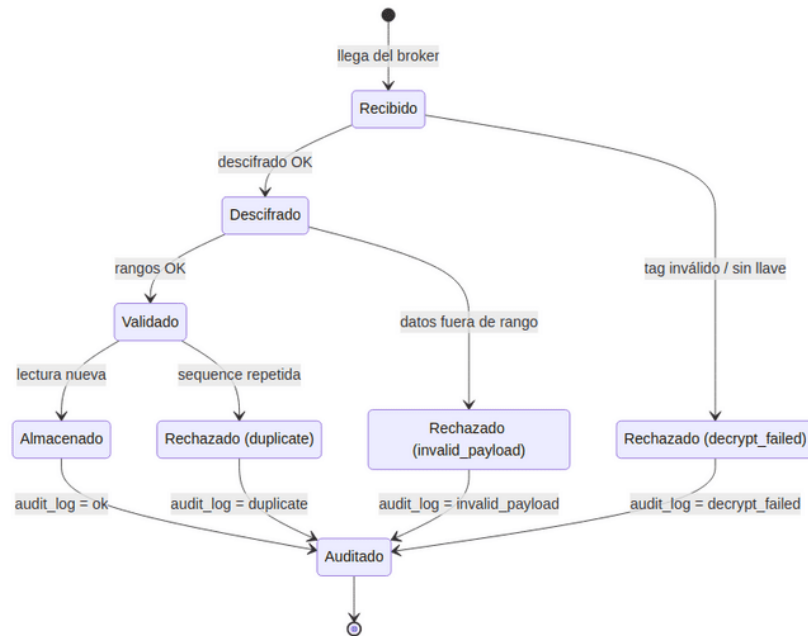
Ilustración 15 Diagrama de secuencia de la consulta de lecturas



4.4.3.3 Diagramas de Estado

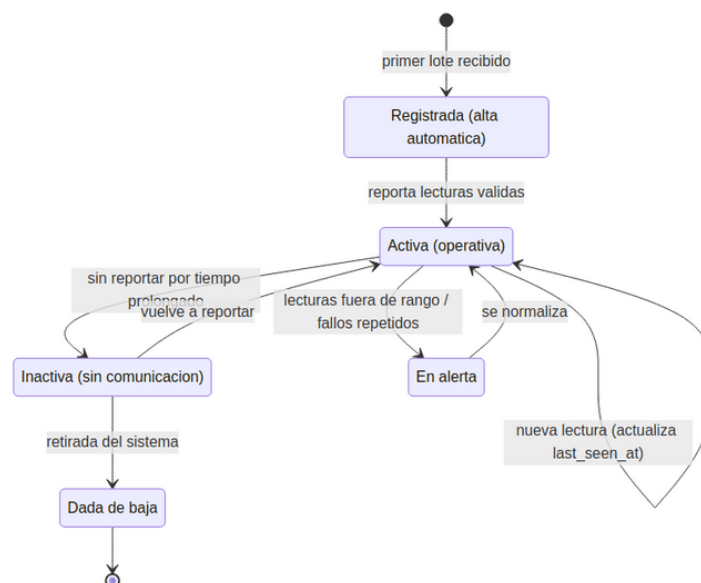
4.4.3.3.1 Diagrama de estado: Ciclo de vida de un lote de telemetría

Ilustración 16 Diagrama de estados del ciclo de vida de un lote de telemetría



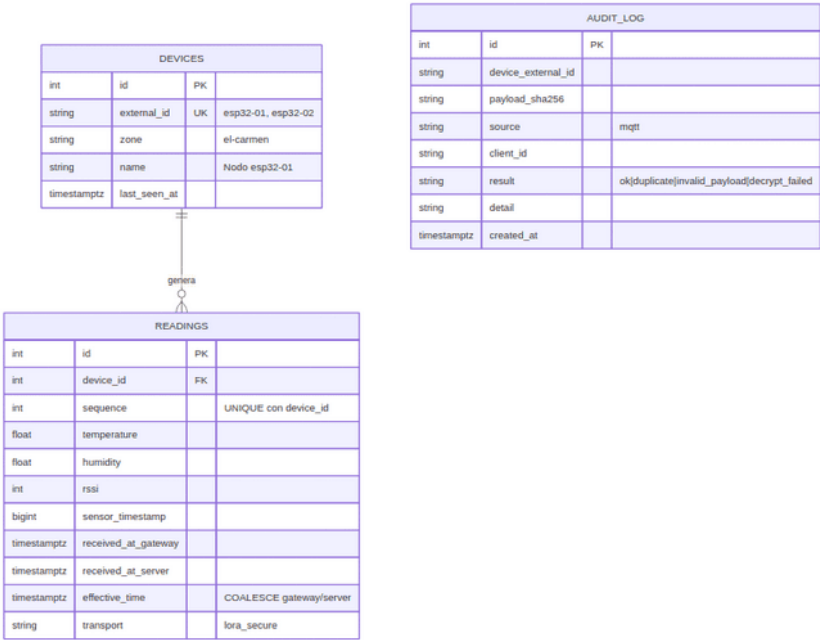
4.4.3.3.2 Diagrama de estado: Dispositivo registrado en el servidor

Ilustración 17 Diagrama de estados de un dispositivo registrado en el servidor



4.4.3.4 Diagrama de Base de Datos

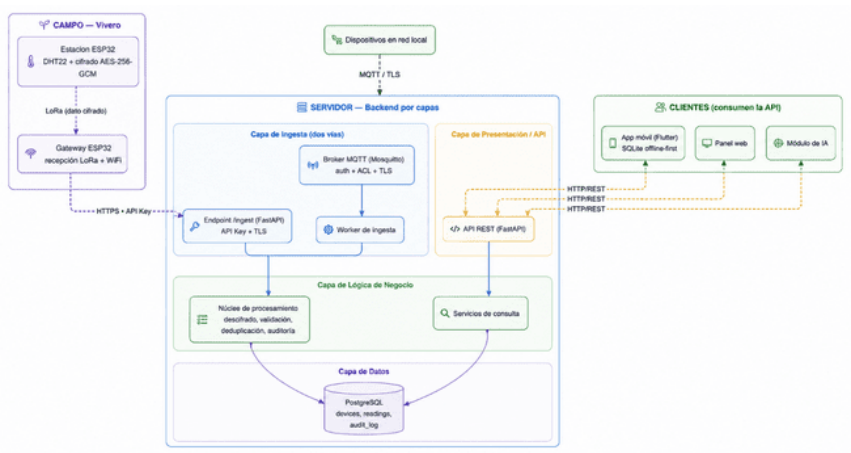
Ilustración 18 Diagrama de la base de datos



Arquitectura del sistema

4.4.3.5 Arquitectura general

Ilustración 19 Arquitectura del programa



El sistema sigue una arquitectura orientada a IoT desplegada mediante contenedores sobre un servidor propio, con dos vías de ingestión de datos que comparten la misma lógica de procesamiento. El flujo de datos es: Estacion ESP32 (sensor DHT22; cifrado AES-256-GCM en el propio microcontrolador) → **LoRa** (el dato viaja cifrado por radiofrecuencia) → **Gateway ESP32** (recepción LoRa y conexión WiFi) → **HTTPS** → **API REST (FastAPI)** → **Base de datos PostgreSQL** → consumidores: **aplicación móvil** (almacenamiento local offline-first en SQLite), **panel web de monitoreo** y **módulo de inteligencia artificial**.

De forma complementaria, el sistema mantiene una vía de ingestión por **MQTT** sobre **TLS** (bróker Mosquito y worker de ingesta), esto lo utiliza la misma red del servidor. Ambas vías – **HTTP** y **MQTT** – convergen en la misma función de procesamiento (proces_envelope) de modo que el descifrado, la validación, la deduplicación y la auditoría son idénticos también son independientes del canal de entrada.

Justificación de la doble vía de ingestión. El diseño original estaba contemplado con MQTT como único canal de entrada. Durante la implementación se determinó que el gateway estaría físicamente ubicado en el vivero, una red distinta a la del servidor lo cual exigía exponer el bróker MQTT a internet: una alternativa compleja de asegurar y con mayor superficie de ataque. Se optó por que el gateway remoto utilice **HTTP/REST** sobre **TLS** aprovechando la misma infraestructura de dominio, túnel seguro y autenticación por API Key ya existir. Esta decisión aporta robustez ante conectividad intermitente – característica del entorno rural – y simplifica el firmware del microcontrolador.

El canal MQTT se conserva para la ingestión interna, así manteniendo el protocolo como componente central de la arquitectura de sincronización

4.4.3.6 Tecnologías utilizadas

Tabla 13 Tecnologías usadas

Categoría	Herramienta	Uso dentro del proyecto
Lenguaje	Python 3.12	Lenguaje principal del backend (API y worker)
Framework de API	FastAPI	Construcción de la API REST y validación automática de datos
Servidor de aplicación	Uvicorn	Servidor que ejecuta la API

Categoría	Herramienta	Uso dentro del proyecto
Gestor de base de datos	PostgreSQL 16	Almacenamiento de estaciones, lecturas y auditoría
ORM / acceso a datos	SQLAlchemy	Mapeo de las tablas a objetos de Python
Migraciones	Alembic	Versionado y creación del esquema de la base de datos
Mensajería IoT	Eclipse Mosquitto 2 (MQTT)	Bróker que recibe la telemetría de las estaciones
Cliente MQTT	Paho-MQTT	Librería con la que el worker se conecta al bróker
Cifrado (servidor)	Cryptography (AES-256-GCM)	Cifrado y descifrado autenticado de los datos
Cifrado (dispositivo)	mbedTLS (AES-256-GCM)	Cifra la lectura dentro de la estación ESP32, antes de transmitirla
Seguridad de transporte	TLS 1.3 / OpenSSL	Canal cifrado y certificados del bróker
Autenticación de la API	API Key (cabecera X-API-Key)	Control de acceso a los endpoints de la API
Exposición del servidor	Cloudflare Túnel + dominio propio	Acceso a la API desde Internet con HTTPS, sin abrir puertos
Microcontrolador	ESP32 (Arduino / PlatformIO)	Estación de medición y gateway de recepción
Sensor	DHT22	Medición de temperatura y humedad
Comunicación de largo alcance	LoRa (módulos Reyax RYLR)	Transmisión del dato cifrado desde la estación al gateway
Aplicación móvil	Flutter + SQLite	Visualización y almacenamiento local offline-first
Contenedores	Docker y Docker Compose	Empaquetado y orquestación de los cuatro servicios
Pruebas	pytest	Batería de pruebas automatizadas (52 pruebas)
Control de versiones	Git / GitHub	Versionado del código y respaldo del proyecto
Sistema operativo del servidor	Ubuntu Server 22.04	Sistema sobre el que corre el backend

4.4.4 Roles Scrum

Tabla 14 Roles de Scrum

Rol Scrum	Función	Integrante(s)
Product Owner	Define las prioridades y los requisitos del producto; representa las necesidades del usuario final (el vivero)	Ing. Pedro Nivela

Rol Scrum	Función	Integrante(s)
Scrum Master	Facilita el proceso, elimina obstáculos y vela por el cumplimiento de la metodología	David Guerrero
Scrum Team (equipo de desarrollo)	Construye los incrementos del producto en cada Sprint	Todo el equipo mientras que en el backend sincronización segura y el cifrado: David Ricardo Guerrero Salvatierra

4.4.5 Definición de Hecho (DoD)

4.4.5.1 Criterios Generales

Una funcionalidad se considere “Realizada” cuando se cumple con los siguientes criterios generales:

- El código fuente debe compilar y ejecutarse correctamente sin ningún error.
- La funcionalidad desarrollada debe cumplir los requisitos establecidos en la historia de usuario correspondiente
- La funcionalidad debe estar integrada correctamente con los demás módulos del sistema (bróker, base de datos, API).
- El comportamiento debe estar verificado mediante pruebas, sin errores en los casos contemplados
- El código debe estar versionado en el repositorio (Git) sin romper funcionalidades previas
- No se deben exponer datos sensibles ni detalles internos del sistema ante errores

4.4.5.2 Criterios Específicos del Proyecto

Además de los criterios generales cada función debe cumplir con los criterios específicos correspondientes a su naturaleza:

- La recepción de telemetría debe aceptar únicamente lotes provenientes de estaciones autenticadas en el bróker

- El descifrado debe procesar correctamente los lotes cifrados con la llave valida del dispositivo, y rechazar todo mensaje manipulado, registrándolo como decrypt_failed
- La ingesta de datos debe validar que cada lectura se encuentre dentro de los rangos físicos permitidos antes de almacenarla
- El almacenamiento debe descartar automáticamente las lecturas duplicadas, conservando una única copia por estación y secuencia.
- Cada lote procesado debe generar exactamente un registro de auditoría con resultado y su huella de integridad (Sha-256)
- La consulta de telemetría debe devolver la información únicamente en el formato acordado con la aplicación móvil (contrato de datos)
- La comunicación entre los clientes externos y el bróker debe realizarse a través de un canal cifrado (TLS)

4.4.6 Product Backlog

Tabla 15 Estado del Product Backlog

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Infraestructura base del backend (servidor, contenedores Docker)	HU03	Alta	Por hacer
2	Modelo de datos y migraciones (devices, readings, audit_log)	HU03	Alta	Por hacer
3	API REST de consulta de lecturas y dispositivos	HU04	Alta	Por hacer
4	Recepción de telemetría vía MQTT (worker de ingesta)	HU03	Alta	Por hacer
5	Validación de datos y deduplicación de lecturas	HU03	Alta	Por hacer
6	Auditoría de lotes con huella de integridad (SHA-256)	HU03	Media	Por hacer
7	Cifrado de extremo a extremo (AES-256-GCM)	HU03	Alta	Por hacer
8	Detección de manipulación de mensajes	HU03	Alta	Por hacer
9	Autenticación MQTT y control de acceso (ACL)	HU03	Alta	Por hacer
10	Transporte cifrado (TLS 1.3)	HU03	Alta	Por hacer

Nro.	Tarea	Historia de usuario	Prioridad	Estado
11	Soporte de umbrales para alerta (datos para HU05)	HU05	Media	Por hacer
12	Pruebas automatizadas y documentación técnica	HU03 y HU04	Media	Por hacer
13	Cifrado AES-256-GCM en el firmware de la estación ESP32	HU03	Alta	Por hacer
14	Endpoint de ingestión HTTP para el gateway remoto	HU03	Alta	Por hacer
15	Autenticación de la API mediante API Key	HU03	Alta	Por hacer
16	Exposición del servidor con dominio propio y túnel seguro	HU04	Alta	Por hacer
17	Firmware del gateway: recepción LoRa y reenvío HTTPS	HU03	Alta	Por hacer
18	Persistencia del número de secuencia en memoria no volátil	HU03	Media	Por hacer
19	Integración y validación del sistema con datos reales	HU03, HU04	Alta	Por hacer

4.4.7 Sprint 1: Infraestructura, base de datos y API

4.4.7.1 Planificación del Sprint

Tabla 16 Infraestructura del backend, base de datos y API de consulta

SPRINT 1: Infraestructura del backend, base de datos y API de consulta			
Duración:	27/05/2026 – 12/06/2026 (17 días)	Prioridad:	Alta
Objetivo:	Preparar el servidor y levantar la infraestructura base del backend mediante contenedores (la base de datos, el bróker y el API), definir el modelo de datos con migraciones versionadas y exponer los endpoints de consulta con el formato requerido por la aplicación móvil		
Historias de usuario:	HU03: Sincronización y almacenamiento centralizado: HU04: Visualización de históricos y monitoreo		
Tareas de desarrollo:	- Configuración del servidor (sistema operativo, Docker, red, firewall, almacenamiento). - Definición de la orquestación de servicios con Docker Compose. - Modelado de la base de datos (devices, readings, audit_log). - Creación de migraciones con Alembic		

SPRINT 1: Infraestructura del backend, base de datos y API de consulta	
	• Implementación de los endpoints REST de consulta • Verificación del contrato de datos con la aplicación móvil
Plan de entrega:	• Configuración del servidor. Preparación de Ubuntu Server, Instalación de Docker, configuración de red y almacenamiento dedicado Tiempo de desarrollo: (12)h fechas: (27/05/2026 – 31/05/2026) • Orquestación de servicios. Definición de los contenedores de base de datos, bróker y API en un único archivo. Tiempo de desarrollo: (8)h – Fechas: (01/06/2026 – 04/06/2026) • Modelo de datos y migraciones. Definición de las tablas y aplicación de migraciones. Tiempo de desarrollo: (10) h — Fechas: (05/06/2026 – 09/06/2026) • Endpoints de consulta. Implementación y verificación del formato de respuesta. Tiempo de desarrollo: (10)h — Fechas (10/06/2026 – 12/06/2026)

4.4.7.2 Sprint Backlog Sprint 1

Tabla 17 Sprint Backlog del Sprint 1

Historia de usuario	Tarea	Duración estimada
HU03	Infraestructura base del Backend (servidor, contenedores)	20 horas
HU03	Modelo de datos y migraciones (devices, readings y audit_log)	10 horas
HU04	API REST de consulta de lecturas y dispositivos	10 horas
	Total	40 horas

4.4.7.3 Desarrollo del Sprint 1

4.4.7.3.1 Funcionalidad de infraestructura del backend, modelo de datos y API de consulta

Descripción: En esta funcionalidad se construyó la base del sistema aquí se desplegaron cuatro servicios en contenedores (base de datos PostgreSQL, bróker MQTT, worker y API), se definió el modelo de datos mediante migraciones versionadas y se implementó la API REST que entrega las lecturas en el formato requerido por la aplicación móvil.

Los cuatro servicios del backend —base de datos, bróker MQTT, worker y API -- se ejecutan como contenedores en estado activo (Up):

Ilustración 20 Servicios del backend en ejecución contenedores Docker

```
david@dsrver:~$ cd ~/Back-App-Monitoreo
david@dsrver:~/Back-App-Monitoreo$ docker compose ps

```

NAME	IMAGE	COMMAND	SERVICE	CREATED	STATUS	PORTS
monitoreo-api	back-app-monitoreo-api	"uvicorn app.main:ap..."	api	5 days ago	Up 5 days	0.0.0.0:8000->8000/tcp, [::]:8000->8000/tcp
monitoreo-db	postgres:16	"docker-entrypoint.s..."	db	13 days ago	Up 13 days (healthy)	127.0.0.1:5432->5432/tcp
monitoreo-mqtt	eclipse-mosquitto:2	"docker-entrypoint.s..."	mosquitto	5 days ago	Up 5 days	127.0.0.1:1883->1883/tcp, 0.0.0.0:8883->8883/tcp, [::]:8883->8883/tcp
monitoreo-worker	back-app-monitoreo-worker	"python -m app.main"	worker	5 days ago	Up 5 days	

```
david@dsrver:~/Back-App-Monitoreo$
```

La API devuelve las lecturas con las claves exactas que consume la aplicación móvil (device_id, temperature, humidity, etc.), confirmando la compatibilidad del contrato de datos:

Ilustración 21 Respuesta de la API con el contrato de datos

```
{
  "version": 1,
  "deviceId": "esp32-01",
  "sensorName": "Nodo esp32-01",
  "sequence": 102,
  "temperature": 25.7,
  "humidity": 55.2,
  "rssi": -70,
  "sensorTimestamp": null,
  "receivedAtGateway": "2026-07-10T16:37:31.689736Z",
  "syncStatus": "synced",
  "transport": "lora_secure"
},
{
  "version": 1,
  "deviceId": "esp32-01",
  "sensorName": "Nodo esp32-01",
  "sequence": 101,
  "temperature": 25.9,
  "humidity": 55.0,
  "rssi": -70,
  "sensorTimestamp": null,
  "receivedAtGateway": "2026-07-10T16:36:54.658497Z",
  "syncStatus": "synced",
  "transport": "lora_secure"
},
{
  "version": 1,
  "deviceId": "esp32-01",
  "sensorName": "Nodo esp32-01",
  "sequence": 100,
  "temperature": 26.2,
  "humidity": 54.7,
  "rssi": -70,
  "sensorTimestamp": null,
  "receivedAtGateway": "2026-07-10T16:36:18.511840Z",
  "syncStatus": "synced",
  "transport": "lora_secure"
}
],
"count": 3
}
david@dsrver:~/Back-App-Monitoreo$
```

```

class Reading(Base):
    """Lectura de telemetría asociada a un dispositivo."""

    __tablename__ = "readings"
    __table_args__ = (
        UniqueConstraint("device_id", "sequence",
            name="uq_readings_device_sequence"),
        Index("ix_readings_device_effective_time", "device_id", "effective_time"),
    )

    id: Mapped[int] = mapped_column(primary_key=True)
    device_id: Mapped[int] = mapped_column(ForeignKey("devices.id"),
        nullable=False)
    sequence: Mapped[int] = mapped_column(nullable=False)
    temperature: Mapped[float] = mapped_column(nullable=False)
    humidity: Mapped[float] = mapped_column(nullable=False)
    rssi: Mapped[int] = mapped_column(nullable=False)
    sensor_timestamp: Mapped[int | None] = mapped_column(BigInteger)
    received_at_gateway: Mapped[datetime | None] =
        mapped_column(DateTime(timezone=True))
    received_at_server: Mapped[datetime] = mapped_column(
        DateTime(timezone=True), server_default=func.now(), nullable=False)
    effective_time: Mapped[datetime] = mapped_column(
        DateTime(timezone=True),
        Computed("COALESCE(received_at_gateway, received_at_server)",
            persisted=True))
    transport: Mapped[str] = mapped_column(String(64), nullable=False,
        default="lora_secure")

    device: Mapped[Device] = relationship(back_populates="readings")

```

La clase Reading corresponde a la tabla donde se guardan las lecturas de telemetría. Cada lectura queda asociada a la estación que la generó a través del campo device_id, que actúa como clave foránea hacia la tabla devices. Los campos temperature, humidity y rssi contienen los valores medidos. Para ubicar cada dato en el tiempo se guardan tres marcas: sensor_timestamp, que corresponde al momento en que el sensor tomó la medición; received_at_gateway, cuando el gateway la recibió; y received_at_server, cuando llegó al servidor. Por último, effective_time es una columna calculada que, mediante la función COALESCE, toma la hora del gateway y, si esta no existe, la del servidor; de esta manera las lecturas pueden ordenarse por el momento real en que se midieron.

4.4.7.4 Sprint Review Sprint 1

4.4.7.4.1 Descripción de la funcionalidad entregada

Al cierre del primer Sprint se entregó la infraestructura base del backend en funcionamiento: los cuatro servicios desplegados como contenedores independientes además del modelo de datos aplicado mediante migraciones versionadas y la API REST entregado las lecturas en el formato acordado con la aplicación móvil.

4.4.7.4.2 Evidencia de la implementación:

Ilustración 22 Servicios del backend en ejecución (contenedores Docker)

```
David@server:~/Back-App-Monitoreo$ cd ~/Back-App-Monitoreo && docker compose ps
NAME                IMAGE                COMMAND                SERVICE    CREATED        STATUS                PORTS
monitoreo-api        back-app-monitoreo-api    "devicors app.main:ap..." api        5 days ago     Up 11 minutes        0.0.0.0:8000->8000/tcp, [::]:8000->8000/tcp
monitoreo-db         postgres:16           "docker-entrypoint.s..." db        4 weeks ago     Up 11 minutes (healthy) 127.0.0.1:5432->5432/tcp
monitoreo-mqtt       eclipse-mosquitto:2     "/docker-entrypoint..." mosquitto  3 weeks ago     Up 11 minutes        127.0.0.1:1883->1883/tcp, 0.0.0.0:1883->1883/tcp
monitoreo-worker     back-app-monitoreo-worker    "python -m app.main"    worker    5 days ago     Up 11 minutes
```

Ilustración 23 Respuesta de la API con el contrato de datos

```
David@server:~/Back-App-Monitoreo$ curl -s -H "X-API-Key: d964ecb0094b7a95907d1145adec9c3c64e91b68b034efed7a96c8db43df" "https://api.darigo.com/api/v1/telemetry/re
adings?batchSize=3" | python3 -m json.tool
{
  "items": [
    {
      "version": 1,
      "deviceId": "esp32-01",
      "sensorName": "Modo esp32-01",
      "sequence": 194,
      "temperature": 24.7,
      "humidity": 85.3,
      "tssi": -70,
      "sensorTimestamp": null,
      "receivedAtGateway": "2024-07-13T08:48:51.450712Z",
      "syncStatus": "synced",
      "transport": "lorawan_secure"
    },
    {
      "version": 1,
      "deviceId": "esp32-01",
      "sensorName": "Modo esp32-01",
      "sequence": 191,
      "temperature": 24.7,
      "humidity": 85.3,
      "tssi": -70,
      "sensorTimestamp": null,
      "receivedAtGateway": "2024-07-13T08:48:51.450712Z",
      "syncStatus": "synced",
      "transport": "lorawan_secure"
    }
  ]
}
```

- **Retroalimentación establecida**

Durante la verificación de la funcionalidad con el equipo de desarrollo del frontend se detectó que el formato de respuesta de la API no coincidía plenamente con el que esperaba la aplicación móvil. Se identificaron diferencias en los nombres de las claves JSON, lo que habría impedido que la aplicación interpretara correctamente las lecturas recibidas.

Además, se observó que el ordenamiento de las lecturas debía basarse en el momento real de la medición y no en el instante de su llegada al servidor ya que un lote sincronizado con retraso habría quedado situado fuera de su lugar cronológico.

- **Mejoras realizadas**

Se ajustó el contrato de datos de la API, unificando los nombres de las claves del JSON con los que consume la aplicación (deviceId, sequence, temperature, humidity, receivedGateway). Este contrato quedo documentado y protegido mediante una prueba

automatizada de nombre `test_telemetry_readings_contract` de modo que cualquier modificación futura que pueda llegar a romperse sea detectada antes del despliegue.

Además, se incorporó al modelo de datos el campo calculado `effective_time` que toma el momento de recepción en el gateway y en su ausencia el del servidor. Ya con ello las lecturas se ordenan por el momento real de la medición así garantizando la coherencia cronológica del histórico aun cuando la sincronización se produzca con retraso.

4.4.7.5 Pruebas del Sprint 1, Caja Blanca

Tabla 18 Pruebas de caja blanca del Sprint 1 – Infraestructura, base de datos y API

Método	Acción esperada	Acción obtenida	Observación
Levantamiento de los servicios en contenedores	Los servicios (base de datos, bróker y API) inician y quedan operativos	Los servicios iniciaron correctamente; la base de datos quedó en estado <i>healthy</i>	Infraestructura verificada.
Consulta de lecturas por la API (GET /readings)	Responder 200 con las lecturas en el formato del contrato	Respondió 200 con el JSON de lecturas esperado	Endpoint de consulta verificado.
Consulta con el parámetro <code>batchSize</code> fuera de rango	Ajustar el valor al rango permitido (1–500)	El valor se ajustó automáticamente al límite	Validación de parámetros verificada.
Inserción de dos lecturas con la misma estación y secuencia	Impedir duplicado por la restricción unicidad	La segunda inserción no se realizó (restricción <code>uq_readings_device_sequence</code>)	Restricción de unicidad verificada.

4.4.7.6 Incremento y entregable del Sprint 1

Al finalizar el Sprint 1 se complementaron satisfactoriamente los siguientes procesos:

- Despliegue de la infraestructura base del backend en contenedores (base de datos, broker de mensajería y el servicio de la API)

- Definición del modelo de datos y creación de las migraciones versionadas para las tablas de dispositivos lecturas y auditorias.
- Implementación de la restricción de unicidad por estación y numero de secuencia, que es la base de la deduplicacion.
- El desarrollo de la API REST de consulta de lecturas y dispositivos conforme al contrato acordado con la aplicación móvil.

Entregables: Entorno del servidor operativo con sus servicios en ejecución, base de datos con su esquema versionado y API REST de consulta disponible.

4.4.7.7 Estado actual del Product Backlog

Tabla 19 Estado del Product Backlog al cierre del Sprint 1

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Infraestructura base del backend (servidor, contenedores Docker)	HU03	Alta	Hecho
2	Modelo de datos y migraciones (devices, readings, audit_log)	HU03	Alta	Hecho
3	API REST de consulta de lecturas y dispositivos	HU04	Alta	Hecho
4	Recepción de telemetría vía MQTT (worker de ingesta)	HU03	Alta	Por hacer
5	Validación de datos y deduplicación de lecturas	HU03	Alta	Por hacer

Nro.	Tarea	Historia de usuario	Prioridad	Estado
6	Auditoría de lotes con huella de integridad (SHA-256)	HU03	Media	Por hacer
7	Cifrado de extremo a extremo (AES-256-GCM)	HU03	Alta	Por hacer
8	Detección de manipulación de mensajes	HU03	Alta	Por hacer
9	Autenticación MQTT y control de acceso (ACL)	HU03	Alta	Por hacer
10	Transporte cifrado (TLS 1.3)	HU03	Alta	Por hacer
11	Soporte de umbrales para alerta (datos para HU05)	HU05	Media	Por hacer
12	Pruebas automatizadas y documentación técnica	HU03 y HU04	Media	Por hacer
13	Cifrado AES-256-GCM en el firmware de la estación ESP32	HU03	Alta	Por hacer
14	Endpoint de ingestión HTTP para el gateway remoto	HU03	Alta	Por hacer

Nro.	Tarea	Historia de usuario	Prioridad	Estado
15	Autenticación de la API mediante API Key	HU03	Alta	Por hacer
16	Exposición del servidor con dominio propio y túnel seguro	HU04	Alta	Por hacer
17	Firmware del gateway: recepción LoRa y reenvío HTTPS	HU03	Alta	Por hacer
18	Persistencia del número de secuencia en memoria no volátil	HU03	Media	Por hacer
19	Integración y validación del sistema con datos reales	HU03, HU04	Alta	Por hacer

4.4.8 Sprint 2: Ingesta MQTT, validación y auditoría

4.4.8.1 Planificación del Sprint 2

Tabla 20 Recepción de telemetría MQTT validación deduplicacion y auditoria

SPRINT 2: Recepción de telemetría MQTT validación deduplicacion y auditoria			
Duración:	13/06/2026 – 14/06/2026 (2 días)	Prioridad:	Alta
Objetivo:	Implementar el componente que recibe la telemetría desde el bróker MQTT, valida cada lectura evita duplicados y registra la auditoria de cada lote procesando, garantizando que no se pierdan registros ante reconexiones.		
Historias de usuario:	HU03: Sincronización y almacenamiento centralizado		
Tareas de desarrollo:	- Implementación del worker MQTT con suscripción a los tópicos y reconexión automática - Validación de rangos de las lecturas		

SPRINT 2: Recepción de telemetría MQTT validación deduplicacion y auditoria	
	• Deduplicacion mediante restricción de unicidad • Registro de auditoria por lote con huella SHA-256 • Pruebas del flujo completo de ingesta
Plan de entrega:	• Worker MQTT. Suscripción al bróker con QoS1 y reconexión automática ante caídas Tiempo de desarrollo: (6)h fechas: (13/06/2026) • Validación y deduplicacion. Comprobación de rangos y descarte de lecturas repetidas. Tiempo de desarrollo: (5)h – Fechas: (13/06/2026 – 14/06/2026) • Auditoria. Registro de una entrada por lote con su resultado y huella de integridad. Tiempo de desarrollo: (4) h — Fechas: (14/06/2026) • Pruebas de ingesta. Verificación del flujo (lote valido y lote duplicado Tiempo de desarrollo: (3) h — Fechas (14/06/2026)

4.4.8.2 Sprint Backlog Sprint 2

Tabla 21 Sprint Backlog del Sprint 2

Historia de usuario	Tarea	Duración estimada
HU03	Recepción de telemetría vía MQTT (worker de ingesta)	7 horas
HU03	Validación de datos y deduplicacion de lecturas	6 horas
HU03	Auditoria de lotes con huella de integridad (SHA-256)	5 horas
	Total	18 horas

4.4.8.3 Desarrollo del Sprint 2

4.4.8.3.1 Funcionalidad de ingesta de telemetría, validación y auditoría

Descripción: En esta funcionalidad se implementó el componente que recibe la telemetría desde el bróker MQTT. El worker se suscribe a los tópicos de las estaciones, valida cada lectura recibida, descarta los lotes duplicados que llegan tras una reconexión y este la registra en ;a auditoria el resultado es de un lote procesado junto con una huella de seguridad SHA-256

Ilustración 24 Ingesta correcta de un lote de telemetría

```
david@server:~/Back-App-Monitoreo$ docker compose run --rm --entrypoint python worker /scripts/publish_encrypted_test.py --sequences 900,901
[+] 2/25 2/22
+ Container monitoreo-db Running
+ Container monitoreo-mqtt Running
+ Container monitoreo-db Waiting
+ Container monitoreo-db Healthy
Container back-app-monitoreo-worker-run-8dde6d528e7b Creating
Container back-app-monitoreo-worker-run-8dde6d528e7b Created
Conectado a mosquito:8883 como esp32-01 (client_id=esp32-sim-01, TLS=True)...
Publicando lote cifrado en vivo/el-carmen/esp32-01/telemetria (QoS 1), secuencias=(900, 901)...
Publicado OK: deviceId=esp32-01 lecturas=2
david@server:~/Back-App-Monitoreo$ docker compose logs --tail=8 worker
monitoreo-worker | 2026-06-19 04:12:04,845 [INFO] __main__: Conectado a Mosquitto (mosquitto:8883) como client_id=monitoreo-worker
monitoreo-worker | 2026-06-19 04:12:04,846 [INFO] __main__: Suscrito a vivo/el-carmen/telemetria (QoS 1, mid=3)
monitoreo-worker | 2026-06-19 04:12:54,578 [INFO] app.ingest: Lote recibido: tópico=vivo/el-carmen/esp32-01/telemetria zona=el-carmen dispositivo_tópico=esp32-01 byt
es=549
monitoreo-worker | 2026-06-19 04:12:54,775 [INFO] app.ingest: Ingesta completada: dispositivo=esp32-01 insertados=2 duplicados=0 total=2 resultado_auditoria=ok
monitoreo-worker | 2026-06-19 04:12:54,775 [INFO] app.ingest: Auditoria registrada: dispositivo=esp32-01 resultado=ok detalle=insertados=2 duplicados=0 total=2 sha256=
324f6490668b9a3...
monitoreo-worker | 2026-06-24 19:20:15,156 [INFO] app.ingest: Lote recibido: tópico=vivo/el-carmen/esp32-01/telemetria zona=el-carmen dispositivo_tópico=esp32-01 byt
es=549
monitoreo-worker | 2026-06-24 19:20:15,224 [INFO] app.ingest: Ingesta completada: dispositivo=esp32-01 insertados=2 duplicados=0 total=2 resultado_auditoria=ok
monitoreo-worker | 2026-06-24 19:20:15,224 [INFO] app.ingest: Auditoria registrada: dispositivo=esp32-01 resultado=ok detalle=insertados=2 duplicados=0 total=2 sha256=
b1760616b99469...
david@server:~/Back-App-Monitoreo$
```

Aquí el worker recibe el lote desde el bróker MQTT, lo valida y almacena sus lecturas. Este registro indica el número de lecturas insertadas y el resultado ok de la auditoria

Ilustración 25 Registro de la auditoria de los lotes procesados

```
david@server:~/Back-App-Monitoreo$ cd ~/Back-App-Monitoreo && docker compose exec db psql -U monitoreo -d monitoreo -c "SELECT device_external_id, source, result, LEFT
(payload_sha256, 14) AS huella, created_at FROM audit_log ORDER BY created_at DESC LIMIT 6;"
 device_external_id | source | result | huella | created_at
-----
esp32-01 | http | ok | da472f49a600cc08 | 2026-07-10 16:37:31.699742+00
esp32-01 | http | ok | a6cab9231090dabd | 2026-07-10 16:36:56.648431+00
esp32-01 | http | ok | 043a68d927f0d3f0 | 2026-07-10 16:36:18.521702+00
esp32-01 | http | ok | a312510a1c15eb17 | 2026-07-10 16:35:42.135449+00
esp32-01 | http | ok | 3db8e446ab293e94 | 2026-07-10 16:35:07.603648+00
esp32-01 | http | ok | 497e91aac030b667 | 2026-07-10 14:54:57.423236+00
esp32-01 | http | ok | 2bb8436c1b4286d6 | 2026-07-10 14:53:36.425503+00
esp32-01 | http | ok | 32879a31ac1a3ba | 2026-07-10 14:52:58.10905+00
(6 rows)
```

Aquí cada lote recibido queda registrado en la tabla `audit_log` con el identificador de la estación, el origen, el resultado (`ok`, `duplicate`, `invalid_payload` o `decrypt_failed`) y una huella SHA-256 del contenido lo que permite la trazabilidad completa de la información ingresada al sistema.

```
for reading in batch.readings:
    stmt = (
        insert(Reading)
        .values(
            device_id=device_id,
            sequence=reading.sequence,
            temperature=reading.temperature,
            humidity=reading.humidity,
            rssi=reading.rssi,
            sensor_timestamp=reading.sensorTimestamp,
            received_at_gateway=reading.receivedAtGateway,
            transport=transport,
        )
        .on_conflict_do_nothing(constraint="uq_readings_device_sequence")
        .returning(Reading.id)
    )
    row_id = session.scalar(stmt)
    if row_id is None:
        duplicates += 1
```

Esta porción realiza la inserción de las lecturas en la base de datos aplicando la deduplicación. El método `on_conflict_do_nothing`, junto con los campos `device_id` y `sequence`, indica que si llega una lectura cuya combinación de estación y secuencia ya existe en la tabla, la base de datos no la inserta de nuevo en lugar de generar un error. Esto es necesario porque el protocolo MQTT con calidad de servicio QoS 1 garantiza la entrega "al menos una vez", lo que puede provocar que un mismo lote llegue repetido tras una reconexión. La cláusula `.returning(Reading.id)` hace que la instrucción devuelva el identificador de la lectura únicamente cuando esta se insertó realmente. Por ello si `row_id` resulta ser `None` esto significa que la lectura ya existía y se contabiliza como duplicada; en caso contrario se contabiliza como insertada así estos dos controladores son los que ya finalmente se registran en la auditoria del lote

4.4.8.4 Sprint Review Sprint 2

Al cierre del segundo Sprint se entregó el worker de ingesta en funcionamiento se suscribe al bróker MQTT, validando cada lectura recibida así descartamos los lotes duplicados y registrando la auditoría de cada operación con su huella de integridad.

Ilustración 26 Ingesta correcta de un lote de telemetría

```
David@server:~/Back-App-Monitoreo$ cd ~/Back-App-Monitoreo && docker compose logs api | grep -i "Ingesta completada" | tail -5
monitoreo-api | INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=1 duplicados=0 total=1 resultado=ok fuente=http
monitoreo-api | INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=1 duplicados=0 total=1 resultado=ok fuente=http
monitoreo-api | INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=1 duplicados=0 total=1 resultado=ok fuente=http
monitoreo-api | INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=1 duplicados=0 total=1 resultado=ok fuente=http
```

Ilustración 27 Rechazo de un lote duplicado

```
David@server:~/Back-App-Monitoreo$ cd ~/Back-App-Monitoreo && docker compose logs api | grep -i "duplicate\|duplicados=1" | tail -5
monitoreo-api | INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=0 duplicados=1 total=1 resultado=duplicate fuente=http
monitoreo-api | INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=0 duplicados=1 total=1 resultado=duplicate fuente=http
monitoreo-api | INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=0 duplicados=1 total=1 resultado=duplicate fuente=http
monitoreo-api | INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=0 duplicados=1 total=1 resultado=duplicate fuente=http
```

Ilustración 28 Registro de auditoría de los lotes procesados

```
David@server:~/Back-App-Monitoreo$ docker compose exec db psql -U monitoreo -d monitoreo -c "SELECT device_external_id, source, result, LEFT(payload_sha256, 16) AS hsa
lla, created_at FROM audit_log ORDER BY created_at DESC LIMIT 5;"
 device_external_id | source | result | hsa11a | created_at
-----
esp32-01            | http  | ok     | f586f3b043797642 | 2024-07-13 08:48:51.466321+00
esp32-01            | http  | ok     | 0b9743d180831616 | 2024-07-13 08:47:04.327202+00
esp32-01            | http  | ok     | abcdc78a72a0e037 | 2024-07-13 08:46:27.77145+00
esp32-01            | http  | ok     | bed59bd4208232cd | 2024-07-13 08:45:15.264757+00
esp32-01            | http  | ok     | c498b3e111d05368 | 2024-07-13 08:44:39.3863+00
esp32-01            | http  | ok     | 5f5cfe5430a9b7de | 2024-07-13 08:44:02.284885+00
esp32-01            | http  | ok     | 24394891ec996299 | 2024-07-13 08:42:50.367963+00
esp32-01            | http  | ok     | 900a4ba30b81242e | 2024-07-13 08:41:39.090233+00
(5 rows)
```

- **Retroalimentación establecida**

Durante las pruebas de ingesta se detectó que ante una reconexión del cliente al bróker, un mismo lote podría almacenarse dos veces. Pues la causa se identificó en la naturaleza del protocolo MQTT con calidad de servicio QoS1 que garantiza la entrega “al menos una vez” lo que implica que en un mensaje puede reintegrarse si no se confirma su recepción. Si un mecanismo de control, esto habría producido lecturas duplicadas en el histórico y comprometiendo la fiabilidad de los datos, además que se observó igualmente que no existía una trazabilidad de los lotes rechazados cuando un mensaje se descartaba por ser invalido, no quedaba constancia alguna del hecho, lo que impedía diagnosticar incidencias o detectar comportamientos fuera de lugar o incoherentes.

- **Mejoras realizadas**

Se incorporo la deduplicacion a nivel de base de datos mediante una restricci3n de unicidad sobre la combinaci3n de estaci3n y numero de secuencia (uq_readings_device_sequence), aplicada con la cl1usula on_conflict_do_nothing as3 de este m3todo si un lote llega a repetirse la base de datos simplemente no lo va a insertar, en lugar de generar un error. La deduplicacion queda as3 garantizada por el motor de datos y no depende de la l3gica de la aplicaci3n. Adem1s se implement3 el registro de auditoria por lote que almacena el identificado de la estaci3n el origen del mensaje y el resultado del procesamiento (ok, duplicate, invalid_payload o decrypt_failed) y una huella SHA-256 del contenido recibido as3 con esto todo lote que ingresa al sistema sea aceptado o no queda registrado as3 garantizando la trazabilidad de la informaci3n

4.4.8.5 Pruebas del Sprint 2, Caja Blanca

Tabla 22 Pruebas de caja blanca del Sprint 2 – Ingesta MQTT, validaci3n y auditor3a

M3todo	Acci3n esperada	Acci3n obtenida	Observaci3n
Ingesta de un lote v1lido por MQTT	Almacenar la lectura y auditar el resultado como "ok"	La lectura se almacen3 y se audit3 como "ok"	Ingesta verificada.
Reenv3o del mismo lote tras una reconexi3n (QoS 1)	No duplicar la lectura en el hist3rico	Al inicio la lectura se almacen3 dos veces, porque QoS 1 garantiza la entrega "al menos una vez". Tras incorporar la restricci3n de unicidad con ON CONFLICT DO NOTHING, el reenv3o dej3 de generar duplicados	Falla detectada y corregida; deduplicaci3n verificada.

Método	Acción esperada	Acción obtenida	Observación
Ingesta de un lote inválido (fuera de rango)	Rechazar el lote y dejar constancia en la auditoría	Inicialmente el lote se descartaba sin registrar el hecho. Tras implementar el registro de auditoría por lote, el rechazo quedó auditado como "invalid_payload" con su huella SHA-256	Falla detectada y corregida; trazabilidad verificada.

4.4.8.6 Incremento y Entregable del Sprint 2

Al finalizar el Sprint 2 se completaron satisfactoriamente los siguientes procesos:

- Implementación del worker de ingesta suscrito al broker MQTT, que procesa de forma continua los lotes de telemetría recibidos.
- La validación de las lecturas recibidas conforme a los rangos físicos admitidos por el sensor.
- Deduplicación de las lecturas a nivel de base de datos, que impide almacenar dos veces un mismo lote ante reenvíos del protocolo.
- Registro de auditoría por el lote con el identificador de la estación, el origen del mensaje además el resultado del procesamiento y una huella SHA-256 del contenido que se ha recibido.

Entregables: Worker de ingesta operativo mecanismo de deduplicación garantizado por el motor de base de datos y bitácora de auditoría de todos los lotes que se han procesado.

4.4.8.7 Estado actual del Product Backlog

Tabla 23 Estado del Product Backlog al cierre del Sprint 2

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Infraestructura base del backend (servidor, contenedores Docker)	HU03	Alta	Hecho
2	Modelo de datos y migraciones (devices, readings, audit_log)	HU03	Alta	Hecho
3	API REST de consulta de lecturas y dispositivos	HU04	Alta	Hecho
4	Recepción de telemetría vía MQTT (worker de ingesta)	HU03	Alta	Hecho
5	Validación de datos y deduplicación de lecturas	HU03	Alta	Hecho
6	Auditoría de lotes con huella de integridad (SHA-256)	HU03	Media	Hecho
7	Cifrado de extremo a extremo (AES-256-GCM)	HU03	Alta	Por hacer
8	Detección de manipulación de mensajes	HU03	Alta	Por hacer
9	Autenticación MQTT y control de acceso (ACL)	HU03	Alta	Por hacer
10	Transporte cifrado (TLS 1.3)	HU03	Alta	Por hacer
11	Soporte de umbrales para alerta (datos para HU05)	HU05	Media	Por hacer
12	Pruebas automatizadas y documentación técnica	HU03 y HU04	Media	Por hacer
13	Cifrado AES-256-GCM en el firmware de la estación ESP32	HU03	Alta	Por hacer
14	Endpoint de ingestión HTTP para el gateway remoto	HU03	Alta	Por hacer
15	Autenticación de la API mediante API Key	HU03	Alta	Por hacer
16	Exposición del servidor con dominio propio y túnel seguro	HU04	Alta	Por hacer
17	Firmware del gateway: recepción LoRa y reenvío HTTPS	HU03	Alta	Por hacer
18	Persistencia del número de secuencia en memoria no volátil	HU03	Media	Por hacer

Nro.	Tarea	Historia de usuario	Prioridad	Estado
19	Integración y validación del sistema con datos reales	HU03, HU04	Alta	Por hacer

4.4.9 Sprint 3: Seguridad – Cifrado extremo a extremo y transporte seguro

4.4.9.1 Planificación del Sprint 3

Tabla 24 Cifrado de extremo a extremo, detección de manipulación y transporte seguro

SPRINT 3: Cifrado de extremo a extremo, detección de manipulación y transporte seguro			
Duración:	15/06/2026 – 19/06/2026 (5 días)	Prioridad:	Alta
Objetivo:	Implementar el cifrado de extremo a extremo con AES-256-GCM la detección de manipulación de mensajes, la autenticación del bróker con control de acceso por dispositivo y el transporte cifrado mediante TLS 1.3 garantizando la confidencialidad e integridad de la información.		
Historias de usuario:	HU03: Sincronización y almacenamiento centralizado		
Tareas de desarrollo:	• Implementación del descifrado AES-256-GCM en el worker • Gestión de llaves por dispositivo • Detección de manipulación (rechazo y auditoria como decrypt_failed) • Autenticación MQTT y listas de control de acceso (ACL) • Configuración de TLS 1.3 con certificados • Elaboración del contrato de cifrado para el firmware		
Plan de entrega:	• Descifrado AES-256-GCM. Implementación del descifrado y verificación del tag de autenticación Tiempo de desarrollo: (8) h — Fechas: (15/06/2026 – 16/06/2026) • Gestión de llaves. Llave simétrica por dispositivo, gestionada de forma segura Tiempo de desarrollo: (4) h — Fechas: (16/06/2026) • Detección de manipulación. Usuarios del bróker y restricción de publicación por tópico Tiempo de desarrollo: (5) h — Fechas: (17/06/2026) • Autenticación MQTT y ACL. Usuarios del bróker y restricción de publicación por tópico Tiempo de desarrollo: (4) h — Fechas: (18/06/2026) • Transporte TLS 1.3. Generación de certificados y configuración del canal seguro Tiempo de desarrollo: (6) h — Fechas: (18/06/2026 -19/06/2026)		

SPRINT 3: Cifrado de extremo a extremo, detección de manipulación y transporte seguro	
	• Contrato de cifrado. Documento de especificación para el equipo de firmware (ESP32) Tiempo de desarrollo: (3) h — Fechas: (19/06/2026)

4.4.9.2 Sprint Backlog Sprint 3

Tabla 25 Estado del Product Backlog al cierre del Sprint 3

Historia de usuario	Tarea	Duración estimada
HU03	Cifrado de extremo a extremo (AES-256-GCM)	15 horas
HU03	Detección de manipulación de mensajes	5 horas
HU03	Atención MQTT y control de acceso (ACL)	4 horas
HU03	Transporte cifrado (TLS 1.3)	6 horas
	Total	30 horas

4.4.9.3 Desarrollo del Sprint 3

4.4.9.3.1 Funcionalidad de cifrado de extremo a extremo, detección de manipulación y transporte seguro

En esta funcionalidad se implementó la seguridad de la información del sistema. Estas lecturas viajan cifradas con AES-256-GCM desde su origen, de modo que el contenido resulta ser ilegible durante todo el trayecto. Esto hace que el sistema verifique además de la autenticidad de cada lote por ejemplo si un mensaje fue alterado durante la transmisión, la verificación criptográfica falla, el lote se rechaza y el intento queda registrado en la auditoría. Además, se configuro la autenticación del bróker con control de acceso por el dispositivo y el transporte cifrado mediante TLS 1.3

Ilustración 29 Descifrado correcto de los lotes cifrados

```

dev@server1:~/Back-App-Monitoreo$ cd ~/Back-App-Monitoreo && docker compose logs api | grep -i "ingesta completada" | tail -10
monitoreo-api INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=1 duplicados=0 total=1 resultado=ok fuente=http
monitoreo-api INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=1 duplicados=0 total=1 resultado=ok fuente=http
monitoreo-api INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=1 duplicados=0 total=1 resultado=ok fuente=http
monitoreo-api INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=1 duplicados=0 total=1 resultado=ok fuente=http
monitoreo-api INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=1 duplicados=0 total=1 resultado=ok fuente=http
monitoreo-api INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=1 duplicados=0 total=1 resultado=ok fuente=http
monitoreo-api INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=1 duplicados=0 total=1 resultado=ok fuente=http
monitoreo-api INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=1 duplicados=0 total=1 resultado=ok fuente=http
monitoreo-api INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=1 duplicados=0 total=1 resultado=ok fuente=http
monitoreo-api INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=1 duplicados=0 total=1 resultado=ok fuente=http

```

Aquí el sistema recibe los sobres cifrados, verifica su autenticidad, los descifra con la llave del dispositivo y almacena las lecturas. La auditoría registra el resultado que marca como OK en cada caso

Ilustración 30 Rechazo de un lote manipulado

```
david@server:~/Back-App-Monitoreo$ curl -s -w "%{HTTP_CODE}" -X POST "https://api.darigo.com/api/v1/telemetry/ingest" \
-H "X-API-Key: 2944ac0994b7a5907d51565adeeb9c3cc46951b48b0345efed7a96c0db43df" \
-H "Content-Type: application/json" \
-d '{"*"}',{"deviceId":"esp32-01","alg":"AES-256-GCM","payload":"XXXXXXXXXX"}
{"status":"decrypt_failed","detail":"autenticación fallida"}
HTTP/400
```

Esto sucede al alterar deliberadamente el contenido cifrado, la verificación del tag de autenticación falla y el sistema rechaza el lote dando con el resultado decrypt_failed, sin almacenar ninguna lectura esto demuestra que el cifrado autenticado AES-256-GCM detecta cualquier modificación de la información en el tránsito: aquí no es posible alterar un dato sin poseer la llave

Ilustración 31 Auditoría del intento de manipulación

```
david@server:~/Back-App-Monitoreo$ docker compose exec db psql -U monitoreo -d monitoreo -c "SELECT device_external_id, source, result, detail, created_at FROM audit_log WHERE result = 'decrypt_failed' ORDER BY created_at DESC LIMIT 5;"
 device_external_id | source | result | detail | created_at
-----
 esp32-01           | http  | decrypt_failed | autenticación fallida | 2026-07-13 07:07:40.929071+00
 esp32-01           | http  | decrypt_failed | sobre inválido       | 2026-07-09 02:39:34.886221+00
 esp32-01           | http  | decrypt_failed | sobre inválido       | 2026-07-09 02:37:50.430145+00
 esp32-01           | http  | decrypt_failed | sobre inválido       | 2026-07-09 02:37:40.504093+00
 esp32-01           | http  | decrypt_failed | sobre inválido       | 2026-07-09 02:37:19.440591+00
(5 rows)
```

Se muestra que el sistema no solo rechaza el lote alterado si no que deja constancia del intento en el registro de auditoría, con la fecha el dispositivo y el motivo del rechazo, garantizando la trazabilidad ante incidentes de seguridad

Ilustración 32 Conexión al bróker MQTT mediante TLS 1.3

```
david@server:~/Back-App-Monitoreo$ openssl s_client -connect localhost:8883 -tls_3 </dev/null 2>/dev/null | grep -i "protocol\|cipher"
New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
```

Este es el canal de comunicación con el bróker opera sobre TLS 1.3 la versión vigente del protocolo, lo que protege el tráfico frente a interceptación y suplantación

```
def decrypt_envelope(envelope: dict[str, Any], device_keys: dict[str, bytes]) -> dict[str, Any]:
    """Descifra un sobre JSON y devuelve el dict del lote en claro."""
    if not _is_encrypted_envelope(envelope):
        raise DecryptError("sobre inválido")

    device_id = envelope.get("deviceId")
    if envelope.get("alg") != ALGORITHM:
        raise DecryptError("sobre inválido")

    key = get_device_key(device_keys, device_id.strip())

    blob = base64.b64decode(envelope["payload"], validate=True)
    if len(blob) < NONCE_SIZE + TAG_SIZE + 1:
        raise DecryptError("sobre inválido")

    nonce = blob[:NONCE_SIZE]
    ciphertext_and_tag = blob[NONCE_SIZE:]

    try:
        plaintext = AESGCM(key).decrypt(nonce, ciphertext_and_tag, _aad(device_id.strip()))
    except Exception as exc:
        raise DecryptError("autenticación fallida") from exc

    return json.loads(plaintext.decode("utf-8"))
```

La función `decrypt_envelope` recibe el sobre cifrado y lo abre en cuatro pasos primero valida la estructura del sobre esto comprueba que contenga los campos esperados y que el algoritmo declarado sea AES-256-GCM para después recuperar la llave correspondiente al dispositivo mediante `get_device_key` para cada estación posee su propia llave simetría de 32 bytes gestionada fuera del repositorio. Después separa el contenido binario el campo `payload` viaja codificado en Base64 y una vez decodificado se divide en el nonce de 12 bytes que encabeza el bloque y el resto que contiene el texto cifrado junto con el tag de autenticación de 16 bytes.

Finalmente, AESGCM(key).decrypt(...) realiza el descifrado y la verificación de autenticidad en una sola operación. El identificador del dispositivo se pasa como dato autenticado adicional (AAD) así se vincula criptográficamente el mensaje a su emisor, de modo que un sobre no puede atribuirse a otra estación. Aquí reside la propiedad esencial del cifrado autenticado si un solo bit del contenido fue alterado durante la transmisión la verificación del tag falla y la operación lanza una excepción. El lote se rechaza sin almacenar ninguna lectura y el intento queda registrado en la auditoria como decrypt_failed. No es posible modificar el contenido sin poseer la llave

4.4.9.4 Sprint Review Sprint 3

Descripción de la funcionalidad entregada

Al cierre del tercer Sprint se entregó la seguridad de la información implementada que es el descifrado autenticado AES-256-GCM con gestión de llaves por dispositivo, detección y auditoria de mensajes manipulados como la autenticación del bróker con control de acceso y el transporte cifrado con TLS 1.3

Ilustración 33 Descifrado correcto de los lotes cifrados

```
david@dsrver:~/Back-App-Monitoreo$ cd ~/Back-App-Monitoreo && docker compose logs api | grep -i "resultado=ok" | tail -5
monitoreo-api | INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=1 duplicados=0 total=1 resultado=ok fuente=http
monitoreo-api | INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=1 duplicados=0 total=1 resultado=ok fuente=http
monitoreo-api | INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=1 duplicados=0 total=1 resultado=ok fuente=http
monitoreo-api | INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=1 duplicados=0 total=1 resultado=ok fuente=http
monitoreo-api | INFO:monitoreo_ingest.service:Ingesta completada: dispositivo=esp32-01 insertados=1 duplicados=0 total=1 resultado=ok fuente=http
```

Ilustración 34 Rechazo de un lote manipulado

```
david@dsrver:~/Back-App-Monitoreo$ curl -s -w "%{http_code}" -X POST "https://api.dsrge.com/api/v1/telemetry/ingest" \
-H "X-API-Key: d344ee0896b7a9907d5156adeeb9c3c646951b68b0345fed7a96c8b43df" \
-H "Content-Type: application/json" \
-d '{"device_id": "esp32-01", "alg": "AES-256-GCM", "payload": "XXXXXX0wafPQeeG9GBhyTyzCA6W9kFA/WirGu6Bz3fOnAKIFXZz7rrX0wTNE88iftr8NOopocfgFahOulrWUz3XoTrNLSy8WYJmIvqm8Z"}'
{"status": "decrypt_failed", "detail": "autenticación fallida"}
HTTP/1.1 400
```

Ilustración 35 Auditoria del intento de manipulación

```
david@dsrver:~/Back-App-Monitoreo$ docker compose exec db psql -U monitoreo -d monitoreo -c "SELECT device_external_id, source, result, detail, created_at FROM audit_log WHERE result = 'decrypt_failed' ORDER BY created_at DESC LIMIT 5;"
 device_external_id | source | result | detail | created_at
-----
 esp32-01 | http | decrypt_failed | autenticación fallida | 2026-07-14 15:51:05.59179+00
 esp32-01 | http | decrypt_failed | autenticación fallida | 2026-07-13 07:07:40.525071+00
 esp32-01 | http | decrypt_failed | sobre inválido | 2026-07-09 02:39:34.858221+00
 esp32-01 | http | decrypt_failed | sobre inválido | 2026-07-09 02:37:50.630145+00
 esp32-01 | http | decrypt_failed | sobre inválido | 2026-07-09 02:37:40.804093+00
(5 rows)
```

Ilustración 36 Conexión al bróker MQTT mediante TLS 1.3

```
david@dsrver:~/Back-App-Monitoreo$ openssl s_client -connect localhost:8883 -tls1_3 </dev/null 2>/dev/null | grep -i "protocol\|cipher"
New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
david@dsrver:~/Back-App-Monitoreo$
```

- **Retroalimentación establecida**

Al habilitar el transporte seguro, la conexión al bróker mediante TLS fallo con un error de discrepancia de nombre de host (hostname mismatch). El certificado había sido generado únicamente para el nombre del contenedor así por lo que las conexiones dirigidas al servidor mediante su dirección de red eran rechazadas por el cliente al no poder validar la identidad del servidor. Esta incidencia, aunque de configuración resultaba bloqueante sin huella ahora ningún dispositivo podía establecer un canal seguro. Esto hizo detectar además que el sistema en su configuración inicial aceptaba lotes sin cifrar. Pues bien, esto facilitaba las pruebas durante el desarrollo en un despliegue real habría permitido que un dispositivo enviara telemetría en claro, anulando la garantía de confidencialidad que el sistema pretende ofrecer

- **Mejoras realizadas**

Se regenero el certificado del bróker incluyendo la extensión SAN (Subject Alternative Name) declarando en el todos los nombres y direcciones por los que el servidor puede ser alcanzado. Con ello la validación de identidad se completa correctamente y el canal TLS 1.3 se establece sin errores. La incidencia dejo constancia de un principio de diseño: el certificado debe emitirse contemplando todas las rutas de acceso previas y no solo la utilizada durante el desarrollo. Se incorporó la política REQUIRE_ENCRYPTION activa en el entorno de producción, que rechaza todo lote que no venga cifrado. De este modo el cifrado deja de ser una opción y pasa a ser un requisito del sistema un dispositivo que intente enviar telemetría en claro es rechazado y el intento queda auditado.

4.4.9.5 Pruebas del Sprint 3, Caja Blanca

Tabla 26 Pruebas de caja blanca del Sprint 3 – Cifrado extremo a extremo y transporte seguro

Método	Acción esperada	Acción obtenida
Descifrado de un lote cifrado válido (AES-256-GCM)	Recuperar el contenido y almacenarlo	El contenido se descifró y almacenó correctamente
Recepción de un lote	Rechazar el lote y auditar "decrypt_failed"	El lote se rechazó y se auditó como "decrypt_failed"; no se almacenó

Método	Acción esperada	Acción obtenida
manipulado (un byte alterado)		
Conexión al bróker mediante TLS 1.3	Establecer el canal seguro	Al inicio la conexión falló con error de discrepancia de nombre de host (hostname mismatch), pues el certificado solo declaraba el nombre del contenedor. Tras regenerar el certificado con la extensión SAN, el canal TLS 1.3 se estableció sin errores
Envío de un lote sin cifrar en producción	Rechazar el lote	En la configuración inicial el sistema aceptaba lotes sin cifrar. Tras activar la política REQUIRE_ENCRYPTION, todo lote sin cifrar se rechaza y se audita

4.4.9.6 Incremento y Entregable del Sprint 3

Al finalizar el Sprint 3 se completaron de manera satisfactoria las siguientes cosas:

- Implementación del descifrado autenticado AES-256-GCM con gestión de llaves por dispositivo.
- La detección y auditoria de mensajes manipulados que impide el almacenamiento de datos alterados.
- Autenticación de los clientes del broker MQTT y el control del acceso por tópico mediante listas de control (ACL)
- Habilitación del transporte cifrado TLS 1.3 y adopción de la política que exige el cifrado de todo lote entrante

Entregables: La capa de seguridad de la información operativa con el cifrado de extremo a extremo canal de transporte cifrado, control de acceso y auditoria de los intentos de manipulación.

4.4.9.7 Estado del Sprint Backlog 3

Tabla 27 Estado del Product Backlog al cierre del Sprint 3

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Infraestructura base del backend (servidor, contenedores Docker)	HU03	Alta	Hecho
2	Modelo de datos y migraciones (devices, readings, audit_log)	HU03	Alta	Hecho
3	API REST de consulta de lecturas y dispositivos	HU04	Alta	Hecho
4	Recepción de telemetría vía MQTT (worker de ingesta)	HU03	Alta	Hecho
5	Validación de datos y deduplicación de lecturas	HU03	Alta	Hecho
6	Auditoría de lotes con huella de integridad (SHA-256)	HU03	Media	Hecho
7	Cifrado de extremo a extremo (AES-256-GCM)	HU03	Alta	Hecho
8	Detección de manipulación de mensajes	HU03	Alta	Hecho
9	Autenticación MQTT y control de acceso (ACL)	HU03	Alta	Hecho
10	Transporte cifrado (TLS 1.3)	HU03	Alta	Hecho
11	Soporte de umbrales para alerta (datos para HU05)	HU05	Media	Por hacer
12	Pruebas automatizadas y documentación técnica	HU03 y HU04	Media	Por hacer
13	Cifrado AES-256-GCM en el firmware de la estación ESP32	HU03	Alta	Por hacer
14	Endpoint de ingestión HTTP para el gateway remoto	HU03	Alta	Por hacer
15	Autenticación de la API mediante API Key	HU03	Alta	Por hacer
16	Exposición del servidor con dominio propio y túnel seguro	HU04	Alta	Por hacer
17	Firmware del gateway: recepción LoRa y reenvío HTTPS	HU03	Alta	Por hacer
18	Persistencia del número de secuencia en memoria no volátil	HU03	Media	Por hacer
19	Integración y validación del sistema con datos reales	HU03, HU04	Alta	Por hacer

4.4.10 Sprint 4: Pruebas automatizadas y documentación

4.4.10.1 Planificación del Sprint

Tabla 28 Pruebas automatizadas y documentación

SPRINT 4: Pruebas automatizadas y documentación			
Duración:	19/06/2026 – 21/06/2026 (3 días)	Prioridad:	Media
Objetivo:	Construir una batería de pruebas automatizadas que valide el cifrado, la ingesta, la deduplicación y la API elaborar también la documentación técnica del backend para su mantenimiento y defensa.		
Historias de usuario:	HU03 y HU04: (Validación integral del backend)		
Tareas de desarrollo:	• Implementación de pruebas automatizadas con pytest • Aislamiento del entorno de pruebas (base de datos de prueba dedicada) • Organización de las pruebas por módulos (cifrado, worker, API) • Elaboración de la documentación técnica (README.md)		
Plan de entrega:	• Pruebas de cifrado. Validación del cifrado, las llaves y la detección de manipulación. Tiempo de desarrollo: (6)h fechas: (19/06/2026) • Pruebas de ingesta y API. Validación del worker, la deduplicación y los endpoints Tiempo de desarrollo: (7)h – Fechas: (20/06/2026) • Documentación técnica. Redacción del README con arquitectura, modelo de datos los endpoints y decisiones de seguridad Tiempo de desarrollo: (5) h — Fechas (20/06/2026 – 21/06/2026)		

4.4.10.2 Sprint Backlog Sprint 4

Tabla 29 Estado del Product Backlog al cierre del Sprint 4

Historia de usuario	Tarea	Duración estimada
HU03, HU04	Batería de pruebas automatizadas	18 horas
	Total	18 horas

4.4.10.3 Desarrollo del Sprint 4

4.4.10.3.1 Funcionalidad de pruebas automatizadas y documentación técnica

En esta funcionalidad se construyó una batería de pruebas automatizadas que valida el comportamiento del backend sin intervención manual el cifrado y la detección de manipulación la ingesta de telemetría la deduplicación de lecturas y los endpoints de la API. Las pruebas se organizaron por módulo y se ejecutan sobre una base de datos dedicada, aislada de la base de producción además que se elaboró la documentación técnica del sistema para su mantenimiento y defensa

Ilustración 37 Ejecución de la batería de pruebas automatizadas

```
david@dsrserver:~/Back-App-Monitoreo$ cd ~/Back-App-Monitoreo && bash scripts/run_tests.sh
==> Levantando PostgreSQL (servicio db)...
[+] up 1/1
  Container monitoreo-db Running
==> Esperando PostgreSQL...
/var/run/postgresql:5432 - accepting connections
==> Creando base monitoreo_test (idempotente)...
Base monitoreo_test ya existe.
==> Ejecutando pytest en contenedor test...
[+] 1/1t 1/1l
  Container monitoreo-db Running
Container monitoreo-db Waiting
Container monitoreo-db Healthy
Container back-app-monitoreo-test-run-42fe2elad68a Creating
Container back-app-monitoreo-test-run-42fe2elad68a Created

=====
pytest: shared (cifrado)
PYTHONPATH=/app/shared
=====

===== test session starts =====
platform linux -- Python 3.12.13, pytest-9.1.1, pluggy-1.6.0 -- /usr/local/bin/python3.12
cachedir: .pytest_cache
rootdir: /app/shared
configfile: ../pytest.ini
plugins: anyio-4.14.1
collecting ... collected 18 items

shared/tests/test_crypto.py::test_roundtrip_encrypt_decrypt PASSED [ 5%]
shared/tests/test_crypto.py::test_fixed_test_vector_reproducible PASSED [ 11%]
shared/tests/test_crypto.py::test_tampered_tag_fails PASSED [ 16%]
shared/tests/test_crypto.py::test_wrong_aad_device_id PASSED [ 22%]
shared/tests/test_crypto.py::test_missing_device_key PASSED [ 27%]
shared/tests/test_crypto.py::test_plaintext_rejected_when_required PASSED [ 33%]
shared/tests/test_crypto.py::test_plaintext_allowed_when_optional PASSED [ 38%]
shared/tests/test_crypto.py::test_parse_device_keys_valid_single PASSED [ 44%]
shared/tests/test_crypto.py::test_parse_device_keys_valid_multiple PASSED [ 50%]
shared/tests/test_crypto.py::test_parse_device_keys_empty_string PASSED [ 55%]
shared/tests/test_crypto.py::test_parse_device_keys_invalid_format PASSED [ 61%]
shared/tests/test_crypto.py::test_parse_device_keys_invalid_base64 PASSED [ 66%]
shared/tests/test_crypto.py::test_parse_device_keys_wrong_key_size PASSED [ 72%]
shared/tests/test_crypto.py::test_invalid_envelope_structure PASSED [ 77%]
```

Ilustración 38 Ejecución de la batería de pruebas automatizadas

```
shared/tests/test_ingest_service.py::test_normalize_flat_with_device_id PASSED [ 83%]
shared/tests/test_ingest_service.py::test_normalize_flat_without_device_id_uses_expected PASSED [ 88%]
shared/tests/test_ingest_service.py::test_normalize_batch_with_readings_is_unchanged PASSED [ 94%]
shared/tests/test_ingest_service.py::test_normalize_flat_without_device_id_and_no_expected_returns_unchanged PASSED [100%]

===== 10 passed in 9.19s =====

pytest: worker (ingesta)
PYTHONPATH=/app/shared:/app/worker
===== test session starts =====
platform linux -- Python 3.12.13, pytest-9.1.1, pluggy-1.6.0 -- /usr/local/bin/python3.12
cachedir: .pytest_cache
rootdir: /app
configfile: pytest.ini
plugins: anyio-4.14.1
collecting ... collected 13 items

worker/tests/test_ingest.py::test_valid_batch_inserts_and_audits_ok PASSED [ 7%]
worker/tests/test_ingest.py::test_duplicate_batch_audits_duplicate PASSED [ 15%]
worker/tests/test_ingest.py::test_device_id_mismatch_topic_invalid_payload PASSED [ 23%]
worker/tests/test_ingest.py::test_out_of_range_temperature_invalid_payload PASSED [ 30%]
worker/tests/test_ingest.py::test_device_upsert_and_last_seen_at PASSED [ 38%]
worker/tests/test_ingest.py::test_exactly_one_audit_row_per_batch PASSED [ 46%]
worker/tests/test_ingest.py::test_invalid_topic_audits_once PASSED [ 53%]
worker/tests/test_ingest.py::test_encrypted_batch_ingest_ok PASSED [ 61%]
worker/tests/test_ingest.py::test_tampered_encrypted_decrypt_failed PASSED [ 69%]
worker/tests/test_schemas.py::test_reading_in_coerces_strings PASSED [ 76%]
worker/tests/test_schemas.py::test_reading_in_rejects_temperature_out_of_range PASSED [ 84%]
worker/tests/test_schemas.py::test_batch_rejects_empty_readings PASSED [ 92%]
worker/tests/test_schemas.py::test_batch_rejects_empty_device_id PASSED [100%]

===== 13 passed in 7.06s =====

pytest: api (REST)
PYTHONPATH=/app/shared:/app/api
===== test session starts =====
platform linux -- Python 3.12.13, pytest-9.1.1, pluggy-1.6.0 -- /usr/local/bin/python3.12
cachedir: .pytest_cache
```

Ilustración 39 Ejecución de la batería de pruebas automatizadas

```
rootdir: /app
configfile: pytest.ini
plugins: anyio-4.14.1
collecting ... collected 21 items

api/tests/test_ingest_route.py::test_ingest_encrypted_ok PASSED [ 4%]
api/tests/test_ingest_route.py::test_ingest_duplicate_returns_409 PASSED [ 9%]
api/tests/test_ingest_route.py::test_ingest_tampered_returns_decrypt_failed PASSED [ 14%]
api/tests/test_ingest_route.py::test_ingest_invalid_envelope_schema PASSED [ 19%]
api/tests/test_ingest_route.py::test_ingest_flat_esp32_reading_defaults_rssi PASSED [ 23%]
api/tests/test_ingest_route.py::test_ingest_flat_esp32_without_device_id_uses_envelope PASSED [ 28%]
api/tests/test_routes.py::test_health_ok PASSED [ 33%]
api/tests/test_routes.py::test_db_check_ok PASSED [ 38%]
api/tests/test_routes.py::test_telemetry_readings_contract PASSED [ 42%]
api/tests/test_routes.py::test_batch_size_clamp_999 PASSED [ 47%]
api/tests/test_routes.py::test_batch_size_clamp_zero PASSED [ 52%]
api/tests/test_routes.py::test_batch_size_default_empty PASSED [ 57%]
api/tests/test_routes.py::test_readings_between_filters_effective_time PASSED [ 61%]
api/tests/test_routes.py::test_devices_list_format PASSED [ 66%]
api/tests/test_routes.py::test_readings_between_start_after_end PASSED [ 71%]
api/tests/test_routes.py::test_readings_empty PASSED [ 76%]
api/tests/test_routes.py::test_internal_error_generic_response PASSED [ 80%]
api/tests/test_schemas.py::test_telemetry_out_coerces_string_numbers PASSED [ 85%]
api/tests/test_schemas.py::test_telemetry_out_defaults PASSED [ 90%]
api/tests/test_schemas.py::test_telemetry_out_rejects_invalid_sequence PASSED [ 95%]
api/tests/test_schemas.py::test_device_out_format PASSED [100%]

===== 21 passed in 11.05s =====

TOTAL: 52 passed
==> Pruebas completadas.
```

Las 52 pruebas del sistema se ejecutan de forma aislada por componente y aprueban en su totalidad 18 corresponden al módulo de criptografía y los esquemas de ingesta, 13 al worker de ingesta MQTT y la 21 a la API REST y al Endpoint de ingestión HTTP

```
def test_tampered_tag_fails(device_keys: dict[str, bytes]) -> None:
    # 1. Se cifra un lote válido
    envelope = encrypt_envelope(
        TEST_VECTOR_PLAINTEXT_BATCH, device_id="esp32-01", key=TEST_VECTOR_KEY
    )

    # 2. Se altera deliberadamente un byte del contenido cifrado
    blob = bytearray(base64.b64decode(envelope["payload"]))
    blob[20] ^= 0xFF
    envelope["payload"] = base64.b64encode(bytes(blob)).decode()

    # 3. Se exige que el sistema rechace el sobre manipulado
    with pytest.raises(DecryptError, match="autenticación fallida"):
        decrypt_envelope(envelope, device_keys)
```

Esta prueba automatiza el escenario de un ataque de manipulación de datos y verifica que el sistema lo detecte primero se cifra un lote válido con la llave del dispositivo, obtenido el sobre que viajara por la red. A continuación, se altera deliberadamente un solo byte del contenido cifrado: la operación `blob[20] ^= 0xFF` invierte todos los bits del byte situado en la posición 20 así logrando simular la acción de un atacante que intercepta el mensaje y lo modifica.

Finalmente, la instrucción `pytest.raises(DecryptError, match="autenticación fallida")` exige que el sistema lance la excepción al intentar descifrar el sobre alterado. Si el descifrado tuviera éxito el sistema lo va a aceptar como un dato manipulado y la prueba fallaría. Así la propiedad de seguridad más importante del sistema la imposibilidad de alterar una lectura sin poseer la llave queda verificada de forma automática en cada ejecución de la batería de pruebas así no depende de una comprobación manual.

4.4.10.4 Sprint Review – Sprint 4

4.4.10.4.1 Descripción de la funcionalidad entregada

Al cierre del cuarto Sprint se entregó la batería de pruebas automatizadas del backend se hizo 52 pruebas organizadas por modulo ejecutándose sobre una base de datos aislada junto con la documentación técnica del sistema

Ilustración 40 Ejecución de la batería de pruebas automatizadas (52 pruebas aprobadas)

```
api/tests/test_ingest_route.py::test_ingest_encrypted_ok PASSED [ 4%]
api/tests/test_ingest_route.py::test_ingest_duplicate_returns_409 PASSED [ 9%]
api/tests/test_ingest_route.py::test_ingest_tampered_returns_decrypt_failed PASSED [ 14%]
api/tests/test_ingest_route.py::test_ingest_invalid_envelope_schema PASSED [ 19%]
api/tests/test_ingest_route.py::test_ingest_flat_esp32_reading_defaults_rssi PASSED [ 23%]
api/tests/test_ingest_route.py::test_ingest_flat_esp32_without_device_id_uses_envelope PASSED [ 28%]
api/tests/test_routes.py::test_health_ok PASSED [ 33%]
api/tests/test_routes.py::test_db_check_ok PASSED [ 38%]
api/tests/test_routes.py::test_telemetry_readings_contract PASSED [ 42%]
api/tests/test_routes.py::test_batch_size_clamp_999 PASSED [ 47%]
api/tests/test_routes.py::test_batch_size_clamp_zero PASSED [ 52%]
api/tests/test_routes.py::test_batch_size_default_empty PASSED [ 57%]
api/tests/test_routes.py::test_readings_between_filters_effective_time PASSED [ 61%]
api/tests/test_routes.py::test_devices_list_format PASSED [ 66%]
api/tests/test_routes.py::test_readings_between_start_after_end PASSED [ 71%]
api/tests/test_routes.py::test_readings_empty PASSED [ 76%]
api/tests/test_routes.py::test_internal_error_generic_response PASSED [ 80%]
api/tests/test_schemas.py::test_telemetry_out_coerces_string_numbers PASSED [ 85%]
api/tests/test_schemas.py::test_telemetry_out_defaults PASSED [ 90%]
api/tests/test_schemas.py::test_telemetry_out_rejects_invalid_sequence PASSED [ 95%]
api/tests/test_schemas.py::test_device_out_format PASSED [100%]

===== 21 passed in 10.67s =====

TOTAL: 52 passed
==> Pruebas completadas.
```

- **Retroalimentación establecida**

Al ejecutar la batería completa de pruebas se produjo un conflicto entre los módulos del proyecto ya que los paquetes de la API y del worker comparten nombres internos por lo que al importarse en una misma ejecución se sobrescribían mutuamente provocando fallos que no correspondían a errores reales del código si no a la forma de ejecutar las pruebas.

Se observó también que las pruebas se ejecutaban contra la base de datos desarrollo lo que suponía un riesgo evidente ya que era una prueba destructiva podría haber alterado o eliminado datos reales del sistema

- **Mejoras realizadas**

Se reorganizó la ejecución de las pruebas en tres pasadas independientes una por modulo: Criptografía, worker y API cada una con su propia ruta de importación. Esto elimina el conflicto de nombres y permite además identificar con precisión que componente falla cuando una prueba no aprueba así se aisló el entorno de pruebas mediante una base de datos dedicada (monitoreo_test), creada de forma automática antes de cada ejecución. La base de producción queda así protegida y ninguna prueba puede alterarla por destructiva que sea

4.4.10.5 Pruebas del Sprint, Caja blanca

Tabla 30 Pruebas de caja blanca del Sprint 4 – Pruebas automatizadas y documentación

Método	Acción esperada	Acción obtenida	Observación
Ejecución de la batería completa de pruebas automatizadas	Ejecutar todas las pruebas y que aprueben	Al ejecutar todo junto se produjo un conflicto de nombres entre los paquetes de la API y el worker, que provocaba fallos ajenos al código. Tras reorganizar la ejecución en tres pasadas independientes (criptografía, worker y API), las 52 pruebas aprobaron	Falla detectada y corregida; suite verificada (52/52).
Ejecución de las pruebas sobre la base de datos	Ejecutar sin afectar los datos reales	Inicialmente las pruebas corrían contra la base de desarrollo, con riesgo de alterar datos reales. Tras aislar el entorno con una base dedicada (monitoreo_test) creada antes de cada ejecución, la base de producción quedó protegida	Falla detectada y corregida; aislamiento del entorno verificado.

4.4.10.6 Incremento y Entregable del Sprint

Al finalizar el Sprint 4 se completaron satisfactoriamente los siguientes procesos:

- Construcción de la batería de pruebas automatizadas del backend organizada por modulo (criptografía, worker de ingesta y API)
- Aislamiento del entorno de pruebas mediante una base de datos dedicada creada automáticamente antes de cada ejecución así protege los datos de producción.
- Habilitación de la consulta de lecturas por rango de fechas, que provee los datos necesarios para la evaluación de umbrales y la generación de alertas.
- Elaboración de la documentación y técnica del sistemas además del contrato de la API para los módulos que la consumen

Entregables: Batería de pruebas automatizadas ejecutable y aprobada como el entorno de pruebas aislado y su documentación técnica del backend

4.4.10.7 Estado del Sprint Backlog 4

Tabla 31 Estado del Product Backlog al cierre del Sprint 4

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Infraestructura base del backend (servidor, contenedores Docker)	HU03	Alta	Hecho
2	Modelo de datos y migraciones (devices, readings, audit_log)	HU03	Alta	Hecho
3	API REST de consulta de lecturas y dispositivos	HU04	Alta	Hecho
4	Recepción de telemetría vía MQTT (worker de ingesta)	HU03	Alta	Hecho
5	Validación de datos y deduplicación de lecturas	HU03	Alta	Hecho
6	Auditoría de lotes con huella de integridad (SHA-256)	HU03	Media	Hecho
7	Cifrado de extremo a extremo (AES-256-GCM)	HU03	Alta	Hecho
8	Detección de manipulación de mensajes	HU03	Alta	Hecho
9	Autenticación MQTT y control de acceso (ACL)	HU03	Alta	Hecho
10	Transporte cifrado (TLS 1.3)	HU03	Alta	Hecho
11	Soporte de umbrales para alerta (datos para HU05)	HU05	Media	Hecho
12	Pruebas automatizadas y documentación técnica	HU03 y HU04	Media	Hecho
13	Cifrado AES-256-GCM en el firmware de la estación ESP32	HU03	Alta	Por hacer
14	Endpoint de ingestión HTTP para el gateway remoto	HU03	Alta	Por hacer
15	Autenticación de la API mediante API Key	HU03	Alta	Por hacer
16	Exposición del servidor con dominio propio y túnel seguro	HU04	Alta	Por hacer
17	Firmware del gateway: recepción LoRa y reenvío HTTPS	HU03	Alta	Por hacer

Nro.	Tarea	Historia de usuario	Prioridad	Estado
18	Persistencia del número de secuencia en memoria no volátil	HU03	Media	Por hacer
19	Integración y validación del sistema con datos reales	HU03, HU04	Alta	Por hacer

4.4.11 Sprint 5: Integración extremo a extremo y despliegue

4.4.11.1 Planificación del Sprint 5

Tabla 32 Integración con el hardware, seguridad de acceso y despliegue en producción

SPRINT 5: Integración con el hardware, seguridad de acceso y despliegue en producción			
Duración:	05/07/2026 – 09/07/2026 (5 días)	Prioridad:	Media
Objetivo:	Integrar el hardware real con el backend y exponer el sistema en Internet, logrando el flujo completo como automático desde el sensor físico hasta la aplicación en el móvil con el cifrado de extremo a extremo y su control de acceso		
Historias de usuario:	HU03: Sincronización y almacenamiento centralizado HU04: Visualización de históricos y monitoreo		
Tareas de desarrollo:	• Autenticación de las API mediante API Key • Endpoint de ingestión HTTP para el gateway remoto, con lógica de procesamiento compartida • Exposición del servidor con dominio propio y túnel seguro permanente • Cifrado AES-256-GCM en el firmware de la estación ESP32 • Firmware del gateway: recepción LoRa, WiFi configurable y reenvío HTTPS. • Persistencia del número de secuencia en memoria no volátil (NVS) • Registro de la hora de recepción para el control de disponibilidad del sensor		
Plan de entrega:	• Autenticación por API Key. Verificación de la llave de acceso con comparación en tiempo constante. Tiempo de desarrollo: (4)h fechas: (05/07/2026) • Dominio y túnel permanente. Registro del dominio y configuración del túnel seguro como servicio del sistema Tiempo de desarrollo: (5)h – Fechas: (05/07/2026 – 06/07/2026) • Endpoint de ingestión HTTP. Recepción de sobres cifrados del gateway remoto; extracción de la lógica compartida (process_envelope) Tiempo de desarrollo: (8) h — Fechas (07/07/2026) • Cifrado en la estación ESP32. Implementación de AES-256-GCM con mbedTLS en el microcontrolador Tiempo de desarrollo: (10) h — Fechas (07/07/2026 – 08/07/2026) • Firmware del gateway. Recepción LoRa, Wifi configurable sin reprogramación y reenvío HTTPS con API Key Tiempo de desarrollo: (8) h — Fechas (08/07/2026) • Persistencia del contador y hora de recepción. Almacenamiento del número de secuencia en NVS y registro de la hora de llegada		

SPRINT 5: Integración con el hardware, seguridad de acceso y despliegue en producción	
	Tiempo de desarrollo: (5) h — Fechas (08/07/2026 – 09/07/2026) Almacenamiento del número de secuencia en NVS y registro de la hora de llegada. Verificación del flujo completo con lecturas del sensor físico Tiempo de desarrollo: (4) h — Fechas (09/07/2026)

4.4.11.2 Sprint Backlog Sprint 5S

Tabla 33 Sprint Backlog del Sprint 5

Historia de usuario	Tarea	Duración estimada
HU03	Batería de pruebas automatizadas	8 horas
HU3	Endpoint de ingestión HTTP para el gateway remoto	4 horas
HU03	Autenticación de la API mediante API Key	4 horas
HU04	Exposición del servidor con dominio propio y túnel seguro	5 horas
HU03	Firmware del gateway: recepción LoRa y reenvió HTTPS	8 horas
HU03	Persistencia del número de secuencia en memoria no volátil	5 horas
HU03, HU04	Integración y validación del sistema con datos reales	4 horas
	Total	34 horas

4.4.11.3 Desarrollo del Sprint 5

4.4.11.3.1 Funcionalidad de integración con el hardware, seguridad de acceso y despliegue en producción

Descripción: En esta funcionalidad se integró el hardware real con el backend y se expuso el sistema a internet, logrando el flujo completo y automático desde el sensor físico hasta la aplicación móvil. Se implementó el cifrado AES-256-GCM dentro del microcontrolador de la estación de modo que el dato viaja ilegible desde su origen esto se incorporó de un Endpoint de ingestión HTTP para el gateway remoto además que se protegió la API mediante una llave de acceso y se expuso el servidor con un dominio propio y un túnel seguro permanente

Ilustración 41 Servidor accesible desde Internet mediante dominio propio

```

david@dserver:~/Back-App-Monitoreo$ curl -s -w "%{http_code}\n" https://api.darigs.com/health
{"status":"ok","service":"Back-App-Monitoreo"}
HTTP/200

```

Aquí el sistema se expone públicamente a través de un dominio con certificado HTTPS valido, sin necesidad de abrir puertos ni revelar la dirección IP del servidor esto permite que el gateway del vivero envíe datos desde cualquier red.

Ilustración 42 Control de acceso a la API mediante llave de aplicación sin llave de acceso

```

david@server1:~/Back-App-Monitoreo$ echo "--- SIN LLAVE ---"
curl -s -w "%{http_code}" "https://api.dariga.com/api/v1/telemetry/readings?batchSize=1"
echo "--- CON LLAVE ---"
--- SIN LLAVE ---
{"detail":"Credenciales inválidas"}
HTTP/1.1 401
--- CON LLAVE ---
david@server1:~/Back-App-Monitoreo$ echo "--- CON LLAVE ---"
curl -s -w "%{http_code}" -H "X-API-Key: d964aeb0894b7a95907d51546adeeb9c3c646951b4b0345efed7a96c0db43df" "https://api.dariga.com/api/v1/telemetry/readings?batchSize=1"
--- CON LLAVE ---
{"items":[{"version":1,"deviceId":"esp32-01","sensorName":"Modo esp32-01","sequence":140,"temperature":24.7,"humidity":85.5,"zssi":70,"sensorTimestamp":null,"receivedAtGateway":"2024-07-13T08:16:18.004837Z","syncStatus":"synced","transport":"lora_secure"}],"count":1}
HTTP/1.1 200

```

La API rechaza la petición con el código 401 que es de credenciales inválidas, pero con la llave correcta entrega los datos junto con el cifrado del contenido y el canal TLS, esto confirma tres capas de protección independientes que es una defensa en profundidad

Ilustración 43 Recepción y envío del dato cifrado por parte del gateway

```

PS C:\Users\Usuario\Desktop\Sincronizacion de datos\PlatformIO\Projects\Modo_Meteorologico_Lastenia> C:\Users\Usuario\
.platformio\penv\Scripts\platformio.exe device monitor --port COM7 --baud 115200
0VfzQMK7ta5j6GjadizPwCz87Yd3HMT0U1F+Tnl6A=="}
[HTTP] Código: 200
[HTTP] Respuesta: {"status":"ok","sequence":125,"inserted":1}
>>> ENVIO AL BACKEND: EXITO <<<

>>> DATO RECIBIDO DE LA ESTACION <<<
+RCV=1,106,GK6QJ5T-unSX7UO_D6HlefazttYohsAzAluPbIyil9CEUuotGONmYBENGW4J9TuiIj-n3bVSJzZwQDmISnI6DrbFndlUmK8I_irvWjw,-63,10
[RCV] Payload URL-safe (recibido): GK6QJ5T-unSX7UO_D6HlefazttYohsAzAluPbIyil9CEUuotGONmYBENGW4J9TuiIj-n3bVSJzZwQDmISnI6DrbFndlUmK8I_irvWjw
[RCV] RSSI=-63 SNR=10
[RCV] Payload base64 estandar (convertido): GK6QJ5T-unSX7UO_D6HlefazttYohsAzAluPbIyil9CEUuotGONmYBENGW4J9TuiIj+n3bVSJzZwQDmISnI6DrbFndlUmK8I_irvWjw=
[RCV] Sobre reconstruido: {"v":1,"deviceId":"esp32-01","alg":"AES-256-GCM","payload":"GK6QJ5T-unSX7UO_D6HlefazttYohsAzAluPbIyil9CEUuotGONmYBENGW4J9TuiIj+n3bVSJzZwQDmISnI6DrbFndlUmK8I_irvWjw=="}
[HTTP] Código: 200
[HTTP] Respuesta: {"status":"ok","sequence":126,"inserted":1}
>>> ENVIO AL BACKEND: EXITO <<<

```

Este gateway recibe el sobre cifrado por radiofrecuencia LoRa y lo reenvía al servidor por HTTPS se observa que el contenido (payload) es ilegible el gateway actúa como simple retransmisor sin la capacidad de interpretar la información que transporta esta captura constituye la evidencia del cifrado de extremo a extremo y el dato viaja protegido desde el sensor hasta el servidor

Ilustración 44 dato descifrado y almacenado

```
David@server:~/Back-App-Monitoreo$ curl -s -H "X-API-Key: d964aab0094b7a95907d31564adeeb9c0c46951b48b0345efed7a9cc0db43df" "https://api.dariga.com/api/v1/telemetry/readings?batchSize=3" | python3 -m json.tool
{
  "items": [
    {
      "version": 1,
      "deviceId": "esp32-01",
      "sensorName": "Nodo esp32-01",
      "sequence": 150,
      "temperature": 24.7,
      "humidity": 85.4,
      "rxssi": -70,
      "sensorTimestamp": null,
      "receivedAtGateway": "2024-07-13T08:22:19.878545Z",
      "syncStatus": "syncd",
      "transport": "lorawan_secure"
    },
    {
      "version": 1,
      "deviceId": "esp32-01",
      "sensorName": "Nodo esp32-01",
      "sequence": 149,
      "temperature": 24.7,
      "humidity": 85.5,
      "rxssi": -70,
      "sensorTimestamp": null,
      "receivedAtGateway": "2024-07-13T08:21:49.503677Z",
      "syncStatus": "syncd",
      "transport": "lorawan_secure"
    }
  ]
}
```

El sobre ilegible de la ilustración anterior aparece aquí descifrado se puede ver temperatura humedad y número de secuencia. Solo el servidor que posee la llave puede interpretarlo el contraste entre ambas capturas demuestra el funcionamiento del cifrado de extremo a extremo

```

String SecurityManager::encryptPayloadOnly(const String& plaintext, const String&
deviceId) {
    const size_t plainLen = plaintext.length();

    // 1. Nonce aleatorio de 12 bytes, generado por hardware
    unsigned char nonce[NONCE_LEN];
    esp_fill_random(nonce, NONCE_LEN);

    unsigned char ciphertext[MAX_PLAINTEXT];
    unsigned char tag[TAG_LEN];

    // 2. Se carga la llave AES de 256 bits
    mbedtls_gcm_context ctx;
    mbedtls_gcm_init(&ctx);
    mbedtls_gcm_setkey(&ctx, MBEDTLS_CIPHER_ID_AES, AES_KEY, 256);

    // 3. El deviceId se usa como dato autenticado adicional (AAD)
    const unsigned char* aad = (const unsigned char*)deviceId.c_str();
    const size_t aadLen = deviceId.length();

    // 4. Cifrado autenticado: produce el texto cifrado y el tag
    int ret = mbedtls_gcm_crypt_and_tag(
        &ctx, MBEDTLS_GCM_ENCRYPT, plainLen,
        nonce, NONCE_LEN,
        aad, aadLen,
        (const unsigned char*)plaintext.c_str(),
        ciphertext, TAG_LEN, tag);
    mbedtls_gcm_free(&ctx);

    // 5. Se ensambla el bloque: nonce(12) || ciphertext || tag(16)
    unsigned char blob[NONCE_LEN + MAX_PLAINTEXT + TAG_LEN];
    const size_t blobLen = NONCE_LEN + plainLen + TAG_LEN;
    memcpy(blob, nonce, NONCE_LEN);
    memcpy(blob + NONCE_LEN, ciphertext, plainLen);
    memcpy(blob + NONCE_LEN + plainLen, tag, TAG_LEN);

    // 6. Codificación en Base64 para su transmisión por LoRa
    const unsigned int b64Len = encode_base64_length(blobLen);
    unsigned char b64[b64Len + 1];
    encode_base64(blob, blobLen, b64);
    b64[b64Len] = '\0';

    return String((char*)b64);
}

```

Esta función implementa el cifrado autenticado dentro del propio microcontrolador de la estación si emplea la biblioteca mbedTLS que es el elemento que convierte el sistema en un esquema de cifrado de extremo a extremo en sentido estricto el dato nunca existe en claro fuera del dispositivo que lo midió y del servidor que lo recibe

El proceso consta de seis pasos el primero que se genera nonce aleatorio de 12 bytes mediante `esp_fill_random` que utiliza el generador de números aleatorios por hardware del ESP32 este valor debe ser distinto en cada mensaje ya que reutilizarlo comprometería la seguridad del mismo cifrado. Lo siguiente se carga la llave simétrica de 256 bits en el contexto GCM el identificador del dispositivo se pasa como dato autenticado adicional (ADD) no se cifra, pero queda protegido contra alteración de modo que un mensaje no puede atribuirse a otra estación.

La operación `mbedtls_gcm_crypt_and_tag` realiza el cifrado y el cálculo del tag de autenticación en un solo paso. El tag de 16 bytes es una huella criptográfica del contenido cualquier modificación posterior, aunque sea un solo bit va a hacer invalidar su verificación en el servidor .

Finalmente, los tres elementos se ensamblan en un único bloque nonce (12 bytes || tag (16 bytes)) y se codifican en Base64 para su transmisión por el enlace LoRa este es exactamente el formato que el servidor espera y sabe descomponer

4.4.11.4 Sprint Review Sprint 5

4.4.11.4.1 Descripción de la funcionalidad entregada

Al cierre del quinto Sprint se entregó el sistema completo en funcionamiento: el dato viaja cifrado desde el sensor físico, por radiofrecuencia LoRa, a través del gateway, por Internet mediante dominio propio, hasta el servidor que lo descifra y almacena quedando disponible en la aplicación móvil y en el panel de monitoreo sin intervención manual

Ilustración 45 Recepción y reenvío del dato cifrado por parte del gateway

```
PS C:\Users\Usuario\Desktop\Sincronizacion de datos\PlatformIO\Projects> c:\Users\Usuario\.platformio\penv\Scripts\platformio.exe device monitor --port COM7 --baud 115200
>>> DATO RECIBIDO DE LA ESTACION <<<
+RCV-1,106,KqjsQ5tWllzeHhuuG0nu#BFTagHzMcq4ZRQ1lyxa_jCvd-eFmVRyQ11u0xeUfXa15_bzrlccw0kjhpM81gX0R8CgU703TVoxFfzzgeQ,-87,10
[RCV] Payload URL-safe (recibido): KqjsQ5tWllzeHhuuG0nu#BFTagHzMcq4ZRQ1lyxa_jCvd-eFmVRyQ11u0xeUfXa15_bzrlccw0kjhpM81gX0R8CgU703TVoxFfzzgeQ
[RCV] RSSI=-87 SNR=10
[RCV] Payload base64 estandar (convertido): KqjsQ5tWllzeHhuuG0nu#BFTagHzMcq4ZRQ1lyxa_jCvd-eFmVRyQ11u0xeUfXa15_bzrlccw0kjhpM81gX0R8CgU703TVoxFfzzgeQ=
[RCV] Sobre reconstruido: {"v":1,"deviceId":"esp32-01","alg":"AES-256-GCM","payload":"KqjsQ5tWllzeHhuuG0nu#BFTagHzMcq4ZRQ1lyxa_jCvd-eFmVRyQ11u0xeUfXa15_bzrlccw0kjhpM81gX0R8CgU703TVoxFfzzgeQ="}
[HTTP] Código: 200
[HTTP] Respuesta: {"status":"ok","sequence":260,"inserted":1}
>>> ENVIO AL BACKEND: EXITO <<<
```

Ilustración 46 El mismo dato ya descifrado y almacenado

```
David@server1:~/Back-App-Monitoreo$ docker compose exec db psql -U monitoreo -d monitoreo -c "SELECT device_id, sequence, temperature, humidity, received_at_gateway, re
```

device_id	sequence	temperature	humidity	received_at_gateway	received_at_server
1	260	28.2	79.1	2026-07-22 04:25:20.77541+00	2026-07-22 04:25:20.785478+00

(1 row)

- **Retroalimentación establecida**

La integración con el hardware puso de manifiesto cuatro incidencias que no se habían anticipado en el diseño:

- a) El sobre cifrado excedía la capacidad del enlace de radio. El sobre completo ocupaba 194 bytes, tamaño que el módulo LoRa no lograba transmitir de forma estable
- b) El contador de secuencia se reiniciaba al reprogramar la estación. Al volver a empezar desde uno las lecturas nuevas colisionaban con las ya almacenadas y eran rechazadas por duplicidad, impidiendo el ingreso de datos
- c) La aplicación móvil no podía determinar si un sensor estaba activo, al carecer las lecturas de una marca temporal de recepción. Adicionalmente la aplicación interpretaba como hora local las marcas registradas en UTC mostrando un retraso permanente de cinco horas
- d) Algunos mensajes llegaban con bytes alterados por el enlace de radio y eran rechazados por la verificación criptográfica

- **Mejoras realizadas**

- a) **Optimización del mensaje transmitido:** Se redujo la carga enviada por radiofrecuencia de 194 a 103 bytes una reducción del 47% transmitiendo únicamente el contenido cifrado y delegando en el gateway la reconstrucción del sobre completo. Además se eliminó el identificador del dispositivo del texto en claro, conservándolo como dato autenticado (AAD) y se adoptó una codificación en Base64 apta para la transmisión. La seguridad criptográfica se mantuvo intacta
- b) **Persistencia del contador en memoria no volátil:** El número de secuencia se almacena en la memoria NVS del microcontrolador de este modo que sobrevive a reinicios y reprogramaciones. La deduplicación vuelve así a ser confiable el contador continúa desde el último valor emitido en lugar de reiniciarse en uno, las lecturas nuevas dejan de colisionar con las ya almacenadas
- c) **Registro de la hora de recepción y corrección de la zona horaria:** El servidor registra al instante en que recibe cada lectura, lo que permite a la aplicación móvil determinar si una estación sigue viva o dejó de reportar adicionalmente, las marcas temporales se convierten de UTC a la hora local antes de mostrarse, eliminando el retraso permanente de cinco horas que presentaba la interfaz
- d) **Robustecimientos del enlace de radio:** Se ajustaron los parámetros de transmisión LoRa hacia un factor de dispersión más robusto (SF12) reduciendo así la cantidad de mensajes alterados por el canal. Los pocos mensajes que aun llegan corruptos son rechazados por la verificación criptográfica del cifrado autenticado, de tal manera que ningún dato manipulado o dañado ingresa a la base de datos esta decisión constituye además de una evidencia directa de que la integridad de extremo a extremo funciona

Con estas correcciones el sistema quedo operando de extremo a extremo con datos reales de forma continua, cumpliendo el objetivo del sprint

4.4.11.5 Pruebas del Sprint 5

4.4.11.5.1 Caja Negra

Tabla 34 Ingestión segura por telemetría por HTTP

Nombre del campo	Tipo de campo	Valor permitido	Observación
Llave de aplicación (X-API-Key)	Cabecera HTTP (texto)	Llave válida registrada en el servidor	Sin llave o con llave inválida, la petición se rechaza con código 401.
Sobre cifrado (payload)	Cuerpo JSON, carga en Base64 (AES-256-GCM)	Sobre íntegro cifrado con la llave del dispositivo	Un sobre manipulado se rechaza con código 400 (decrypt_failed) y se registra en auditoría; no se almacena.
deviceId	Texto	Identificador de estación registrado (p. ej., esp32-01)	Se emplea como dato autenticado (AAD); vincula el mensaje a su estación de origen.
sequence	Numérico	Entero ≥ 0 , nuevo por dispositivo	Una secuencia repetida se rechaza como duplicada (código 409); no se inserta dos veces.
temperature	Numérico	Entre -40 y 85 °C	Fuera de rango se rechaza como invalid_payload (código 400).
humidity	Numérico	Entre 0 y 100 %	Fuera de rango se rechaza como invalid_payload (código 400).

4.4.11.5.2 Caja Blanca

Tabla 35 Procesamiento de la ingestión y control de acceso

Método	Acción esperada	Acción obtenida	Observación
Ingesta HTTP de lote cifrado válido	Almacenar la lectura y responder 200	200 {"status":"ok","inserted":1}; lectura almacenada	Ingestión por HTTP verificada.
Petición sin llave de aplicación	Rechazar con 401	401 (credenciales inválidas)	Autenticación por API Key verificada.
Reenvío del mismo lote (misma secuencia)	No duplicar y responder 409	409 {"status":"duplicate","inserted":0}	Deduplicación de extremo a extremo verificada.
Lote manipulado (un byte alterado)	Rechazar y auditar decrypt_failed	400 {"status":"decrypt_failed"}; auditado (source = http)	Detección de manipulación verificada.

Método	Acción esperada	Acción obtenida	Observación
Sobre con esquema inválido	Rechazar como invalid_payload	400 {"status":"invalid_payload"}	Validación de entrada verificada.

4.4.11.6 Incremento y Entregables del Sprint 5

Al finalizar el Sprint 5 se completaron satisfactoriamente los siguientes procesos:

- Cifrado AES-256-GCM ejecutado en el microcontrolador no volátil (NVS) que garantizaba la deduplicación
- Persistencia del controlador de secuencia en memoria no volátil (NVS) que garantiza la deduplicación tras reinicios
- Firmware del gateway que es para la recepción LoRa reconstrucción del sobre cifrado y reenvió al servidor por Wifi/HTTP
- Endpoint HTTP de ingestión (/ingest) protegido con autenticación por llave de aplicación (API Key)
- Unificación de la lógica de ingestión de las vías MQTT y HTTP en un único módulo compartido
- Adaptación de la lectura del ESP32 al formato de lote y estampado de la hora de recepción
- Publicación del servicio en Internet mediante dominio propio con túnel permanente (HTTPS)
- Integración de extremo a extremo verificada con hardware real y datos reales

Entregables: Firmware de estación y gateway operativos; backend con ingestión dual (MQTT y HTTP) y autenticación sistema publicado en Internet y integración completa funcionando

4.4.11.7 Estado actual del Product Backlog 5

Tabla 36 Estado del Product Backlog al cierre del Sprint 5

Nro.	Tarea	Historia de usuario	Prioridad	Estado
1	Infraestructura base del backend (servidor, contenedores Docker)	HU03	Alta	Hecho

Nro.	Tarea	Historia de usuario	Prioridad	Estado
2	Modelo de datos y migraciones (devices, readings, audit_log)	HU03	Alta	Hecho
3	API REST de consulta de lecturas y dispositivos	HU04	Alta	Hecho
4	Recepción de telemetría vía MQTT (worker de ingesta)	HU03	Alta	Hecho
5	Validación de datos y deduplicación de lecturas	HU03	Alta	Hecho
6	Auditoría de lotes con huella de integridad (SHA-256)	HU03	Media	Hecho
7	Cifrado de extremo a extremo (AES-256-GCM)	HU03	Alta	Hecho
8	Detección de manipulación de mensajes	HU03	Alta	Hecho
9	Autenticación MQTT y control de acceso (ACL)	HU03	Alta	Hecho
10	Transporte cifrado (TLS 1.3)	HU03	Alta	Hecho
11	Soporte de umbrales para alerta (datos para HU05)	HU05	Media	Hecho
12	Pruebas automatizadas y documentación técnica	HU03 y HU04	Media	Hecho
13	Cifrado AES-256-GCM en el firmware de la estación ESP32	HU03	Alta	Hecho
14	Endpoint de ingestión HTTP para el gateway remoto	HU03	Alta	Hecho
15	Autenticación de la API mediante API Key	HU03	Alta	Hecho
16	Exposición del servidor con dominio propio y túnel seguro	HU04	Alta	Hecho
17	Firmware del gateway: recepción LoRa y reenvío HTTPS	HU03	Alta	Hecho
18	Persistencia del número de secuencia en memoria no volátil	HU03	Media	Hecho
19	Integración y validación del sistema con datos reales	HU03, HU04	Alta	Hecho

4.4.12 Resultado Final del Producto

4.4.12.1 Funcionalidades implementadas

Tabla 37 Funcionalidades implementadas

Funcionalidad	Estado
API REST con el contrato de la aplicación móvil	Implementado y probado
Base de datos PostgreSQL con migraciones versionadas	Implementado y probado
Ingesta de telemetría mediante MQTT	Implementado y probado
Ingesta de telemetría mediante HTTP con API Key (gateway remoto)	Implementado y probado
Lógica de procesamiento compartida entre MQTT y HTTP	Implementado y probado
Deduplicación de lecturas (tolerancia a reenvíos)	Implementado y probado
Auditoría de lotes con huella SHA-256	Implementado y probado
Cifrado de extremo a extremo (AES-256-GCM) desde el microcontrolador	Implementado y probado
Detección de manipulación de mensajes	Implementado y probado
Persistencia del contador de secuencia en memoria NVS	Implementado y probado
Autenticación MQTT y control de acceso (ACL)	Implementado
Transporte cifrado (TLS 1.3 / HTTPS)	Implementado y probado
Publicación en Internet mediante dominio propio y túnel permanente	Implementado y probado
Firmware de estación y gateway	Implementado y probado
Pruebas automatizadas (52)	Aprobadas
Documentación técnica	Elaborada
Funcionalidad	Estado
API REST con el contrato de la aplicación móvil	Implementado y probado

4.4.12.2 Evidencia final del sistema

Ilustración 47 Dato ingresado y descartado por intento simulado de inserción de datos ajeno a la estación

```
David@server:~/Back-App-Monitoreo$ docker compose run --rm --entrypoint python worker ./scripts/publish_encrypted_test.py --sequences 600 --tamper
(x) 2/21 2/22
Container monitoreo-mqtt Running
Container monitoreo-db Running
Container monitoreo-db Waiting
Container monitoreo-db Healthy
Container back-app-monitoreo-worker-run-7b36d0bb3d7d Creating
Container back-app-monitoreo-worker-run-7b36d0bb3d7d Created
MODO TAMPER: se alteró el ciphertext - se espera decrypt failed en audit_log
Conectando a mosquitto:1883 como esp32-01 (client_id=esp32-sim-01, TLS=False)...
Publicando lote cifrado en vivero/el-carmen/esp32-01/telemetria (QoS 1), secuencias=(600,1,...
Publicado OK: deviceId=esp32-01 lecturas=1
David@server:~/Back-App-Monitoreo$ docker compose logs --tail=10 worker
monitoreo-worker | 2026-06-19 03:47:29,130 [INFO] _main_ | Iniciando worker de ingesta MQTT (broker=mosquitto:1883, cifrado_obligatorio=True, backoff_inicial=3s)
monitoreo-worker | 2026-06-19 03:47:29,131 [INFO] _main_ | Intentando conectar a Mosquitto mosquitto:1883...
monitoreo-worker | 2026-06-19 03:47:29,177 [INFO] _main_ | Conectado a Mosquitto (mosquitto:1883) como client_id=monitoreo-worker
monitoreo-worker | 2026-06-19 03:47:29,177 [INFO] _main_ | Suscrito a vivero/+/*/telemetria (QoS 1, mid=1)
monitoreo-worker | 2026-06-19 03:47:55,182 [INFO] app.ingest: Lote recibido: tópico=vivero/el-carmen/esp32-01/telemetria zona=el-carmen dispositivo_tópico=esp32-01 byt
esp=549
monitoreo-worker | 2026-06-19 03:47:55,398 [INFO] app.ingest: Ingesta completada: dispositivo=esp32-01 insertados=2 duplicados=0 total=2 resultado_auditoria=ok
monitoreo-worker | 2026-06-19 03:47:55,398 [INFO] app.ingest: Auditoria registrada: dispositivo=esp32-01 resultado=ok detalle=insertados=2 duplicados=0 total=2 sha256=
ea42a2a08a7f0e4...
monitoreo-worker | 2026-06-19 03:48:45,782 [INFO] app.ingest: Lote recibido: tópico=vivero/el-carmen/esp32-01/telemetria zona=el-carmen dispositivo_tópico=esp32-01 byt
esp=361
monitoreo-worker | 2026-06-19 03:48:45,783 [WARNING] app.ingest: Descifrado fallido: dispositivo=esp32-01 motivo=autenticación fallida
monitoreo-worker | 2026-06-19 03:48:45,822 [INFO] app.ingest: Auditoria registrada: dispositivo=esp32-01 resultado=decrypt failed detalle=autenticación fallida
David@server:~/Back-App-Monitoreo$ docker compose exec db psql -U monitoreo -d monitoreo -c "SELECT device_external_id, result, detail FROM audit_log ORDER BY id DESC
LIMIT 1;"
 device_external_id | result | detail
-----
 esp32-01 | decrypt_failed | autenticación fallida
(1 row)
Activar Windows
Ve a Configuración para activar Windows.
```

Dato ingresado y descartado por intento simulado de inserción de datos ajeno a la estación

Ilustración 48 Dato ingresado y aceptado publicado por la estación y el gateway

```
David@server:~/Back-App-Monitoreo$ docker compose run --rm --entrypoint python worker ./scripts/publish_encrypted_test.py --sequences 600,601
(x) 2/21 2/22
Container monitoreo-db Running
Container monitoreo-mqtt Running
Container monitoreo-db Waiting
Container monitoreo-db Healthy
Container back-app-monitoreo-worker-run-1827f0c74408 Creating
Container back-app-monitoreo-worker-run-1827f0c74408 Created
Conectando a mosquitto:1883 como esp32-01 (client_id=esp32-sim-01, TLS=False)...
Publicando lote cifrado en vivero/el-carmen/esp32-01/telemetria (QoS 1), secuencias=(600, 601)...
Publicado OK: deviceId=esp32-01 lecturas=2
David@server:~/Back-App-Monitoreo$ docker compose logs --tail=15 worker
monitoreo-worker | 2026-06-19 03:47:29,130 [INFO] _main_ | Iniciando worker de ingesta MQTT (broker=mosquitto:1883, cifrado_obligatorio=True, backoff_inicial=3s)
monitoreo-worker | 2026-06-19 03:47:29,131 [INFO] _main_ | Intentando conectar a Mosquitto mosquitto:1883...
monitoreo-worker | 2026-06-19 03:47:29,177 [INFO] _main_ | Conectado a Mosquitto (mosquitto:1883) como client_id=monitoreo-worker
monitoreo-worker | 2026-06-19 03:47:29,177 [INFO] _main_ | Suscrito a vivero/+/*/telemetria (QoS 1, mid=1)
monitoreo-worker | 2026-06-19 03:47:55,182 [INFO] app.ingest: Lote recibido: tópico=vivero/el-carmen/esp32-01/telemetria zona=el-carmen dispositivo_tópico=esp32-01 byt
esp=549
monitoreo-worker | 2026-06-19 03:47:55,398 [INFO] app.ingest: Ingesta completada: dispositivo=esp32-01 insertados=2 duplicados=0 total=2 resultado_auditoria=ok
monitoreo-worker | 2026-06-19 03:47:55,398 [INFO] app.ingest: Auditoria registrada: dispositivo=esp32-01 resultado=ok detalle=insertados=2 duplicados=0 total=2 sha256=
ea42a2a08a7f0e4...
David@server:~/Back-App-Monitoreo$ curl "http://localhost:8000/api/v1/telemetry/readings?batchSize=10"
{"items":[{"version":1,"deviceId":"esp32-01","sensorName":"Modo esp32-01","sequence":601,"temperature":24.5,"humidity":76.0,"rssi":-71,"sensorTimestamp":17181040876,"re
ceivedAtGateway":"2026-06-19T03:47:55.132953Z","syncStatus":"synced","transport":"lora_secure"},{"version":1,"deviceId":"esp32-01","sensorName":"Modo esp32-01","sequence
":600,"temperature":24.0,"humidity":75.0,"rssi":-70,"sensorTimestamp":17181040878,"receivedAtGateway":"2026-06-19T03:47:55.132953Z","syncStatus":"synced","transport":"lo
ra_secure"},{"version":1,"deviceId":"esp32-01","sensorName":"Modo esp32-01","sequence":601,"temperature":24.5,"humidity":76.0,"rssi":-71,"sensorTimestamp":17181040876,"r
eceivedAtGateway":"2026-06-19T04:57:50.674940Z","syncStatus":"synced","transport":"lora_secure"},{"version":1,"deviceId":"esp32-01","sensorName":"Modo esp32-01","sequen
ce":1500,"temperature":24.0,"humidity":75.0,"rssi":-70,"sensorTimestamp":17181040878,"receivedAtGateway":"2026-06-19T04:57:50.674940Z","syncStatus":"synced","transport":"
lora_secure"},{"version":1,"deviceId":"esp32-01","sensorName":"Modo esp32-01","sequence":612,"temperature":24.6,"humidity":78.2,"rssi":-71,"sensorTimestamp":1717000000,"
receivedAtGateway":"2026-06-10T14:32:00Z","syncStatus":"synced","transport":"lora_secure"}],"count":5}David@server:~/Back-App-Monitoreo$
```

Dato ingresado y aceptado publicado por la estación y el Gateway

4.4.12.3 Conclusiones del desarrollo

El desarrollo del backend mediante la metodología Scrum permitió construir, de forma incremental y verificable el componente de sincronización segura del sistema de monitoreo ambiental del vivero de cacao. Cada Sprint entrego un incremento funcional y probado en el servidor real la infraestructura y la API de consulta la ingesta MQTT con deduplicacion y

auditoria, la seguridad de extremo a extremo, la validación automatizada y finalmente la integración con el hardware y el despliegue seguro en Internet

El sistema resultante garantiza la confidencialidad e integridad de la información ambiental desde su origen el dato nace cifrado en el microcontrolador de la estación y solo el servidor puede descifrarlo de modo que ningún intermediario accede a su contenido. La incorporación de una segunda vía de ingestión por HTTP autenticado permitió sincronizar los datos del gateway remoto sin exponer el bróker MQTT a internet

La aplicación de la metodología SCRUM facilitó la organización del trabajo mediante sus interacciones cortas así permitiendo desarrollar y validar las funcionalidades de manera incremental. Este enfoque favoreció la detección temprana de incidencias reales como el error de identidad de certificado TLS o el desbordamiento del enlace de radio y aseguro que cada funcionalidad cumpliera la definición de hecho antes de avanzar

CAPÍTULO V

5. EVALUACIÓN DE RESULTADOS

5.1 Introducción

La validación de un sistema informático se debe someter a distintos tipos de pruebas, lo cual hace que cada una de estas este orientada a verificar un aspecto diferente de su comportamiento. Las pruebas de caja negra evalúan el sistema desde el exterior, sin conocer su implementación interna además que se proporcionan entradas determinadas y se comprueba que las salidas correspondan a lo esperado aplicando así técnicas como la partición de equivalencia y el análisis de valores limite. Las pruebas de caja blanca estas examinan las rutas internas de ejecución del código, verificando que cada camino lógico incluidos los de error se comporten según lo previsto. A estas se suman las pruebas unitarias, que validan de forma aislada cada componente y las pruebas de integración que comprueban que los componentes funcionan correctamente al operar en conjunto.

En este proyecto la validación del componente de sincronización se realizó combinando estos enfoques. Ya que las pruebas de caja negra se aplicación sobre los puntos de entrada del sistema, evaluando como responde ante lotes de telemetrías con distintos valores ya sea temperatura humedad o identidad del dispositivo y el estado del cifrado sin considerar la implementación del procesamiento el descifrado y la verificación de integridad, la validación de los rangos, la deduplicacion de lecturas y el registro de la auditoria comprobando que cada resultado posible sea (ok, duplicate, invalid_payload o decrypt_failed) se produjera en la circunstancia correspondiente. Ambos tipos se documentaron por Sprint, de modo que cada incremento entregado fue verificado antes de avanzar al siguiente.

Además de eso se construyó una batería de pruebas automatizadas ejecutada con la herramienta pytest sobre una base de datos aislada, organizada en tres grupos según el módulo evaluado que es el cifrado, worker de ingesta y API REST. Su carácter automatizado permite repetir la verificación completa ante cualquier cambio del código, garantizando que las funcionalidades previamente validadas continúen operando correctamente. Finalmente se realizaron pruebas de integración de extremo a extremo con el hardware real siguiendo el recorrido completo del dato desde su captura hasta su disponibilidad para consultarla.

Los resultados de estas pruebas se apoyan en tres fuentes objetivas del propio sistema como la base de datos, donde se verifica el almacenamiento efectivo de las lecturas y el riesgo de auditoría de cada lote procesado y su salida de la batería automatizada. De este modo la evaluación que se presenta en este capítulo no depende de apreciaciones subjetivas si no de evidencia verificable y reproducible.

5.2 Presentación y monitoreo de resultados

5.2.1 Planificación de la evaluación

Tabla 38 Planificación de la evaluación del componente de sincronización

Proceso evaluar	Método de validación	Resultado esperado
Recepción y almacenamiento de telemetría (ingesta de datos en el servidor)	Se ejecutaron diez pruebas enviando lotes cifrados con secuencias nuevas, cinco a través de MQTT y cinco por HTTP. En cada una se comprobó si la lectura quedaba guardada en la base de datos, qué resultado registraba la bitácora de auditoría y cuánto tardaba el dato desde que llegaba hasta que se almacenaba.	Que las diez lecturas se guarden sin que se pierda ninguna. Cada lote debe dejar una fila en la tabla de lecturas y un registro de auditoría con resultado ok, y el almacenamiento debe completarse en menos de un segundo. Las dos vías deben arrojar el mismo resultado.
Deduplicación de lecturas (integridad del histórico ante reenvíos)	Se realizaron diez pruebas de reenvío: en cada una se mandó un lote y, sin cambiar su número de secuencia, se volvió a enviar el mismo. Se contó cuántas filas quedaban guardadas por cada par estación-secuencia y qué resultado registraba la auditoría para el reenvío.	Que se conserve una sola copia por cada par estación-secuencia, sin ningún duplicado en la base. Cada reenvío debe quedar registrado en la auditoría como duplicate y, por la vía HTTP, responder con código 409.
Detección de manipulación de datos (verificación de integridad del cifrado)	Se realizarán 10 pruebas enviando lotes cifrados manipulados, alterando un byte del contenido antes de su transmisión. En cada prueba se medirá si el lote fue rechazado, si la lectura llegó a almacenarse y el resultado registrado en la auditoría.	Los 10 lotes manipulados deben ser rechazados (100 % de detección). Ninguna lectura alterada debe almacenarse en la base de datos (0 % de admisión). Cada intento debe generar un registro de auditoría con resultado decrypt_failed y su huella SHA-256. Por tratarse de un control de seguridad, no se admite ningún fallo de detección.

5.2.2 Ejecución del monitoreo

5.2.2.1 Proceso 1 Recepción y almacenamiento de telemetría

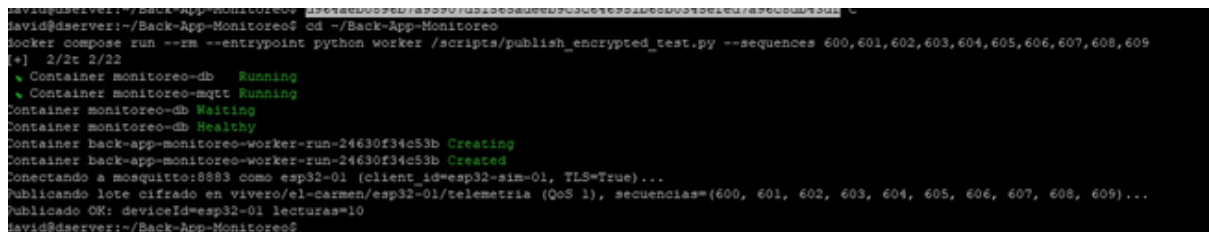
5.2.2.1.1 Descripción

Esta prueba verifica que un lote de telemetría cifrado, enviado desde una estación, sea recibido por el servidor, descifrado, validado y almacenado en la base de datos, quedando disponible para consulta. Se ejecutan diez pruebas: cinco por la vía MQTT y cinco por la vía HTTP, con el fin de comprobar que ambos caminos de entrada producen el mismo resultado.

Procedimiento: Para cada prueba se publica un lote de telemetría cifrado con un número de secuencia nuevo, empleando el simulador de estación que se conecta al broker mediante TLS y publica con calidad de servicio 1:

```
docker compose run --rm --entrypoint python worker
/scripts/publish_encrypted_test.py --sequences 600
```

Ilustración 49 Publicación de un lote cifrado de telemetría mediante MQTT sobre TLS



```
David@dsrserver:~/Back-App-Monitoreo$ docker compose run --rm --entrypoint python worker /scripts/publish_encrypted_test.py --sequences 600,601,602,603,604,605,606,607,608,609
[*] 2/2t 2/22
Container monitoreo-db Running
Container monitoreo-mqtt Running
Container monitoreo-db Waiting
Container monitoreo-db Healthy
Container back-app-monitoreo-worker-run-24630f34c53b Creating
Container back-app-monitoreo-worker-run-24630f34c53b Created
Conectando a mosquitto:8883 como esp32-01 (client_id=esp32-sim-01, TLS=True)
Publicando lote cifrado en vivero/el-carmen/esp32-01/telemetria (QoS 1), secuencias=(600, 601, 602, 603, 604, 605, 606, 607, 608, 609)...
Publicado OK: deviceId=esp32-01 lecturas=10
David@dsrserver:~/Back-App-Monitoreo$
```

A continuación, se consulta la base de datos para confirmar que las lecturas quedaron almacenadas y para medir el tiempo transcurrido entre su recepción y su almacenamiento:

```
docker compose exec db psql -U monitoreo -d monitoreo -c "SELECT sequence, transport, temperature,
humidity, ... FROM readings WHERE sequence BETWEEN 600 AND 609 ORDER BY sequence;"
```

Ilustración 50 Lecturas almacenadas en la base de datos tras la ingesta

```
david@dsrver:~/Back-App-Monitoreo$ docker compose exec db psql -U monitoreo -d monitoreo -c "SELECT sequence, transport, temperature, humidity, ROUND(EXTRACT(EPOCH FROM (received_at_server - COALESCE(received_at_gateway, received_at_server))))::numeric,3) AS latencia_seg FROM readings WHERE sequence BETWEEN 600 AND 609 ORDER BY sequence;"
sequence | transport | temperature | humidity | latencia_seg
-----
600 | lora_secure | 24 | 75 | 0.232
601 | lora_secure | 24.5 | 76 | 0.232
602 | lora_secure | 25 | 77 | 0.232
603 | lora_secure | 25.5 | 78 | 0.232
604 | lora_secure | 26 | 79 | 0.232
605 | lora_secure | 26.5 | 80 | 0.232
606 | lora_secure | 27 | 81 | 0.232
607 | lora_secure | 27.5 | 82 | 0.232
608 | lora_secure | 28 | 83 | 0.232
609 | lora_secure | 28.5 | 84 | 0.232
(10 rows)
```

Finalmente se verifica el registro de auditoría, donde cada lote procesado queda clasificado según su resultado:

```
docker compose exec db psql -U monitoreo -d monitoreo -c "SELECT device_external_id, source, result, created_at FROM audit_log ORDER BY id DESC LIMIT 10;"
```

Ilustración 51 Registro de auditoría de los lotes procesados por ambas vías de ingesta

```
david@dsrver:~/Back-App-Monitoreo$ docker compose exec db psql -U monitoreo -d monitoreo -c "SELECT device_external_id, source, result, created_at FROM audit_log ORDER BY id DESC LIMIT 10;"
device_external_id | source | result | created_at
-----
esp32-01 | http | ok | 2024-07-22 07:36:24.693395+00
esp32-01 | http | ok | 2024-07-22 07:35:47.557452+00
esp32-01 | http | ok | 2024-07-22 07:35:11.552332+00
esp32-01 | http | ok | 2024-07-22 07:34:35.07514+00
esp32-01 | http | ok | 2024-07-22 07:33:59.42237+00
esp32-01 | http | ok | 2024-07-22 07:33:22.808733+00
esp32-01 | http | ok | 2024-07-22 07:32:46.591514+00
esp32-01 | mqtt | ok | 2024-07-22 07:32:17.310144+00
esp32-01 | http | ok | 2024-07-22 07:32:10.400734+00
esp32-01 | http | ok | 2024-07-22 07:31:34.181486+00
(10 rows)
```

5.2.2.1.2 Toma de datos

Tabla 39 Resultados del Proceso 1 – Recepción y almacenamiento de telemetría

Nro.	Secuencia	Temperatura	Humedad	¿Se almacenó?	Auditoría	Latencia (s)	Observación
1	600	24,0 °C	75 %	Sí	ok	0,232	Lote recibido y almacenado correctamente
2	601	24,5 °C	76 %	Sí	ok	0,232	Almacenamiento sin errores
3	602	25,0 °C	77 %	Sí	ok	0,232	Almacenamiento sin errores
4	603	25,5 °C	78 %	Sí	ok	0,232	Almacenamiento sin errores

Nro.	Secuencia	Temperatura	Humedad	¿Se almacenó?	Auditoría	Latencia (s)	Observación
5	604	26,0 °C	79 %	Sí	ok	0,232	Almacenamiento sin errores
6	605	26,5 °C	80 %	Sí	ok	0,232	Almacenamiento sin errores
7	606	27,0 °C	81 %	Sí	ok	0,232	Almacenamiento sin errores
8	607	27,5 °C	82 %	Sí	ok	0,232	Almacenamiento sin errores
9	608	28,0 °C	83 %	Sí	ok	0,232	Almacenamiento sin errores
10	609	28,5 °C	84 %	Sí	ok	0,232	Almacenamiento sin errores
Promedio	—	26,3 °C	79,5 %	10 de 10	ok	0,232	Ninguna lectura perdida (0 % de pérdida)

5.2.2.2 Proceso 2 Deduplicación de lecturas

5.2.2.2.1 Descripción

Esta prueba verifica que el reenvío de un mismo lote no genere registros duplicados en el histórico. El nivel de calidad de servicio 1 del protocolo MQTT garantiza la entrega al menos una vez, por lo que estos reenvíos constituyen un comportamiento esperado ante reconexiones la deduplicación se resuelve mediante una restricción de unicidad sobre el par estación-secuencia definida ya en la base de datos, de modo que la protección no depende de la lógica de la aplicación

Procedimiento: En cada prueba se publica un lote y sin modificar su número de secuencia se vuelve a publicar el mismo lote:

```
docker compose exec db psql -U monitoreo -d monitoreo -c "SELECT sequence, transport, temperature, humidity, ... FROM readings WHERE sequence BETWEEN 600 AND 609 ORDER BY sequence;"
```

Ilustración 52 Envío de un lote cifrado manipulado para la prueba de detección de integridad

```
david@dsrver:~/Back-App-Monitoreo$ docker compose run --rm --entrypoint python worker /scripts/publish_encrypted_test.py --sequences 800 --tamper
[+] 2/2t 2/22
  Container monitoreo-db Running
  Container monitoreo-mqtt Running
Container monitoreo-db Waiting
Container monitoreo-db Healthy
Container back-app-monitoreo-worker-run-06deaaa6761 Creating
Container back-app-monitoreo-worker-run-06deaaa6761 Created
MODO TAMPER: se alteró el ciphertext - se espera decrypt_failed en audit_log
Conectando a mosquito:8883 como esp32-01 (client_id=esp32-sim-01, TLS=True)...
Publicando lote cifrado en vivo/el-carmen/esp32-01/telemetria (QoS 1), secuencias=(800,...)
Publicado OK: deviceId=esp32-01 lecturas=1
david@dsrver:~/Back-App-Monitoreo$ docker compose run --rm --entrypoint python worker /scripts/publish_encrypted_test.py --sequences 700
[+] 2/2t 2/22
  Container monitoreo-mqtt Running
  Container monitoreo-db Running
Container monitoreo-db Waiting
Container monitoreo-db Healthy
Container back-app-monitoreo-worker-run-3bfbfd084870 Creating
Container back-app-monitoreo-worker-run-3bfbfd084870 Created
Conectando a mosquito:8883 como esp32-01 (client_id=esp32-sim-01, TLS=True)...
Publicando lote cifrado en vivo/el-carmen/esp32-01/telemetria (QoS 1), secuencias=(700,...)
Publicado OK: deviceId=esp32-01 lecturas=1
david@dsrver:~/Back-App-Monitoreo$
```

A continuación, se comprueba que no existan registros duplicados en la base de datos

Ilustración 53 Verificación de ausencia de registros duplicados en la base de datos

```
Publicado OK: deviceId=esp32-01 lecturas=1
david@dsrver:~/Back-App-Monitoreo$ docker compose exec db psql -U monitoreo -d monitoreo -c "SELECT device_id, sequence, COUNT(*) FROM readings GROUP BY device_id, sequence HAVING COUNT(*) > 1;"
 device_id | sequence | count
-----
(0 rows)
```

Finalmente se consulta el registro de auditoría, donde el lote original queda clasificado como ok y su reenvío como duplicate:

Ilustración 54 Auditoría del lote original y de su reenvío

```
david@dsrver:~/Back-App-Monitoreo$ docker compose exec db psql -U monitoreo -d monitoreo -c "SELECT device_external_id, source, result, created_at FROM audit_log ORDER BY id DESC LIMIT 5;"
 device_external_id | source | result | created_at
-----
esp32-01            | mqtt  | duplicate | 2024-07-23 03:34:20.91134+00
esp32-01            | mqtt  | decrypt_failed | 2024-07-22 03:33:48.589241+00
esp32-01            | http  | ok       | 2024-07-22 08:33:04.948517+00
esp32-01            | http  | ok       | 2024-07-22 08:32:28.939469+00
esp32-01            | http  | ok       | 2024-07-22 08:31:52.438921+00
(5 rows)
```

5.2.2.2.2 Toma de datos

Tabla 40 Resultados del Proceso 2 - Deduplicación de lecturas

Nro. de prueba	Secuencia	Envíos realizados	Filas almacenadas	Resultado del reenvío auditoría	Observación	Nro. de prueba
1	700	2	1	duplicate	Reenvío detectado y descartado	1
2	701	2	1	duplicate	Reenvío detectado y descartado	2
3	702	2	1	duplicate	Reenvío detectado y descartado	3

Nro. de prueba	Secuencia	Envíos realizados	Filas almacenadas	Resultado del reenvío en auditoría	Observación	Nro. de prueba
4	703	2	1	duplicate	Reenvío detectado y descartado	4
5	704	2	1	duplicate	Reenvío detectado y descartado	5
6	705	2	1	duplicate	Reenvío detectado y descartado	6
7	706	2	1	duplicate	Reenvío detectado y descartado	7
8	707	2	1	duplicate	Reenvío detectado y descartado	8
9	708	2	1	duplicate	Reenvío detectado y descartado	9
10	709	2	1	duplicate	Reenvío detectado y descartado	10
Total	—	20 envíos	10 filas	10 duplicados detectados	0 registros duplicados en la base de datos (100 % de efectividad)	Total

5.2.2.3 Proceso 3 Detección de manipulación de datos

5.2.2.3.1 Descripción

Esta prueba verifica que un lote cuyo contenido ha sido alterado sea rechazado por el servidor y no llegue a almacenarse. El cifrado autenticado AES-256-GCM incorpora una etiqueta de autenticación; si el contenido se modifica, la verificación falla durante el descifrado y el lote se descarta, dejando constancia en el registro de auditoría

En cada una de las diez pruebas se envía un lote cifrado con un byte alterado:

Ilustración 55 Envío de un lote cifrado manipulado para la prueba de detección de integridad

```
david@dserver:~/Back-App-Monitoreo$ docker compose run --rm --entrypoint python worker \
  /scripts/publish_encrypted_test.py --sequences 800 --tamper
[+] 2/2t 2/22
  ✓ Container monitoreo-mqtt Running
  ✓ Container monitoreo-db Running
Container monitoreo-db Waiting
Container monitoreo-db Healthy
Container back-app-monitoreo-worker-run-1165c5795a97 Creating
Container back-app-monitoreo-worker-run-1165c5795a97 Created
MODO TAMPER: se alteró el ciphertext - se espera decrypt_failed en audit_log
Conectando a mosquito:8883 como esp32-01 (client_id=esp32-sim-01, TLS=True)...
Publicando lote cifrado en vivero/el-carmen/esp32-01/telemetria (QoS 1), secuencias=(800,...)
Publicado OK: deviceId=esp32-01 lecturas=1
david@dserver:~/Back-App-Monitoreo$
```

A continuación, se consulta el registro de auditoría para verificar que el intento quedó documentado:

Ilustración 56 Registro de auditoría de los intentos de manipulación detectados

```
david@dserver:~/Back-App-Monitoreo$ docker compose exec db psql -U monitoreo -d monitoreo -c \
"SELECT created_at, device_external_id, result, detail
FROM audit_log WHERE result='decrypt_failed'
ORDER BY created_at DESC LIMIT 10;"
    created_at      | device_external_id | result      | detail
-----+-----+-----+-----
2026-07-23 03:54:17.682295+00 | esp32-01          | decrypt_failed | autenticación fallida
2026-07-23 03:33:48.589241+00 | esp32-01          | decrypt_failed | autenticación fallida
2026-07-14 15:51:08.99178+00  | esp32-01          | decrypt_failed | autenticación fallida
2026-07-13 07:07:40.929071+00 | esp32-01          | decrypt_failed | autenticación fallida
2026-07-09 02:39:34.858221+00 | esp32-01          | decrypt_failed | sobre inválido
2026-07-09 02:37:50.630145+00 | esp32-01          | decrypt_failed | sobre inválido
2026-07-09 02:37:40.804093+00 | esp32-01          | decrypt_failed | sobre inválido
2026-07-09 02:37:19.640591+00 | esp32-01          | decrypt_failed | sobre inválido
2026-07-09 02:36:26.806284+00 | esp32-01          | decrypt_failed | sobre inválido
2026-07-08 23:18:15.378767+00 | esp32-01          | decrypt_failed | sobre inválido
(10 rows)
```

Ilustración 57 Verificación de que los lotes manipulados no fueron almacenados en la base de datos

```
david@dserver:~/Back-App-Monitoreo$ docker compose exec db psql -U monitoreo -d monitoreo -c "SELECT sequence FROM readings WHERE sequence BETWEEN 800 AND 809;"
sequence
-----
(0 rows)
```

5.2.2.3.2 Toma de datos

Tabla 41 Resultados del Proceso 3 - Detección de manipulación de datos

Nro. de prueba	Secuencia enviada	¿Fue rechazado?	¿Se almacenó?	Resultado en auditoría	Observación	Nro. de prueba
1	800	Sí	No	decrypt_failed	Autenticación fallida; manipulación detectada	1

Nro. de prueba	Secuencia enviada	¿Fue rechazado?	¿Se almacenó?	Resultado en auditoría	Observación	Nro. de prueba
2	801	Sí	No	decrypt_failed	Autenticación fallida; manipulación detectada	2
3	802	Sí	No	decrypt_failed	Autenticación fallida; manipulación detectada	3
4	803	Sí	No	decrypt_failed	Autenticación fallida; manipulación detectada	4
5	804	Sí	No	decrypt_failed	Autenticación fallida; manipulación detectada	5
6	805	Sí	No	decrypt_failed	Autenticación fallida; manipulación detectada	6
7	806	Sí	No	decrypt_failed	Autenticación fallida; manipulación detectada	7
8	807	Sí	No	decrypt_failed	Autenticación fallida; manipulación detectada	8
9	808	Sí	No	decrypt_failed	Autenticación fallida; manipulación detectada	9
10	809	Sí	No	decrypt_failed	Autenticación fallida; manipulación detectada	10
Total	—	10 de 10 rechazados	0 almacenados	100 % de detección	Ninguna lectura alterada ingresó a la base de datos	Total

5.3 Interpretación objetiva

Tabla 42 Mitigación de las causas del problema

Causa identificada en el diagnóstico	Situación previa	Situación con el sistema	Mitigación
Latencia en la consolidación de datos por transcripción manual	El dato tardaba horas o días en estar disponible	Disponible en 0,232 segundos promedio	100 % de eliminación de la etapa de transcripción manual
Pérdida de registros durante la transcripción	Registros perdidos si la anotación no se transcribía	10 de 10 lecturas almacenadas sin pérdida	100 % (0 % de pérdida de datos)
Duplicidad de registros al consolidar cuadernos	Una misma lectura podía anotarse dos veces sin control	20 envíos → 10 registros únicos; 10 reenvíos descartados	100 % de efectividad en deduplicación
Ausencia de garantía de integridad de la información	Un dato podía alterarse sin dejar rastro	10 de 10 lotes manipulados detectados y rechazados	100 % de detección; 0 % de admisión
Falta de trazabilidad de las intervenciones	No existía registro de quién ni cuándo	Cada lote auditado con huella SHA-256 y resultado	100 % de lotes trazables

Recepción y el almacenamiento de telemetría, antes de la implementación del sistema el registro de las variables ambientales del vivero se realizaba de forma manual los estudiantes anotaban las mediciones en cuadernos de campo y posteriormente las transcribían a hojas de cálculo lo cual de modo que un dato tomado en la parcela podía tardar horas o incluso días en quedar disponible para su consulta, y en ocasiones se perdía si la anotación no llegaba a transcribirse. Ya con el sistema implementado el dato viaja automáticamente desde la estación hasta el repositorio central sin intervención humana. En las diez pruebas ejecutadas se almacenaron las diez lecturas enviadas sin pérdida alguna de registros con un tiempo promedio de 0,232 segundos entre la recepción del dato y su disponibilidad en la base de datos. La mejora es sustancial lo que quiere decir que el intervalo entre la medición y su consulta pasó de un proceso de horas a menos de un segundo, con una tasa de pérdida del 0 % así eliminando por completo la etapa de transcripción manual que constituía el principal cuello de botella del proceso anterior.

Con ello se mitigan por completo las dos primeras causas identificadas en el diagnóstico: la latencia en la consolidación de los datos, que pasó de horas a menos de un segundo (100 % de eliminación de la etapa de transcripción manual), y la pérdida de registros durante la transcripción, con un 0 % de pérdida en las pruebas realizadas.

Deduplicación de lecturas, en el procedimiento manual no existía ningún mecanismo que evitara el registro repetido de una misma medición ya que al consolidar la información de varios cuadernos y hojas de cálculo individuales una misma lectura podía anotarse dos veces por distintos estudiantes o duplicarse al copiarla así generando inconsistencias en los informes que solo se detectaban cuando se detectaban esto mediante una revisión visual posterior ya con el sistema, la unicidad de cada lectura está garantizada por el propio motor de base de datos mediante una restricción sobre el par estación-secuencia. En las diez pruebas realizadas se efectuaron veinte envíos (cada lote seguido de su reenvío) y se almacenaron únicamente diez registros con cero duplicados en la base de datos y los diez reenvíos correctamente identificados y descartados. Esto representa una efectividad del 100 % en la prevención de duplicados resultado que además se verificó en condiciones reales de operación donde los reenvíos generados por el gateway fueron igualmente descartados de forma automática. Ya que de este modo queda mitigada la tercera causa del diagnóstico la duplicidad de registros al consolidar la información de distintos cuadernos, con una efectividad del 100 % en la prevención de duplicados.

Detección de manipulación de datos en el registro manual no existía forma alguna de comprobar si un dato había sido alterado después de su anotación lo cual causaba que una cifra podía modificarse borrarse o escribirse incorrectamente sin dejar rastro de manera que la confiabilidad de la información dependía por completo de la buena fe y la atención de quien la registraba. Ya con el sistema la integridad de cada lectura está protegida criptográficamente desde su origen mediante cifrado autenticado y cualquier alteración del contenido incluso de un solo byte impide su descifrado y provoca el rechazo del lote. En las diez pruebas realizadas con lotes manipulados los diez fueron detectados y rechazados, ninguno de ellos llegó a almacenarse en la base de datos y cada intento quedó registrado en la bitácora de auditoría con el detalle de la falla de autenticación. La mejora consiste en pasar de un registro sin ninguna garantía de integridad a uno con 100 % de detección de alteraciones y 0 % de admisión de datos manipulados, además de contar con una trazabilidad completa de cada intento, algo inexistente en el procedimiento anterior. Con esto se mitigan las dos últimas causas identificadas: la ausencia de garantía de integridad de la información, con un 100 % de detección de alteraciones y 0 % de admisión de datos manipulados, y la falta de trazabilidad de las intervenciones, ya que el 100 % de los lotes queda registrado en la bitácora de auditoría con su huella SHA-256

CAPÍTULO VI

6. CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

El diagnóstico de la situación inicial permitió confirmar las falencias existentes en la gestión de registros y el acceso a la información del vivero de cacao tanto la encuesta aplicada a los estudiantes como la entrevista al coordinador evidenciaron que el registro manual generaba demoras en la consolidación de los datos, propiciaba errores de transcripción y dispersaba la información en cuadernos y hojas de cálculo individuales, lo que impedía disponer de una única fuente confiable. Estas carencias afectaban tanto directamente la planificación y evaluación de los procesos formativos como a las decisiones que se tomaban sobre información incompleta o desactualizada y justificaron plenamente la necesidad de un sistema de sincronización automatizada.

La revisión bibliográfica sobre protocolos de comunicación livianos permitió sustentar técnicamente las decisiones de diseño adoptadas el estudio del protocolo MQTT reveló que su modelo de publicación y suscripción, junto con sus niveles de calidad de servicio ya que resulta apropiado para entornos con conectividad intermitente como el rural, pero sin embargo también se evidenció que la garantía de entrega "al menos una vez" implica reenvíos que deben gestionarse explícitamente. Esta comprensión teórica se tradujo en una decisión de diseño concreta que era resolver la deduplicación en el motor de base de datos y no en la lógica de aplicación, lo que otorgó al sistema una robustez que no se habría alcanzado sin ese fundamento previo.

La recolección de datos mediante entrevistas, encuestas y observación directa permitió organizar y comprender el funcionamiento real del vivero antes de proponer una solución. El contacto con el personal técnico y los estudiantes hizo posible identificar no solo las limitaciones evidentes del registro manual sino también las restricciones del entorno como la ausencia de instrumentos de medición además de las condiciones de conectividad y las limitaciones presupuestarias institucionales. Este conocimiento resultó ser determinante para dimensionar una propuesta técnicamente viable y económicamente accesible ya que es ajustada a los recursos realmente disponibles en el sitio y no a un escenario ideal.

El diseño e implementación del modelo basado en el protocolo MQTT se concretó en un sistema funcional que integra la recepción el descifrado la validación como la deduplicación, la auditoría y además el almacenamiento centralizado de la telemetría. La arquitectura resultante incorporó dos vías de ingesta que convergen en una única lógica de procesamiento esto permitió atender tanto a los dispositivos de la red local como al gateway remoto sin exponer el broker a Internet. Además de la protección de la información se resolvió mediante cifrado autenticado desde el origen del dato así complementado con transporte cifrado y control de acceso, configurando una defensa en profundidad acorde a los recursos del entorno.

El análisis del rendimiento del sistema confirmó el cumplimiento de los objetivos planteados mediante evidencia verificable y reproducible. Estas pruebas ejecutadas demostraron que la totalidad de las lecturas enviadas se almacenó sin pérdida con un tiempo promedio de disponibilidad de 0,232 segundos lo que la deduplicación evitó el 100% de los registros repetidos ante reenvíos y a su vez que la totalidad de los lotes manipulados fue detectada y rechazada sin ingresar a la base de datos. Estos resultados están respaldados por la batería de

pruebas automatizadas y la bitácora de auditoría sostienen que el sistema mejora efectivamente la confiabilidad operativa del monitoreo.

6.2 Recomendaciones

Al coordinador del vivero y al responsable técnico del sistema se recomienda programar respaldos automáticos y diarios de la base de datos PostgreSQL y guardarlos fuera del servidor. Como toda la telemetría del vivero vive en un solo repositorio, si el disco falla se pierde el histórico completo que es justo lo que este proyecto vino a construir. Un `pg_dump` diario enviado a un almacenamiento externo probado cada cierto tiempo restaurándolo evita ese riesgo.

Al responsable técnico del sistema se recomienda cambiar cada cierto tiempo las llaves de cifrado de las estaciones y dejar escrito cómo se reemplazan. El cifrado AES-256-GCM que usa el sistema es fuerte, pero su seguridad depende de que la llave siga siendo secreta; una llave que no se cambia en años se vuelve un riesgo si algún día se filtra. Un ciclo de renovación y una forma segura de cargarla en los ESP32 mantendría la protección en el tiempo.

Al responsable técnico del sistema se recomienda implementar un mecanismo que avise cuando algo se caiga: que se detenga la ingesta que falle un contenedor o que una estación como esp32-01 lleve rato sin reportar. Hoy toca revisar a mano con consultas a la tabla `audit_log`, así que una falla puede pasar horas sin que nadie la note, perdiendo mediciones. Una alerta automática permitiría reaccionar a tiempo.

A los futuros estudiantes que continúen el proyecto se recomienda seguir ampliando la batería de pruebas automatizadas a medida que el sistema crezca, agregando un caso nuevo por cada función que se sume. En este proyecto esas pruebas ya detectaron a tiempo problemas como el conflicto de nombres entre paquetes pues mantenerlas es la forma más barata de que un cambio futuro no rompa lo que hoy funciona y le facilita el trabajo a quien dé mantenimiento después.

Al coordinador del vivero se recomienda definir cada cuánto se archivan las lecturas antiguas. El sistema guarda datos sin parar en las pruebas ya se ingresaron secuencias hasta la 809 y sigue sumando con el gateway en vivo así que la tabla de lecturas crecerá indefinidamente y

las consultas se volverán lentas. Pasar las lecturas viejas a una tabla de solo consulta mantendría el histórico completo sin sacrificar velocidad.

A la carrera de Ingeniería de Software se recomienda, como trabajo a futuro, estudiar un canal autenticado que deje al agricultor ver la última medición sin Internet esto sin romper el cifrado de extremo a extremo. Hoy el dato solo se lee después de que el servidor lo descifra así que un operario parado junto a la estación sin señal no alcanza a ver el valor recién medido. Resolver eso conservando la integridad le daría más utilidad al sistema en el campo.

BIBLIOGRAFÍA

- Arias, F. G. (2012). *El proyecto de investigación: Introducción a la metodología científica*. Caracas: Episteme. Obtenido de https://books.google.com/books/about/El_Proyecto_de_Investigaci%C3%B3n_Introducci.html?id=W5n0BgAAQBAJ
- ATLAS.ti. (2025). *ATLAS.ti*. Obtenido de La entrevista como poderoso método de investigación: <https://atlasti.com/es/guias/guia-investigacion-cualitativa-parte-1/entrevistas>
- Banks, A., & Gupta, R. (2014). *MQTT Version 3.1.1. OASIS Standard*. OASIS.
- Bass, L., Clements, P., & Kazman, R. (2013). *Software Architecture in Practice*. Addison-Wesley.
- Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, 21(5), 61-72.
- Ceruzzi, P. (2003). *History of Modern Computing*. Cambridge: MIT Press.
- Chugá Rosero, M. A. (2022). *Diseño e Implementación de un Sistema de Riego Inteligente y monitoreo de variables edafológicas y meteorológicas activado por IOT*. Quito: Universidad Internacional SEK. Obtenido de <https://repositorio.uisek.edu.ec/handle/123456789/4876>
- Date, C. J. (2019). *Database Design and Relational Theory: Normal Forms and All That Jazz*. (Vol. 2). Berkeley, California.: Apress. Obtenido de <https://link.springer.com/book/10.1007/978-1-4842-5540-7>
- Eclipse Foundation. (2025). Obtenido de Mosquitto Documentation: <https://mosquitto.org/documentation/>
- Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database Systems*. Pearson.

EMQ Technologies Co., Ltd. (2025). Obtenido de EMQX documentation: <https://docs.emqx.com/en/>

Flores Cruz, K. A., & López Ortiz, I. (2023). *Diseño e implementación de una plataforma Web para recolección y visualización de datos en redes de sensores basados en MQTT*. Ciudad de Mexico: Universidad Nacional Autonoma de Mexico. Obtenido de <https://repositorio.unam.mx/contenidos/3718024>

Hernández Sampieri Roberto, F. C. (2014). *Metodología de la investigación* (6 ed.). México: McGraw-Hill Education. Obtenido de <https://dialnet.unirioja.es/servlet/libro?codigo=775008>

Hernández-Florez, N. J., & Klimenko, O. (2025). Metodología de la investigación, enfoque cualitativo, cuantitativo y mixto. *Nexus: Multidisciplinary Research Journal (MIR)*, 2(3), 1-19. Obtenido de <https://nexushouseeditorial.com/index.php/nexus/article/view/24>

HiveMQ. (2025). Obtenido de MQTT Essentials: <https://www.hivemq.com/mqtt-essentials/>

InfluxData Inc. (2025). *InfluxDB documentation*. Obtenido de <https://docs.influxdata.com/>

Kleppmann, M. (2017). *Designing Data-Intensive Applications*. Sebastopol: O'Reilly Media.

Larman, C., & Basili, V. R. (2003). Iterative and incremental development: A brief history. *Computer*, 36(6), 47-56.

Martínez, T. C. (2024). ¿Entrevista o encuesta?: Una diferencia necesaria. *Nueva Época Revista Latinoamericana de Comunicación Social*, 1-15.

Michelson, B. (2006). *Event-Driven Architecture Overview*. Patricia Seybold Group. Obtenido de <http://dx.doi.org/10.1571/bda2-2-06cc>

Ortega, I. L. (2022). *IoT y agricultura 4.0: Control de cultivos en función de la humedad del suelo, mediante sensores y actuadores controlados con protocolo MQTT*. España:

Universidad Miguel Hernández. Obtenido de
<https://dspace.umh.es/handle/11000/28563>

Özsu, M. T., & Valduriez, P. (2011). *Principles of Distributed Database Systems* (Vol. 2). New York: Springer.

Pressman, R. S., & Maxim, B. R. (2019). *Software Engineering: A Practitioner's Approach* (Vol. 9). McGraw-Hill Education.

Richardson, L., & Amundsen, M. (2013). *RESTful Web APIs*. O'Reilly Media.

SalusPlay. (2025). *Tema 5. La Muestra y la Población de estudio*. Obtenido de SalusPlay:
<https://www.salusplay.com/apuntes/apuntes-metodologia-de-la-investigacion/tema-5-la-muestra-y-la-poblacion-de-estudio>

Sánchez y Sánchez, C. L., & Hernández Olvera, K. (2024). La encuesta como técnica de investigación en Ciencia Política. *Revista Mexicana de Opinión Pública*(37), 13-31. Obtenido de <https://doi.org/10.22201/fcpys.24484911e.2024.37.88492>

Shapiro, M., Preguiça, N., Baquero, C., & Zawirski, M. (2011). *A comprehensive study of Convergent and Commutative Replicated Data Types*. INRIA.

Sommerville, I. (2016). *Software Engineering* (Vol. 10). Pearson.

Tanenbaum, A. S., & van Steen, M. (2017). *Distributed Systems*. London: Pearson.

Tarrillo Saldaña, O., Mejía Huamán, J., Dávila Mego, J. S., Pintado Castillo, C. A., Tapia Idrogo, C. E., Chilón Camacho, W. M., & Velez Escobar, S. B. (2024). *Metodología de la investigación una mirada global: Ejemplos prácticos*. Centro de Investigación y Desarrollo (CID). Obtenido de https://doi.org/10.37811/cli_w1078

Timescale Inc. (2025). *Time-series data validation*. Obtenido de TimescaleDB Docs: <https://docs.timescale.com/>

ANEXOS

Anexo A: Asignación de tutor

Anexo A: Asignación de tutor

 Outlook

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

Desde NOTIFICACIONES TITULACION <notificaciones.titulacion@uleam.edu.ec>
Fecha Mié 13/06/2023 03:53 PM
Para ARNEVALD HERNANDEZ ROMULO DANILU <romulo.arnevald@uleam.edu.ec>
CC GUERRERO SALVATERRA DAVID RICARDO <+d200254627@uleam.edu.ec>; REASCOS PINCHAO RAUL SAED <raul.reascos@uleam.edu.ec>


Universidad Laica Eloy Alfaro de Manabí

Período 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 INFORMATICO CON MQTT ORIENTADO A LA SINCRONIZACION DE DATOS EN EL VIBRIO DE CACAO 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: ARNEVALD HERNANDEZ ROMULO DANILU

Apellidos y nombres del estudiante: GUERRERO SALVATERRA DAVID RICARDO

Carrera: SOFTWARE 2024 - NS - (EL CARMEN)

Período de Inducción: Período 2025-2

Se van a 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-Titulacion-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

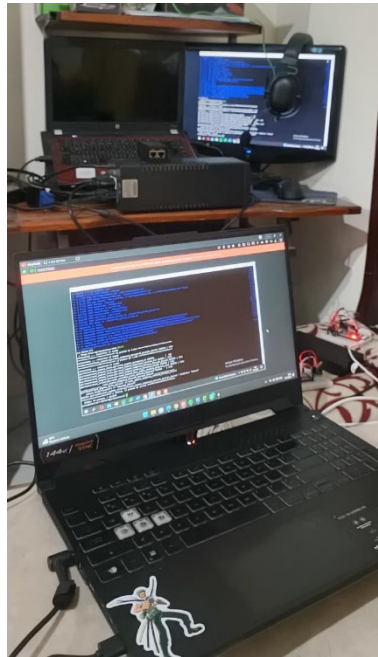
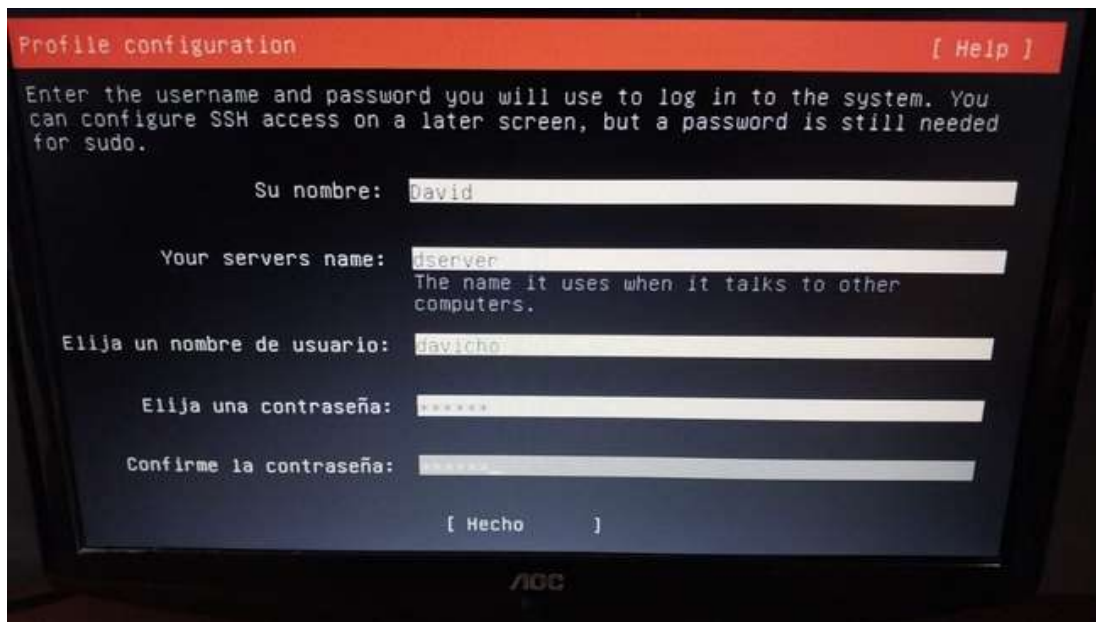


Imagen de manipulación del servidor local y conexión remota

Ilustración 58 Creación del servidor



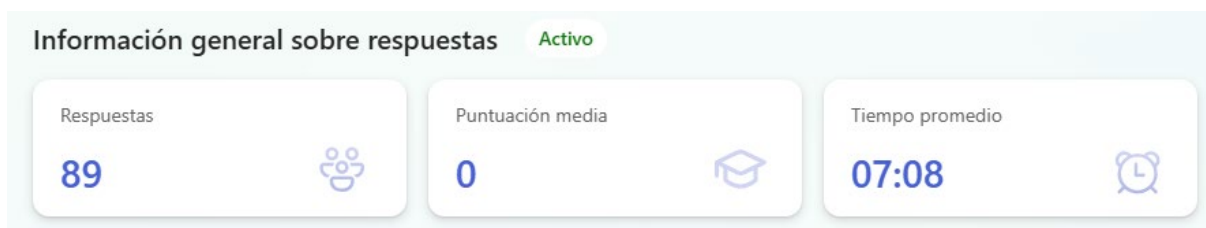
Anexo C: Evidencia de aplicación de encuestas y entrevistas

Anexo C Evidencia de aplicación de encuestas y entrevistas



Entrevista con el coordinador de la carrera y grupo de proyecto de monitoreo ambiental automatizado

Ilustración 59 Cantidad de estudiantes entrevistadas y tiempo promedio



Cantidad de estudiantes encuestados y tiempo promedio

Ilustración 60 Encuesta realizada a los estudiantes de agropecuaria



Encuesta realizada a los estudiantes de agropecuaria

Anexo D: Estructura de la entrevista aplicada

Anexo D Estructura de la entrevista aplicada

ENTREVISTA

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

Estructura de las preguntas planteadas para la entrevista al coordinador encargado del vivero Lastenia Vera, Mg. Pedro Nivelá

Anexo E: Estructura de la encuesta aplicada

Anexo E Estructura de la encuesta aplicada

ENCUESTA DE PROYECTO MONITOREO AMBIENTAL AUTOMATIZADO

* Obligatoria

* Este formulario registrará su nombre, escriba su nombre.

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

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

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

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

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

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

4. ¿Con que facilidad puede identificar las enfermedades del cultivo de cacao mediante la observación directa? *

- ☐ Nunca
- ☐ Rara vez
- ☐ A veces
- ☐ Frecuentemente
- ☐ Siempre

5. ¿Conoce usted que enfermedades pueden propagarse en el cultivo de cacao debido a los cambios climático ? *

- ☐ Si
- ☐ No

6. ¿Con que frecuencia ha aplicado algún producto insecticida o abono sin conocer si la planta realmente lo necesita? *

- ☐ Nunca
- ☐ Rara vez
- ☐ A veces
- ☐ Frecuentemente
- ☐ Siempre

7. ¿Cómo evaluaría la eficacia del método actual de monitoreo manual de variables ambientales (temperatura, humedad) para el desarrollo de las plántulas de cacao? *

- ☐ Muy eficaz
- ☐ Eficaz
- ☐ Ni eficaz ni ineficaz
- ☐ Ineficaz
- ☐ Muy ineficaz

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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



Estructura realizada de las preguntas para la encuesta aplicada a los estudiantes de la carrera de agropecuaria

Anexo F: Certificado Sistema Antiplagio

Anexo F Certificado Sistema Antiplagio



Certificado de análisis
Compilatio Magister+ | ULEAM-ECU

Trabajo de titulación Guerrero
ID : cfc5a76bf63494d1ea9054eea7d0480890d4e034



Nombre del fichero : Trabajo de titulacion Guerrero.txt
Tamaño del archivo original : 2,05 MB
Número de palabras : 33.491

Depositante : David Guerrero Salvatierra
Autor : David Guerrero Salvatierra
Fecha de depósito : 24 de julio de 2026
Tipo de carga : url_submission
fecha de fin de análisis : 25 de julio de 2026

Resumen (sección 1/2)

Localización de los textos sospechosos en el documento :



Incluido en el porcentaje de textos sospechosos :

Similitudes <1%

Pasajes con similitudes a fuentes encontradas en diferentes colecciones.



Detección de IA 7%

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



Idiomas no reconocidos 4%

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



No incluido en el porcentaje de textos sospechosos :

Textos entre comillas 2%

Pasajes entre comillas, a menudo indicativos de una cita.

Fuentes de similitudes (sección 2/2)

Similitudes <1%

Pasajes con similitudes a fuentes encontradas en diferentes colecciones.



Fuente con similitudes fortuitas

Nº	Descripciones	Similitudes	Ubicaciones
1	Plan de contingencia para equipos informáticos de los estudiantes... repositorio.uleam.edu.ec/handle/123456789/9147	<1%	
2	TESIS HEYDER #422a46 Viene de de mi grupo	<1%	
3	www.europeanscrum.org www.europeanscrum.org/uploads/2/4/5/1/24513648/gu%CC%81a_oficial...	<1%	
4	La Muestra y la Población de estudio SalusPlay www.salusplay.com/apuntes/apuntes-metodologia-de-la-investigacion/te...	<1%	

GLOSARIO

AAD (Additional Authenticated Data): Dato adicional que se incorpora al proceso de cifrado autenticado sin ser cifrado, pero que queda protegido contra alteraciones. En este proyecto se emplea el identificador del dispositivo, vinculando cada mensaje a su estación de origen.

AES-256-GCM: Algoritmo de cifrado simétrico autenticado que combina confidencialidad, integridad y autenticidad en una sola operación. La longitud de clave de 256 bits y el modo GCM permiten detectar cualquier alteración del contenido cifrado.

Alembic: Herramienta de migraciones para bases de datos relacionales que permite versionar los cambios del esquema y aplicarlos de forma controlada y reproducible.

API: (Interfaz de Programación de Aplicaciones). Conjunto de definiciones y reglas que permite que dos aplicaciones se comuniquen entre sí, exponiendo funciones o datos sin revelar su implementación interna.

API Key: (llave de aplicación). Credencial en forma de cadena de caracteres que identifica y autoriza a un cliente para consumir un servicio. Se envía en la cabecera de cada petición.

API REST: Estilo de arquitectura para servicios web que utiliza los métodos del protocolo HTTP para operar sobre recursos identificados mediante direcciones únicas.

Auditoría (bitácora): Registro cronológico de los eventos ocurridos en un sistema. En este proyecto documenta el resultado del procesamiento de cada lote recibido.

Backlog: Lista ordenada de tareas o funcionalidades pendientes en la metodología Scrum. El Product Backlog contiene el trabajo del proyecto completo y el Sprint Backlog el de una iteración.

Base64: Método de codificación que representa datos binarios mediante caracteres imprimibles, permitiendo transmitirlos por canales diseñados para texto.

Bluetooth SPP (Serial Port Profile): Perfil del estándar Bluetooth clásico que emula una conexión serial entre dos dispositivos, empleado para el enlace entre el gateway y el teléfono.

Bróker MQTT: Servidor intermediario que recibe los mensajes publicados por los dispositivos y los entrega a los suscriptores correspondientes, desacoplando a emisores y receptores.

Caja blanca (prueba de): Técnica de prueba que examina las rutas internas de ejecución del código, verificando que cada camino lógico se comporte según lo previsto.

Caja negra (prueba de): Técnica de prueba que evalúa el sistema desde el exterior, comprobando que las salidas correspondan a las entradas esperadas sin considerar su implementación interna.

Cifrado de extremo a extremo: Esquema de protección en el que la información se cifra en su origen y solo se descifra en su destino final, de modo que ningún intermediario puede acceder a su contenido.

Contenedor: Unidad de software que empaqueta una aplicación junto con sus dependencias, permitiendo que se ejecute de forma aislada y reproducible en cualquier entorno.

Deduplicación: Proceso mediante el cual se impide el almacenamiento de registros repetidos, conservando una única copia de cada dato.

DHT22: Sensor digital que mide temperatura y humedad relativa del ambiente, empleado en la estación meteorológica del vivero.

Docker: Plataforma de virtualización ligera basada en contenedores, que permite desplegar y orquestar servicios de forma aislada sobre un mismo servidor.

Endpoint: Dirección concreta de una API a la que se dirigen las peticiones para acceder a un recurso o ejecutar una operación determinada.

ESP32: Microcontrolador de bajo costo con conectividad inalámbrica integrada, utilizado tanto en la estación de medición como en el gateway.

FastAPI: Framework de desarrollo web en lenguaje Python orientado a la construcción de interfaces de programación de aplicaciones, con validación automática de datos.

Firmware: Programa almacenado de forma permanente en un dispositivo electrónico, que controla su funcionamiento a bajo nivel.

Gateway: Dispositivo que actúa como puente entre dos redes o tecnologías de comunicación distintas. En este proyecto recibe los datos por radiofrecuencia y los retransmite hacia el servidor.

Hash: Valor de longitud fija calculado a partir de un conjunto de datos mediante una función matemática. Cualquier modificación del contenido produce un valor distinto, lo que permite verificar su integridad.

HTTP / HTTPS: Protocolo de transferencia de información en la web. Su variante segura (HTTPS) añade una capa de cifrado que protege los datos en tránsito.

IoT (Internet de las Cosas): Paradigma en el que objetos físicos dotados de sensores y conectividad recopilan e intercambian información a través de una red.

JSON: Formato ligero de intercambio de datos basado en texto, estructurado en pares de clave y valor, legible tanto por personas como por programas.

Latencia: Tiempo transcurrido entre el envío de un dato y su disponibilidad en el destino.

LoRa: Tecnología de comunicación inalámbrica de largo alcance y bajo consumo, diseñada para transmitir pequeñas cantidades de datos en entornos rurales o extensos.

Lote (batch): Conjunto de lecturas agrupadas que se transmiten y procesan como una sola unidad.

Mosquitto: Implementación de código abierto de un bróker MQTT, utilizada en este proyecto como servidor de mensajería.

MQTT: Protocolo de mensajería ligero basado en el modelo de publicación y suscripción, diseñado para dispositivos con recursos limitados y redes de ancho de banda reducido.

Nonce: Valor único e irrepetible que se emplea una sola vez en una operación criptográfica, garantizando que dos mensajes idénticos produzcan resultados cifrados distintos.

NVS (Almacenamiento No Volátil): Área de memoria del microcontrolador que conserva la información aun después de un reinicio o corte de energía.

Offline-first: Enfoque de diseño en el que la aplicación almacena la información localmente y opera sin conexión, sincronizándose con el servidor cuando la conectividad se restablece.

Payload: Porción de un mensaje que contiene la información útil, en contraposición a los datos de control o encabezado.

PostgreSQL: Sistema gestor de bases de datos relacionales de código abierto, empleado como repositorio central del sistema.

Pytest: Herramienta del lenguaje Python para escribir y ejecutar pruebas automatizadas.

QoS (Calidad de Servicio): Nivel de garantía de entrega que ofrece el protocolo MQTT. El nivel 1 asegura que el mensaje llegue al menos una vez, admitiendo reenvíos.

RSSI: Indicador de la intensidad de la señal recibida en una comunicación inalámbrica, expresado en decibelios.

Scrum: Marco de trabajo ágil para el desarrollo de software, basado en iteraciones cortas denominadas sprints, en las que se entregan incrementos funcionales.

SHA-256: Función criptográfica que genera una huella digital de 256 bits a partir de un conjunto de datos, utilizada para verificar su integridad.

SNR (Relación Señal-Ruido): Medida que compara la potencia de la señal útil frente al ruido de fondo en una transmisión.

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

SQLite Motor de base de datos relacional ligero que almacena la información en un único archivo, empleado en el dispositivo móvil.

Telemetría: Conjunto de mediciones registradas de forma remota y transmitidas hacia un punto central para su almacenamiento y análisis.

TLS (Seguridad de la Capa de Transporte): Protocolo criptográfico que cifra la comunicación entre dos extremos de una red, protegiéndola frente a interceptación y alteración.

Tópico: Cadena jerárquica que identifica el canal en el que se publican y del que se reciben mensajes en el protocolo MQTT.