



UNIVERSIDAD
CATÓLICA
DE CUENCA

UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**FACULTAD DE INFORMÁTICA, CIENCIAS DE LA
COMPUTACIÓN E INNOVACIÓN TECNOLÓGICA**

CARRERA DE SOFTWARE

**Prototipo ERP móvil para la gestión de facturación e inventario en una hacienda
bananera de La Troncal, Cañar**

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO DE SOFTWARE**

AUTOR: VICTOR ALFONSO CASTRO TOMALA

DIRECTOR: ING. LUIS FERNANDO PINOS CASTILLO.

CUENCA - ECUADOR

2026

DIOS, PATRIA, CULTURA Y DESARROLLO



UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**FACULTAD DE INFORMÁTICA, CIENCIAS DE LA
COMPUTACIÓN E INNOVACIÓN TECNOLÓGICA**

CARRERA DE SOFTWARE

**Prototipo ERP móvil para la gestión de facturación e inventario en una hacienda
bananera de La Troncal, Cañar**

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO DE SOFTWARE**

AUTOR: VICTOR ALFONSO CASTRO TOMALA

DIRECTOR: ING. LUIS FERNANDO PINOS CASTILLO.

CUENCA – ECUADOR

2026

DIOS, PATRIA, CULTURA Y DESARROLLO

 Universidad Católica de Cuenca	DECLARATORIA DE AUTORÍA
---	--------------------------------

Yo, Víctor Alfonso Castro Tomala, con cédula de identidad N.º 0928595479, declaro bajo juramento que el presente Trabajo de Titulación es de mi autoría, que no ha sido presentado previamente para ningún grado o calificación profesional, y que las fuentes consultadas han sido debidamente referenciadas en el documento.

Cuenca, **03 de septiembre de 2026**



F:

Víctor Alfonso Castro Tomala

C.I. 0928595479

 Universidad Católica de Cuenca	CERTIFICADO DEL TUTOR
---	------------------------------

Ing. Luis Fernando Pinos Castillo, en calidad de Tutor del presente Trabajo de Titulación, certifico que el documento titulado “*Prototipo ERP móvil para la gestión de facturación e inventario en una hacienda bananera de La Troncal, Cañar*”, desarrollado por el estudiante Víctor Alfonso Castro Tomala, ha sido revisado en su totalidad y cumple con los requisitos metodológicos y científicos exigidos por la Universidad Católica de Cuenca, por lo que autorizo su presentación.



F:

Msg. Luis Fernando Pinos Castillo

C.I. 0301829255

Dedicatoria

Dedico este trabajo de titulación, con todo mi amor, respeto y gratitud, a mis padres, Maribel Tomala y Victor Castro, quienes han sido el pilar fundamental de mi vida y mi mayor apoyo durante todo este camino académico.

Gracias por cada esfuerzo, sacrificio, consejo y palabra de aliento que me brindaron a lo largo de estos años. Su confianza en mí fue una motivación constante para seguir adelante, incluso en aquellos momentos en los que el cansancio, las dificultades o las preocupaciones hicieron que alcanzar esta meta pareciera más difícil.

A ustedes les debo gran parte de la persona que soy hoy. Su ejemplo de trabajo, responsabilidad, perseverancia y dedicación me ha enseñado que los objetivos se alcanzan con esfuerzo constante y que, aun cuando el camino presente obstáculos, siempre existe una razón para continuar.

Cada etapa superada durante mi formación universitaria llevó consigo su apoyo. Por eso, este logro no lo considero únicamente mío, sino también de ustedes, porque estuvieron presentes desde el inicio, acompañándome de diferentes maneras y brindándome la fortaleza necesaria para continuar hasta llegar a este momento.

Este trabajo representa el cierre de una etapa muy importante de mi vida, llena de aprendizajes, experiencias, desafíos y crecimiento tanto personal como profesional. También representa el fruto de los valores y enseñanzas que ustedes me inculcaron desde pequeño.

Espero que este logro sea una forma de recompensar, aunque sea en una pequeña parte, todo lo que han hecho por mí y que puedan sentirse orgullosos al saber que cada sacrificio realizado contribuyó a que hoy pueda cumplir una de mis metas más importantes.

Con todo mi cariño y gratitud, este logro es para ustedes.

Victor Alfonso Castro Tomala

Agradecimiento

Al culminar esta etapa de mi formación profesional, deseo expresar mi más sincero agradecimiento a todas las personas e instituciones que, de una u otra manera, formaron parte de este proceso y contribuyeron para que pudiera alcanzar esta meta.

En primer lugar, agradezco profundamente a mis padres, Maribel Tomala y Victor Castro, por acompañarme durante toda mi formación universitaria. Gracias por su apoyo incondicional, por creer en mis capacidades, por motivarme a continuar y por estar presentes en cada una de las etapas que atravesé durante estos años. Su esfuerzo y confianza fueron fundamentales para que hoy pueda culminar mi carrera y presentar este trabajo de titulación.

Expreso también mi especial agradecimiento al Ing. Luis Fernando Pinos Castillo, tutor de este trabajo de titulación, por su orientación, paciencia, tiempo y conocimientos brindados durante el desarrollo del proyecto. Sus observaciones, recomendaciones y acompañamiento fueron importantes para mejorar cada etapa del trabajo y para culminarlo de manera satisfactoria.

De igual manera, agradezco a la Universidad Católica de Cuenca, institución que me permitió formarme académica y profesionalmente, brindándome los conocimientos, herramientas y espacios necesarios para desarrollar mis capacidades dentro del área de la Ingeniería de Software.

Mi agradecimiento se extiende a todos los docentes de la carrera de Software, quienes a lo largo de mi formación compartieron sus conocimientos y experiencias. Cada asignatura, proyecto y desafío académico aportó de alguna manera a mi crecimiento profesional y permitió desarrollar las competencias necesarias para llevar a cabo este trabajo.

Agradezco especialmente a la Bananera San Diego, a su propietario y a sus trabajadores por brindarme la apertura, confianza y colaboración necesarias para desarrollar este proyecto en un entorno real. Su disposición para proporcionar información sobre los procesos de la hacienda, participar en las actividades de levantamiento de requerimientos y colaborar durante la evaluación del sistema permitió que el proyecto respondiera a necesidades reales de la organización.

La realización de este trabajo de titulación significó aplicar muchos de los conocimientos adquiridos durante mi carrera y enfrentar nuevos desafíos relacionados con el análisis, diseño, desarrollo y evaluación de una solución tecnológica. Por ello, agradezco también a todas aquellas personas que, mediante un consejo, una recomendación, una palabra de ánimo o simplemente su compañía, aportaron para que pudiera continuar avanzando.

Finalmente, agradezco a quienes estuvieron presentes durante este camino y celebraron conmigo cada pequeño avance. Culminar esta etapa representa muchos años de esfuerzo, aprendizaje y perseverancia, y haber contado con el apoyo de tantas personas hizo que el camino fuera mucho más significativo.

A todos quienes formaron parte de esta etapa de mi vida y contribuyeron de alguna manera para alcanzar esta meta, les expreso mi más profundo y sincero agradecimiento.

Victor Alfonso Castro Tomala

Resumen

El presente trabajo describe el diseño y desarrollo de un prototipo ERP móvil para apoyar la gestión de facturación e inventario en la Hacienda Bananera San Diego, ubicada en La Troncal, provincia de Cañar. La organización opera bajo un esquema de propiedad compartida y mantenía registros manuales de ventas, inventario y mano de obra, situación que generaba dispersión de la información, demoras e inconsistencias en los cálculos. El prototipo fue desarrollado con Expo/React Native para la aplicación móvil, Spring Boot para el backend y PostgreSQL para la persistencia de datos, mediante una adaptación de Scrum organizada en cuatro sprints. La solución integra seis roles de usuario, registro de embarques y producción por dueño y finca, cierre financiero con distribución proporcional de costos, inventario diferenciado por propietario, gastos, cuadrilla, jornales, planilla semanal, comprobantes internos en PDF, auditoría y registro sin conexión de embarques con sincronización posterior. La ampliación del alcance funcional se gestionó mediante la actualización incremental del Product Backlog, sin modificar el problema ni el objetivo general aprobados. La evaluación de usabilidad mediante el cuestionario System Usability Scale obtuvo una puntuación promedio de 82,50 puntos sobre 100 ($n = 5$; $DE = 2,50$), superior al valor de referencia de 68 puntos. Las pruebas funcionales verificaron las principales reglas de negocio y controles de acceso. La comparación entre el procedimiento manual y el prototipo mostró una reducción promedio de 74,22 % en el tiempo de registro y una disminución de 91,67 % en las inconsistencias observadas. Estos resultados provienen de un diseño preexperimental de un solo grupo, con medición antes y después y sin grupo de control, por lo que describen el caso estudiado y resultan compatibles con un efecto positivo del prototipo sin permitir atribuirle la mejora de forma exclusiva.

Palabras clave: ERP móvil, gestión de inventario, hacienda bananera, React Native, usabilidad.

Abstract

This work describes the design and development of a mobile ERP prototype to support billing and inventory management at the San Diego Banana Farm, located in La Troncal, Cañar. The organization operates under a shared ownership scheme and previously relied on manual records for sales, inventory, and labor, which caused scattered information, delays, and calculation inconsistencies. The prototype was developed with Expo/React Native for the mobile application, Spring Boot for the backend, and PostgreSQL for data persistence, using an adapted Scrum process organized into four sprints. The solution includes six user roles, shipment and production records by owner and farm, financial closure with proportional cost distribution, owner-specific inventory, expenses, shipment crews, day labor records, weekly payroll, internal PDF receipts, auditing, and offline shipment registration with later synchronization. The functional scope was expanded through incremental Product Backlog updates without changing the approved research problem or general objective. The usability evaluation using the System Usability Scale obtained an average score of 82.5 out of 100 ($n = 5$; $SD = 2.5$), above the reference value of 68. Functional testing verified the main business rules and access controls. The comparison between the manual procedure and the prototype showed an average reduction of 74.22% in recording time and a 91.67% decrease in the observed inconsistencies. These results come from a one-group pre-experimental design with pretest and posttest measurements and no control group, so they describe the case under study and are consistent with a positive effect of the prototype without allowing the improvement to be attributed to it exclusively.

Keywords: *mobile ERP, inventory management, banana farm, React Native, usability.*

Índice

Índice	VIII
1. Introducción.....	1
1.1. Antecedentes.....	1
1.2. Planteamiento del problema.....	2
1.2.1. Situación actual.....	2
1.2.2. Problemas detectados.....	2
1.2.3. Consecuencias	2
1.3. Formulación del problema.....	3
1.4. Objetivos.....	3
1.4.1. Objetivo general	3
1.4.2. Objetivos específicos	3
1.5. Justificación	4
1.6. Alcance y limitaciones	4
2. Marco Teórico.....	6
2.1. Sistemas de Información Empresarial	6
2.2. Sistemas ERP (Enterprise Resource Planning)	6
2.2.1 Características de los sistemas ERP	6
2.2.2 Arquitectura de un sistema ERP	7
2.2.3 Módulos de un sistema ERP	7
2.2.4 ERP móviles	7
2.2.5 Gestión de Inventarios	7
2.2.6 Kardex y control de existencias.....	8
2.2.7 Entradas y salidas	8
2.2.8 Stock mínimo.....	8
2.2.9 Categorías y proveedores.....	8
2.3. Productividad y Calidad en la Producción Bananera	9
2.3.1 Calendario de enfunde y cintas de color	9
2.4. Facturación Electrónica en Ecuador	10
2.4.1 El SRI y la facturación electrónica.....	10
2.4.2 Formato XML y firma digital.....	10
2.4.3 Ambientes de pruebas y producción	10
2.5. Arquitectura de Software.....	11

2.5.1	Arquitectura Cliente-Servidor	11
2.5.2	Arquitectura REST API.....	11
2.5.3	Patrón Modelo-Vista-Controlador (MVC).....	11
2.5.4	Aplicaciones con sincronización offline.....	11
2.5.5	Arquitectura offline-first y sincronización diferida.....	12
2.5.6	Clean Architecture y diseño orientado al dominio	12
2.5.7	Gestión del estado en aplicaciones móviles	13
2.6.	Tecnologías Utilizadas	13
2.6.1	Android y el ecosistema móvil.....	13
2.6.2	Desarrollo nativo frente a multiplataforma	13
2.6.3	React Native y Expo	14
2.6.4	Spring Boot.....	14
2.6.5	PostgreSQL.....	14
2.6.6	JWT y seguridad	15
2.6.7	Git y control de versiones.....	15
2.6.8	Seguridad en aplicaciones móviles y OWASP Mobile Top 10.....	15
2.6.9	Ciclo de vida de los tokens de sesión	16
2.6.10	Criterios de selección y compromisos de la pila tecnológica.....	16
2.7.	Metodologías Ágiles.....	17
2.7.1	Scrum.....	17
2.7.2	Usabilidad y Evaluación SUS.....	17
2.8.	Trabajos Relacionados	18
2.8.1	Aplicación móvil para gestión institucional	18
2.8.2	React Native para desarrollo multiplataforma	18
2.8.3	Digitalización de PYMES agroindustriales.....	18
2.8.4	Alternativas ERP de Código Abierto y Justificación del Desarrollo a Medida	18
3.	Metodología.....	20
3.1.	Tipo de Investigación.....	21
3.2.	Enfoque de Investigación	21
3.3.	Población y Muestra.....	22
3.4.	Técnicas de Recolección de Información	22
3.4.1	Entrevista dirigida.....	23
3.4.2	Observación directa	23

3.4.3	Encuesta de usabilidad (SUS)	24
3.5.	Metodología de Desarrollo de Software	24
3.5.1	Selección de Scrum	24
3.5.2	Adaptación de Scrum a un equipo de un solo desarrollador	24
3.5.3	Organización en sprints.....	25
3.5.4	Evolución del alcance durante el desarrollo	26
3.5.5	Procedimiento de desarrollo	27
3.5.6	Herramientas de desarrollo	28
3.6.	Diseño de la Validación Empírica.....	28
3.6.1	Diseño preexperimental de un solo grupo.....	28
3.6.2	Justificación metodológica de la meta del 70 %	29
3.6.3	Protocolo de medición de tiempos y errores	29
3.6.4	Tratamiento estadístico de los datos.....	30
3.6.5	Amenazas a la validez.....	31
4.	Análisis y Diseño del Sistema.....	31
4.1.	Análisis del Sistema Actual.....	31
4.1.1	Proceso actual de venta (embarque)	32
4.1.2	Proceso propuesto de gestión del embarque	33
4.1.3	Proceso actual de inventario	34
4.1.4	Proceso propuesto de gestión del inventario.....	35
4.2.	Diseño de la Arquitectura.....	36
4.2.1	Diagrama de capas.....	37
4.2.2	Diagrama de componentes.....	38
4.2.2	Diagrama de despliegue.....	39
4.3.	Diseño de la Base de Datos	40
4.3.1	Modelo entidad-relación.....	40
4.3.2	Diccionario de datos	41
5.	Desarrollo e Implementación del Sistema.....	52
5.1.	Especificación de Requisitos de Software (SRS).....	52
5.1.1	Introducción del SRS.....	52
5.1.2	Descripción general	53
5.1.3	Requisitos funcionales.....	54
5.1.4	Requisitos no funcionales.....	58

5.1.5	Historias de usuario	59
5.1.6	Diagrama de casos de uso.....	61
5.2.	Implementación del Sistema.....	62
5.2.1	Ambiente de desarrollo.....	62
5.2.2	Estructura del backend.....	63
5.2.3	Seguridad y control de acceso	63
5.2.4	Ciclo de vida del token de sesión implementado	64
5.2.5	Módulo de inventario	65
5.2.6	Módulo de embarques y cierre financiero	65
5.2.7	Módulo de gastos.....	69
5.2.8	Módulo de mano de obra y planilla semanal	69
5.2.9	Módulo de facturación y comprobantes	70
5.2.10	Estructura del frontend y navegación por roles	71
5.2.11	Modo sin conexión y sincronización	72
5.2.12	Reportes.....	73
5.2.13	Documentación y despliegue de la API.....	74
6.	Pruebas y Resultados	74
6.1.	Pruebas funcionales	74
6.2.	Pruebas de la API REST.....	76
6.2.1	Consideraciones sobre el desempeño técnico.....	77
6.2.2	Evaluación de usabilidad mediante SUS.....	78
6.3.	Comparación de tiempos de registro	81
6.4.	Comparación de errores de registro.....	85
6.5.	Discusión de resultados	86
7.	Conclusiones y Recomendaciones.....	90
7.1.	Conclusiones.....	90
1.7.	Recomendaciones	91
9.	Referencias bibliográficas.....	94
10.	Anexos	97
10.1.	Anexo A. Registro de cambios identificados durante los sprints.....	97
10.2.	Anexo B. Entrevista dirigida y evidencia de aplicación	98
10.3.	Anexo C. Ficha de observación directa.....	101
10.4.	Anexo D. Product Backlog consolidado	104

10.5.	Anexo E. Sprint Backlog y revisiones.....	108
10.6.	Anexo F. Matriz de trazabilidad.....	118
10.7.	Anexo G. Evidencias de pruebas funcionales	119
10.8.	Anexo H. Evidencias de pruebas de la API REST.....	120
10.9.	Anexo I. Instrumento y resultados System Usability Scale	121
10.10.	Anexo J. Medición de tiempos y errores	124
10.11.	Anexo K. Manual breve de usuario	125
10.12.	Anexo L. Acta de validación del propietario	125
10.13.		126
10.14.	Anexo M. Diagramas complementarios	126
10.15.	Anexo N. Acta de aceptación de requerimientos por perfil de usuario	126

Índice de figuras

Figura 1 Diagrama metodológico de la investigación	20
Figura 1 Proceso actual de venta (embarque).....	33
Figura 2 Proceso propuesto de gestión del embarque	34
Figura 3 Proceso actual de inventario.....	35
Figura 5 Proceso propuesto de gestión del inventario	36
Figura 5 Arquitectura general del sistema en tres capas	38
Figura 6 Diagrama de componentes del sistema	38
Figura 7 Diagrama de despliegue del sistema	39
Figura 8 Modelo entidad-relación del sistema	41
Figura 10 Diagrama de casos de uso del sistema	62
Figura 10 Diagrama de estados del embarque	67
Figura 12 Diagrama de secuencia de la finalización y cierre del embarque	68
Figura 12 Diagrama de actividades de la planilla semanal	70
Figura 13 Mapa de navegación de la aplicación por roles	72
Figura 14 Diagrama de secuencia de la sincronización sin conexión	73
Figura 15 Evidencia del análisis y levantamiento de requisitos del Sprint 1	109
Figura 16 Evidencia de autenticación e inventario correspondiente al Sprint 2	111
Figura 17 Evidencia del módulo de embarques y cierre financiero del Sprint 3.....	113
Figura 19 Evidencia del módulo de jornales y planilla semanal del Sprint 3	114
Figura 19 Evidencia de las pruebas de acceso por roles del Sprint 4.....	116
Figura 20 Evidencia del registro sin conexión y sincronización del Sprint 4	116
Figura 21 Evidencia de la aplicación del cuestionario SUS en el Sprint 4	117

Índice de ecuaciones

Ecuación 1 Cálculo del Ratio del Guineo.....	9
Ecuación 2 Cálculo SUS	79
Ecuación 3 Cálculo de la reducción porcentual del tiempo	81

Índice de tablas

Tabla 1 Flujo metodológico de la investigación.....	20
Tabla 2 Distribución final de actividades ejecutadas por sprint.....	26
Tabla 3 Relación entre el alcance inicial y las funcionalidades incorporadas.....	27
Tabla 4 Tabla t_usuario.....	42
Tabla 5 Tabla t_dueno.....	42
Tabla 6 Tabla t_finca.....	42
Tabla 7 Tabla t_comprador.....	43
Tabla 8 Tabla t_trabajador.....	43
Tabla 9 Tabla t_tarea.....	43
Tabla 10 Tabla t_evaluador.....	43
Tabla 11 Tabla t_categoria.....	43
Tabla 12 Tabla t_proveedor.....	44
Tabla 13 Tabla t_producto.....	44
Tabla 14 Tabla t_movimiento_inventario.....	45
Tabla 15 Tabla t_cinta_semana.....	45
Tabla 16 Tabla t_embarque.....	45
Tabla 17 Tabla t_embarque_detalle.....	46
Tabla 18 Tabla t_cuadrilla.....	46
Tabla 19 Tabla t_cierre_embarque.....	46
Tabla 20 Tabla t_cierre_embarque_detalle.....	47
Tabla 21 Tabla t_cierre_embarque_cuadrilla.....	48
Tabla 22 Tabla t_gasto.....	48
Tabla 23 Tabla t_jornal.....	49
Tabla 24 Tabla t_jornal_dueno.....	49
Tabla 25 Tabla t_pago_jornal_semanal.....	49
Tabla 26 Tabla t_facturador_finca.....	50
Tabla 27 Tabla t_mayordomo_dueno.....	50
Tabla 28 Tabla t_auditoria.....	50
Tabla 29 Tabla t_precio_banano_historico.....	51
Tabla 30 Requisitos funcionales del sistema.....	54
Tabla 31 Requisitos no funcionales del sistema.....	58
Tabla 32 Historias de usuario.....	59
Tabla 33 Resultados de las pruebas funcionales.....	76
Tabla 34 Criterios verificados en las pruebas de la API REST.....	77
Tabla 35 Resumen de la evaluación de usabilidad.....	79
Tabla 36 Estadísticos descriptivos del puntaje SUS (n = 5).....	80
Tabla 37 Comparación de tiempos de registro.....	81
Tabla 38 Tiempos absolutos de ejecución por operación.....	82
Tabla 39 Estadísticos descriptivos de la reducción porcentual del tiempo.....	82
Tabla 40 Proyección del ahorro de horas-hombre según la frecuencia operativa declarada.....	84
Tabla 41 Comparación de errores de registro.....	85

Tabla 42 Categorización de los errores según el mecanismo de control del prototipo	85
Tabla 43 Registro de cambios del alcance durante los sprints	97
Tabla 44 Resumen de la entrevista dirigida.....	100
Tabla 45 Registro de tiempos y observaciones de las operaciones evaluadas	101
Tabla 46 Estimación de Story Points y alcance del MVP	104
Tabla 47 Product Backlog consolidado	107
Tabla 48 Resumen del Sprint 1	108
Tabla 49 Resumen del Sprint 2	110
Tabla 50 Resumen del Sprint 3	112
Tabla 51 Resumen del Sprint 4	115
Tabla 52 Matriz de trazabilidad entre requisitos, historias y pruebas	118
Tabla 53 Matriz de evidencias de pruebas funcionales	119
Tabla 54 Registro de evidencias de Postman o Swagger	121
Tabla 55 Matriz anonimizada de respuestas SUS	122
Tabla 56 Hoja consolidada para el cálculo de reducción de tiempos	124
Tabla 57 Hoja consolidada para la comparación de errores	124

Introducción

1.1. Antecedentes

El sector bananero constituye uno de los pilares fundamentales de la economía ecuatoriana. En 2023, las exportaciones del sector alcanzaron USD 3,79 miles de millones FOB [1], y constituye una fuente relevante de empleo directo e indirecto, además de una actividad estratégica para la economía ecuatoriana [2]. Ecuador se posiciona así como uno de los principales exportadores de banano a nivel mundial, con una producción concentrada en provincias de la costa y el austro, entre ellas Cañar.

A pesar de esta relevancia económica, la incorporación de tecnologías digitales en la gestión administrativa de las empresas productoras (particularmente las pequeñas) continúa siendo limitada. En el sector agrícola ecuatoriano, únicamente el 10 % de las PYMES ha adoptado tecnologías digitales, una cifra notablemente inferior al 50 % que registra el sector servicios [3]. Esta brecha digital restringe la capacidad de las pequeñas haciendas para mejorar su productividad, optimizar sus procesos y competir en un entorno cada vez más exigente.

La gestión administrativa de una hacienda bananera involucra procesos críticos como el control de la producción, el registro de ventas, la administración del inventario de productos agrícolas y el control de la mano de obra. Cuando estos procesos se realizan de forma manual, mediante cuadernos físicos y recibos, la información tiende a dispersarse, dificultando su consulta y verificación, e incrementando la probabilidad de errores. Este panorama es especialmente complejo en haciendas operadas bajo esquemas de propiedad compartida, donde varios dueños comparten una misma finca y requieren conocer con precisión su producción individual, sus ingresos y sus gastos.

La masificación de los dispositivos móviles refuerza la pertinencia de una solución de este tipo. Android, el sistema operativo móvil predominante a nivel mundial con cerca del 72 % del mercado [4], está presente en dispositivos de gama media y baja ampliamente difundidos en zonas rurales, lo que convierte a una aplicación móvil multiplataforma en una vía realista para llevar la digitalización al campo. Sin embargo, la adopción de estas tecnologías en el sector agrícola sigue siendo incipiente, lo que abre una oportunidad para iniciativas como la del presente proyecto.

1.2. Planteamiento del problema

1.2.1 Situación actual

La Hacienda Bananera San Diego, ubicada en La Troncal, provincia de Cañar, opera bajo un esquema de propiedad compartida entre siete dueños, cada uno propietario de una o varias fincas con extensiones distintas. La venta de banano se realiza mediante embarques que ocurren una o dos veces por semana, en los que la producción de todos los dueños se procesa de manera conjunta y se vende a un comprador, a un precio por caja que varía según el mercado.

Actualmente, el registro de ventas, el control del inventario de productos de bodega

(fertilizantes como la urea, químicos de fumigación y materiales como fundas y protectores) y el control del personal que labora en la finca se realiza de manera manual, en cuadernos físicos y recibos. El inventario se verifica de forma visual y se anota a mano, mientras que la producción de cada dueño, la liquidación del embarque y los pagos a los trabajadores se calculan manualmente al cierre de cada semana. A esta situación se suma que la hacienda se encuentra en una zona rural donde el acceso a internet es limitado o inexistente, lo que dificulta el uso de soluciones tecnológicas que dependan de una conexión permanente.

1.2.2 Problemas detectados

A partir de la entrevista dirigida y de la observación directa se comprobó que la información se encuentra dispersa en múltiples cuadernos y documentos, lo que dificulta su consulta y actualización oportuna. El cálculo manual de la producción, los ingresos, los gastos del embarque y los jornales resulta lento y propenso a inconsistencias, duplicidad de registros y pérdida de información. Además, la ausencia de un control automatizado del inventario impide conocer en tiempo real la disponibilidad de productos, y menos aún diferenciar a qué dueño pertenece cada existencia.

1.2.3 Consecuencias

Como consecuencia, los procesos administrativos se vuelven ineficientes y poco confiables. Los errores en el cálculo de la producción por dueño pueden derivar en pagos incorrectos, mientras que la falta de control del inventario ocasiona desabastecimiento o sobreacumulación de productos. La imprecisión en el registro de la mano de obra dificulta determinar correctamente cuánto se debe pagar a cada

trabajador y cuánto gasto asume cada dueño. En conjunto, estas deficiencias afectan la transparencia, la organización y la toma de decisiones dentro de la hacienda.

1.3. Formulación del problema

A partir de la situación descrita, se plantea la siguiente pregunta de investigación:

¿Cómo puede un prototipo de sistema ERP móvil mejorar la eficiencia en la gestión de facturación e inventario en la Hacienda Bananera San Diego, ubicada en La Troncal, Cañar?

1.4. Objetivos

1.4.1 Objetivo general

Desarrollar un prototipo de sistema ERP móvil que permita mejorar la gestión de facturación e inventario en la Hacienda Bananera San Diego, ubicada en La Troncal, Cañar, reduciendo en al menos un 70 % el tiempo de registro de operaciones y reduciendo significativamente los errores derivados del registro manual, en un plazo de cuatro sprints de desarrollo bajo metodología Scrum.

Las dos metas cuantitativas de este objetivo tienen origen distinto y conviene señalarlo desde aquí. El umbral de 70 % no procede de la literatura ni de una norma sectorial: es un criterio de aceptación acordado con el propietario durante el diagnóstico, a partir de los cerca de sesenta minutos que las tres operaciones administrativas evaluadas consumían por ciclo de embarque. La referencia de 68 puntos del cuestionario SUS, en cambio, es un valor de comparación reconocido en la literatura de usabilidad. El apartado 3.6.2 detalla el razonamiento del primer umbral y las condiciones experimentales bajo las que se midió su cumplimiento.

1.4.2 Objetivos específicos

1. Analizar los procesos actuales de facturación e inventario en la Hacienda Bananera San Diego mediante observación directa y entrevistas, identificando al menos 5 requerimientos funcionales críticos en un plazo máximo de 2 semanas.
2. Diseñar la arquitectura del sistema ERP móvil, definiendo los módulos de facturación e inventario, el modelo de datos y la API REST, documentando los requerimientos funcionales y no funcionales antes del inicio del segundo sprint.

3. Desarrollar un prototipo funcional (MVP) del sistema ERP móvil utilizando Expo/React Native en el frontend y Spring Boot con PostgreSQL en el backend, completando el 100 % de las funcionalidades básicas en 4 sprints de dos semanas bajo metodología Scrum.
4. Evaluar la usabilidad del prototipo mediante el cuestionario SUS aplicado a un mínimo de 5 usuarios (dueños y empleados), obteniendo una puntuación mínima de 68 puntos sobre 100, equivalente a un nivel de usabilidad aceptable.

1.5. Justificación

Desde una perspectiva práctica, la digitalización de los procesos administrativos de la Hacienda Bananera San Diego permite sustituir los registros dispersos por una solución tecnológica que centraliza la información, automatiza cálculos y aplica validaciones de negocio. El sistema fortalece el control de la producción por dueño, la liquidación de cada embarque, el inventario de productos y el registro de la mano de obra, mejorando la trazabilidad y la disponibilidad de la información [5].

En el ámbito teórico, el proyecto aporta al área de ingeniería de software mediante el desarrollo de un prototipo basado en tecnologías actuales como Expo/React Native, Spring Boot y PostgreSQL, integrando conceptos de sistemas ERP, desarrollo de aplicaciones móviles multiplataforma y evaluación de usabilidad. Asimismo, permite aplicar la metodología ágil Scrum y técnicas de levantamiento de requerimientos, diseño de sistemas y validación con usuarios reales.

Desde el punto de vista social, el trabajo promueve la inclusión tecnológica y la modernización de prácticas tradicionales en el sector agrícola. La arquitectura y el enfoque de desarrollo adoptados podrían adaptarse a otras pequeñas haciendas bananeras de la región que operen bajo esquemas similares de propiedad compartida, contribuyendo a su competitividad y sostenibilidad.

1.6. Alcance y limitaciones

El presente proyecto se limita al diseño y desarrollo de un prototipo funcional (MVP) de un sistema ERP móvil para la gestión de facturación e inventario en la Hacienda Bananera San Diego. El sistema incluye la gestión de usuarios con seis roles diferenciados (administrador, mayordomo, facturador, bodeguero, capitán y dueño), el registro de los dueños y de sus fincas con ubicación geográfica

seleccionada sobre un mapa interactivo, el registro de embarques con producción diferenciada por dueño y por finca (cajas y racimos cortados), la identificación de la semana y el color de cinta del embarque, la gestión del estado del proceso de cada embarque y su finalización con conciliación de la producción real, el cierre financiero del embarque con reparto de costos proporcional a las cajas aportadas, el registro del evaluador de la fruta y de su pago, la reliquidación posterior del embarque, el control del inventario de productos de bodega clasificados por categoría (fertilizantes, químicos y materiales) con sus proveedores y con existencias diferenciadas por dueño, el registro de gastos de cuatro tipos (de embarque, de finca, compra de producto y servicio), el registro de la cuadrilla de embarque y del personal semanal de la finca con una planilla de pago semanal, la emisión de comprobantes en formato PDF que pueden compartirse con los dueños mediante mensajería (WhatsApp), la trazabilidad de las operaciones por usuario mediante auditoría, recordatorios locales para la verificación de jornales, y reportes de ventas por dueño, gastos, mano de obra, inventario y productividad.

Técnicamente, se trata de una aplicación multiplataforma para Android e iOS desarrollada con Expo/React Native, con un backend en Java sobre Spring Boot y base de datos PostgreSQL, comunicados mediante una API REST. El sistema fue diseñado para un número reducido de usuarios concurrentes (aproximadamente 4 a 10), aunque no se realizaron pruebas formales de carga o concurrencia. Aunque contempla compatibilidad con Android e iOS por su desarrollo multiplataforma, la validación experimental de la versión evaluada se realizó en dispositivos Android. Dado que la hacienda se ubica en una zona rural con conectividad a internet limitada o nula, el sistema contempla un modo de operación sin conexión que permite registrar operaciones críticas (como el embarque) almacenándolas localmente en el dispositivo mediante una base de datos SQLite, para posteriormente sincronizarlas de forma automática con el servidor cuando se restablezca la conexión a internet.

Como limitaciones, el prototipo no contempla la implementación de un ERP completo, la facturación electrónica homologada ante el SRI (la cual se aborda únicamente como fundamento teórico y posible trabajo futuro; los comprobantes que emite el sistema son documentos internos de respaldo), módulos financieros complejos de liquidación contable entre socios, la trazabilidad de enfunde por trabajador y lote de cinta, la gestión multisucursal ni el despliegue en un entorno de producción. Asimismo, la funcionalidad de operación sin conexión se implementa de forma básica, orientada al almacenamiento local y la sincronización posterior, sin contemplar mecanismos avanzados de resolución automática de conflictos de datos.

Marco Teórico

1.7. Sistemas de Información Empresarial

Un sistema de información empresarial es un conjunto organizado de componentes

(personas, procesos, datos y tecnología) que recolecta, procesa, almacena y distribuye información para apoyar la toma de decisiones y el control dentro de una organización [6]. Su finalidad es transformar datos operativos en información útil que permita coordinar las actividades de los distintos departamentos y mejorar la eficiencia de los procesos de negocio.

En las pequeñas empresas, la adopción de sistemas de información representa un paso fundamental hacia la modernización de sus operaciones. La sustitución de los registros manuales por sistemas digitales reduce la redundancia de datos, mejora la trazabilidad de las operaciones y facilita el acceso oportuno a la información, aspectos especialmente relevantes en sectores como el agrícola, donde la gestión tradicional aún predomina [3].

1.8. Sistemas ERP (Enterprise Resource Planning)

Los sistemas ERP son plataformas integradas de gestión empresarial que centralizan los procesos de una organización en un único entorno de información, facilitando la toma de decisiones y mejorando la eficiencia operativa [7]. Surgieron en la década de 1990 como evolución de los sistemas MRP (Manufacturing Resource Planning) y se han expandido progresivamente hacia sectores no manufactureros, incluyendo el agrícola [7].

1.8.1 Características de los sistemas ERP

Entre las características principales de un sistema ERP se encuentran la integración de los procesos de negocio en una base de datos común, la modularidad que permite incorporar funcionalidades según las necesidades de la organización, la consistencia de la información entre departamentos y la capacidad de generar reportes consolidados en tiempo real [7]. Esta integración elimina la duplicidad de datos y garantiza que la información sea coherente a lo largo de toda la organización.

1.8.2 Arquitectura de un sistema ERP

Desde el punto de vista arquitectónico, los sistemas ERP modernos suelen estructurarse en capas: una capa de presentación que constituye la interfaz con el usuario, una capa de lógica de negocio donde se ejecutan las reglas y procesos de la organización, y una capa de datos que gestiona el almacenamiento y la persistencia de la información [7]. Esta separación facilita el mantenimiento, la escalabilidad y la posibilidad de acceder al sistema desde distintos tipos de cliente, incluyendo dispositivos móviles.

1.8.3 Módulos de un sistema ERP

Un sistema ERP se compone de módulos independientes pero interconectados, como compras, ventas, inventario, contabilidad, producción y recursos humanos, que comparten una base de datos común [6]. En el contexto del presente proyecto, el prototipo se enfoca en módulos específicos adaptados a la realidad de una hacienda bananera: gestión de inventario de productos, registro de ventas mediante embarques con su cierre financiero, control de gastos, administración de la mano de obra y emisión de comprobantes.

1.8.4 ERP móviles

Los ERP móviles representan una evolución del paradigma ERP tradicional. Al combinar las capacidades de integración de un ERP con la portabilidad de las aplicaciones móviles, permiten acceder a la información de gestión desde cualquier ubicación, lo que resulta especialmente valioso en entornos de trabajo distribuidos como las haciendas agrícolas [7]. En estos contextos, donde el personal opera en el campo y no frente a un computador de escritorio, la movilidad se convierte en un factor determinante para la adopción efectiva de la tecnología.

1.8.5 Gestión de Inventarios

La gestión de inventario comprende el conjunto de procesos de control, supervisión y administración de la entrada, almacenamiento y salida de productos e insumos en una organización. Una gestión eficiente del inventario reduce los errores operativos y mejora la planificación de recursos [8]. En el sector agroindustrial, este proceso involucra tanto productos terminados como insumos de consumo frecuente, con ciclos de rotación diferentes.

1.8.6 Kardex y control de existencias

El Kardex es un registro detallado y cronológico de los movimientos de cada producto o insumo en el inventario, que permite conocer en todo momento las existencias disponibles a partir del registro de entradas y salidas [8]. Su digitalización permite automatizar el cálculo de los saldos y reducir significativamente los errores propios del registro manual en cuadernos. En el presente sistema, este enfoque se lleva un paso más allá: el saldo de cada producto no solo se calcula a partir de sus movimientos, sino que se diferencia por dueño, dado que en una finca de propiedad compartida cada socio compra y consume sus propios insumos.

1.8.7 Entradas y salidas

Las entradas corresponden al ingreso de productos al inventario, generalmente producto de compras, mientras que las salidas representan el consumo o uso de dichos productos en las labores de la finca, como la fumigación o el enfunde. El registro sistemático de ambos movimientos es esencial para mantener un control preciso de las existencias [8].

1.8.8 Stock mínimo

El stock mínimo es el nivel de existencias por debajo del cual es necesario realizar una nueva compra para evitar el desabastecimiento. La definición y el monitoreo automatizado de este umbral permiten anticipar las necesidades de reposición y evitar tanto la escasez como la sobreacumulación de productos [8].

1.8.9 Categorías y proveedores

La clasificación de los productos en categorías (como fertilizantes, químicos y materiales) facilita la organización del inventario y la generación de reportes segmentados, permitiendo un control diferenciado según la naturaleza de cada producto [8]. Asimismo, el registro de proveedores asociado a cada entrada de inventario aporta trazabilidad sobre el origen de los productos y respalda la planificación de compras. En el caso de productos perecederos, como ciertos químicos agrícolas, el control de la fecha de caducidad por lote resulta indispensable para evitar el uso de productos vencidos.

1.9. Productividad y Calidad en la Producción Bananera

La medición de la productividad en una finca bananera se apoya en indicadores que relacionan la fruta cosechada con la fruta efectivamente empacada. En la operación de la Hacienda Bananera San Diego se emplea como indicador el ratio de conversión, calculado mediante la relación entre el número de cajas elaboradas y el número de racimos cortados. Esta fórmula corresponde a la práctica operativa identificada durante la entrevista dirigida y la observación del embarque; un valor más alto indica un mejor aprovechamiento de la fruta por racimo. La importancia del control del peso, del ratio y de la merma en la producción bananera se encuentra respaldada en la literatura técnica del sector [9]. Este indicador se expresa mediante la Ecuación 1.

Ecuación 1 Cálculo del Ratio del Guineo

$$\text{Ratio de conversión} = \frac{\text{Número de cajas elaboradas}}{\text{Número de racimos cortados}}$$

De manera complementaria, la merma representa la cantidad de fruta descartada por no cumplir los estándares de exportación y puede analizarse mediante la diferencia en peso entre la fruta cosechada y la fruta finalmente empacada. El seguimiento de ambos indicadores permite evaluar la eficiencia de la cosecha y la calidad de la producción de cada dueño. En la práctica de la hacienda estudiada, la calidad de la fruta es verificada el día del embarque por un evaluador, cuya remuneración forma parte de los costos que se reparten entre los dueños.

1.9.1 Calendario de enfunde y cintas de color

El enfunde es la labor mediante la cual el racimo recién emitido se cubre con una funda plástica para protegerlo, y a esa funda se le ata una cinta de color que identifica la semana en que fue enfundado. De acuerdo con el calendario de enfunde, cada semana se asigna un color distinto, lo que permite reconocer la edad del racimo en el campo y decidir el momento adecuado de la cosecha, que normalmente ocurre alrededor de las trece semanas de edad [10]. Dado que la fruta se cosecha en ese rango de tiempo, el conjunto de colores se repite en un ciclo de aproximadamente trece semanas. En el presente sistema, esta práctica se modela registrando, para cada embarque, la semana del año y el color de cinta correspondiente, ya sea de forma manual o asignado automáticamente a partir de un calendario de cintas

configurable; adicionalmente, cada detalle de producción admite cintas adicionales cuando en un mismo corte se cosecha fruta de más de una semana de enfunde.

1.10. Facturación Electrónica en Ecuador

La facturación es el proceso administrativo mediante el cual se registran las transacciones comerciales y se emiten documentos que detallan las ventas realizadas, permitiendo el control de ingresos y la trazabilidad financiera de la organización [11]. En el contexto del presente proyecto, esta sección se presenta como fundamento conceptual y posible línea de trabajo futuro, dado que el prototipo no implementa facturación electrónica homologada: los comprobantes que genera son documentos internos de respaldo.

1.10.1 El SRI y la facturación electrónica

En Ecuador, el Servicio de Rentas Internas (SRI) es la entidad encargada de regular la emisión de comprobantes electrónicos. La facturación electrónica consiste en la generación de documentos tributarios en formato digital con validez legal, que deben ser autorizados por el SRI antes de su entrega al comprador [11].

1.10.2 Formato XML y firma digital

Los comprobantes electrónicos se estructuran en formato XML, un estándar que permite representar la información tributaria de manera estructurada y legible por sistemas informáticos. Para garantizar la autenticidad e integridad del documento, este debe contar con una firma digital emitida por una entidad certificadora autorizada [11].

1.10.3 Ambientes de pruebas y producción

El esquema de facturación electrónica del SRI contempla dos ambientes: uno de pruebas, destinado a la validación y desarrollo de las integraciones, y uno de producción, donde los comprobantes adquieren validez tributaria oficial. La integración con estos ambientes constituye una posible extensión futura del presente sistema.

1.11. Arquitectura de Software

1.11.1 Arquitectura Cliente-Servidor

La arquitectura cliente-servidor es un modelo en el que las tareas se distribuyen entre proveedores de recursos o servicios, denominados servidores, y solicitantes de dichos servicios, denominados clientes. El cliente (en este caso la aplicación móvil) envía solicitudes al servidor, que las procesa y devuelve una respuesta [27]. Este modelo permite centralizar la lógica de negocio y los datos en el servidor, mientras múltiples clientes acceden a ellos de manera concurrente.

1.11.2 Arquitectura REST API

REST (Representational State Transfer) es un estilo de arquitectura para el diseño de servicios web que utiliza el protocolo HTTP y sus métodos estándar (GET, POST, PUT, DELETE) para la manipulación de recursos [13]. Una API REST expone los recursos del sistema mediante URLs y permite la comunicación entre el cliente móvil y el servidor de forma desacoplada, ligera y escalable, lo que la hace idónea para aplicaciones móviles.

1.11.3 Patrón Modelo-Vista-Controlador (MVC)

El patrón Modelo-Vista-Controlador separa una aplicación en tres componentes: el Modelo, que representa los datos y la lógica de negocio; la Vista, que constituye la interfaz de usuario; y el Controlador, que gestiona las interacciones y coordina el flujo entre el Modelo y la Vista [13]. Esta separación de responsabilidades facilita el mantenimiento, la escalabilidad y la organización del código.

1.11.4 Aplicaciones con sincronización offline

En contextos rurales con conectividad limitada, las aplicaciones móviles deben contemplar mecanismos de operación sin conexión. El enfoque de almacenamiento local consiste en guardar temporalmente los datos en el dispositivo cuando no hay acceso a internet, para posteriormente sincronizarlos con el servidor una vez restablecida la conexión [28]. Este patrón resulta indispensable en escenarios como el de la Hacienda San Diego, donde las operaciones críticas se realizan en zonas sin cobertura de red.

1.11.5 Arquitectura offline-first y sincronización diferida

El enfoque offline-first invierte el supuesto habitual de las aplicaciones conectadas: en lugar de tratar la ausencia de red como un estado de error, la asume como una condición normal de operación. La aplicación escribe siempre contra un almacén local, que actúa como fuente de verdad para la interfaz, y delega en un proceso independiente la propagación de esos cambios hacia el servidor cuando la conectividad lo permite. El usuario percibe una respuesta inmediata y no queda bloqueado por la latencia o la caída del enlace.

La sincronización diferida se apoya normalmente en una cola de operaciones pendientes, también denominada bandeja de salida, en la que cada transacción se registra con un identificador propio y una marca de tiempo. Cuando la conexión se restablece, la cola se envía en orden y el servidor responde confirmando o rechazando cada elemento. El uso de identificadores generados en el cliente y de operaciones idempotentes evita que un reenvío por fallo intermitente produzca registros duplicados.

El problema central de este esquema es la resolución de conflictos, que aparece cuando el mismo registro fue modificado en el dispositivo y en el servidor durante el período de desconexión. Las estrategias más difundidas son la prevalencia de la última escritura (last-write-wins), sencilla pero susceptible de pérdida silenciosa de datos; la resolución por reglas de dominio, que decide en función del significado del campo modificado; y la resolución asistida, que presenta el conflicto al usuario. La elección determina el modelo de consistencia del sistema, que en estas arquitecturas es de tipo eventual: los datos convergen a un estado común, pero no de forma instantánea.

1.11.6 Clean Architecture y diseño orientado al dominio

La Clean Architecture, formulada por Robert C. Martin [30], organiza el software en capas concéntricas gobernadas por la regla de dependencia: el código de las capas externas puede depender del de las internas, nunca a la inversa. En el centro se ubican las entidades y los casos de uso, que expresan las reglas del negocio; en el exterior, los adaptadores de interfaz y los detalles de infraestructura (marco de trabajo, base de datos, interfaz gráfica), que se consideran decisiones sustituibles. El propósito es que la lógica de negocio permanezca aislada de la tecnología que la soporta y pueda probarse sin levantar el entorno completo.

El diseño orientado al dominio (Domain-Driven Design), propuesto por Eric Evans [31], complementa este planteamiento aportando el vocabulario con el que se modela ese núcleo: entidades, objetos de valor, agregados, repositorios y servicios de dominio, articulados en torno a un lenguaje ubicuo compartido entre desarrolladores y expertos del negocio. En un sistema de gestión agroindustrial, conceptos como embarque, ratio de conversión por dueño o liquidación de cuadrilla constituyen precisamente ese

lenguaje, y su traslado fiel al modelo de datos y a los servicios condiciona la utilidad del sistema.

Ambos enfoques resultan pertinentes para este trabajo porque las reglas más sensibles del prototipo (el reparto proporcional de gastos, el cálculo del ratio por dueño y la liquidación semanal) son reglas de dominio y no detalles de presentación. Mantenerlas concentradas en la capa de servicios del backend, y no dispersas en los controladores o en la aplicación móvil, es la condición que permite modificarlas o someterlas a prueba sin alterar el resto del sistema.

1.11.7 Gestión del estado en aplicaciones móviles

La gestión del estado se refiere a cómo una aplicación mantiene, comparte y actualiza la información que sus distintas pantallas necesitan. Conviene distinguir el estado local de un componente, el estado global de la aplicación (sesión del usuario, rol activo, preferencias) y el estado del servidor, que es una copia en memoria de datos remotos y por tanto puede quedar obsoleta.

En el ecosistema de React Native coexisten varias aproximaciones. El estado local se resuelve con los propios ganchos del marco de trabajo; el estado global, mediante el contexto nativo o bibliotecas especializadas como Redux o Zustand; y el estado del servidor, mediante bibliotecas de caché y revalidación que gestionan el ciclo de carga, error y actualización. La distinción es especialmente relevante en una aplicación offline-first, donde el estado del servidor deja de ser una simple caché y pasa a convivir con datos locales aún no sincronizados, lo que obliga a representar de forma explícita la condición de cada registro ante el usuario.

1.12. Tecnologías Utilizadas

1.12.1 Android y el ecosistema móvil

Android es el sistema operativo móvil con mayor participación en el mercado mundial. Hacia finales de 2024 concentraba alrededor del 72 % del mercado global de sistemas operativos móviles [4]. Su amplia disponibilidad en dispositivos de distintas gamas lo convierte en una plataforma relevante para soluciones dirigidas a contextos rurales. Por esta razón, la compatibilidad con Android constituye una prioridad del proyecto, sin excluir el soporte para iOS proporcionado por el desarrollo multiplataforma [14].

1.12.2 Desarrollo nativo frente a multiplataforma

Existen dos enfoques principales para construir aplicaciones móviles. El desarrollo nativo emplea tecnologías específicas de cada plataforma y ofrece acceso directo a sus capacidades, pero obliga a mantener implementaciones separadas. El desarrollo multiplataforma utiliza una base de código compartida para atender más de un

sistema operativo, reduciendo el esfuerzo de mantenimiento [15]. Debido a que el proyecto fue desarrollado por un solo estudiante y debía atender principalmente dispositivos Android sin descartar iOS, se seleccionó el enfoque multiplataforma.

1.12.3 React Native y Expo

React Native permite construir aplicaciones móviles con JavaScript o TypeScript y componentes de React, compartiendo entre Android e iOS buena parte de la lógica y de la interfaz, lo que reduce el esfuerzo de mantenimiento frente a dos desarrollos separados [15]. Expo complementa este entorno con herramientas y APIs para ejecutar, probar y distribuir la aplicación, además de facilitar el uso de SQLite, geolocalización, notificaciones, impresión y compartición de documentos [16]. Estas características justifican su elección para el frontend móvil del prototipo.

1.12.4 Spring Boot

Spring Boot es un marco de trabajo sobre Java orientado a la construcción de backends con arquitectura de API REST [13]. Forma parte del ecosistema Spring y ofrece configuración automática, gestión de dependencias integrada y soporte nativo para la persistencia de datos mediante Spring Data JPA. Se eligió esta tecnología por su madurez técnica, su amplia comunidad y su consolidación en el desarrollo de backends empresariales, factores que favorecen la disponibilidad de soporte y documentación [17]. Su modelo de desarrollo por capas, además, se alinea de forma natural con el patrón Modelo-Vista-Controlador adoptado en el proyecto.

1.12.5 PostgreSQL

PostgreSQL es un gestor de bases de datos relacional de código abierto con integridad referencial y soporte de transacciones ACID (atomicidad, consistencia, aislamiento y durabilidad) [18]. Surgido del proyecto Postgres de la Universidad de California en Berkeley [19], se seleccionó para este proyecto porque sus garantías de consistencia resultan pertinentes en un sistema de facturación e inventario, donde varios usuarios pueden registrar operaciones de forma concurrente, y porque su compatibilidad nativa con Spring Boot mediante Spring Data JPA reduce la complejidad de integración. Conviene precisar que estas garantías corresponden a capacidades inherentes del gestor, documentadas por el proyecto y ampliamente verificadas por su comunidad, y no a propiedades comprobadas experimentalmente en el presente trabajo: la tesis no ejecutó pruebas formales de concurrencia ni de carga sobre la base de datos, tal como se reconoce en el apartado de limitaciones. Al ser de código abierto, además, no implica costos de licenciamiento para la hacienda.

1.12.6 JWT y seguridad

JSON Web Token (JWT) es un estándar abierto para la transmisión segura de información entre partes en forma de objeto JSON firmado digitalmente. En el contexto de las API REST, JWT se utiliza para la autenticación y autorización de usuarios: tras iniciar sesión, el servidor emite un token que el cliente incluye en cada solicitud posterior, permitiendo verificar su identidad y sus permisos sin necesidad de mantener el estado de la sesión en el servidor [13]. Este mecanismo es fundamental para implementar el control de acceso basado en roles del presente sistema, que además protege las contraseñas mediante el algoritmo de hash BCrypt.

1.12.7 Git y control de versiones

Git es un sistema de control de versiones distribuido que permite registrar los cambios en el código fuente a lo largo del tiempo, facilitando el trabajo organizado y la recuperación de versiones anteriores [29]. Junto con plataformas como GitHub, posibilita el respaldo del código en la nube y la gestión ordenada del desarrollo a lo largo de los sprints. En el presente trabajo, el código fuente del prototipo se organiza en un repositorio privado alojado en GitLab bajo un grupo institucional que agrupa dos proyectos: `sanDiego_Back`, correspondiente a la API REST desarrollada con Spring Boot y PostgreSQL, y `sanDiego_Front`, correspondiente a la aplicación móvil desarrollada con React Native y Expo. El acceso al repositorio se pondrá a disposición del tribunal para su verificación durante la defensa.

1.12.8 Seguridad en aplicaciones móviles y OWASP Mobile Top 10

El proyecto OWASP mantiene el listado Mobile Top 10 [32], que recoge las categorías de riesgo más frecuentes en aplicaciones móviles y constituye una referencia habitual para orientar las decisiones de seguridad en este tipo de desarrollos. Entre las categorías más pertinentes para un cliente móvil que consume una API REST figuran el uso inadecuado de credenciales, la autenticación y la autorización insuficientes, el almacenamiento inseguro de datos en el dispositivo, la comunicación insegura y la protección deficiente de los datos en tránsito y en reposo.

De ese listado se desprenden prácticas concretas que enmarcan el diseño de seguridad del prototipo: verificar la autorización en el servidor y no en la interfaz, de modo que ocultar una opción de menú nunca sustituya al control de acceso en el endpoint; almacenar los secretos del cliente en el almacén cifrado que provee el sistema operativo y no en almacenamiento de texto plano; cifrar el canal de comunicación mediante TLS; limitar el tiempo de validez de las credenciales de sesión; y registrar los eventos de autenticación para permitir la auditoría posterior.

1.12.9 Ciclo de vida de los tokens de sesión

La seguridad de un esquema basado en JWT no depende únicamente de la firma del token, sino de la gestión de su ciclo de vida completo. Cuatro decisiones lo definen. La primera es el tiempo de expiración: un token de acceso de vida corta reduce la ventana de aprovechamiento en caso de robo, pero obliga a reautenticar con mayor frecuencia. La segunda es el mecanismo de renovación, que habitualmente se resuelve con un token de refresco de vida más larga y de un solo uso, lo que permite mantener sesiones prolongadas sin extender la validez del token de acceso.

La tercera es la estrategia de invalidación. Por su naturaleza autocontenida, un JWT firmado sigue siendo válido hasta su expiración aunque el usuario cierre sesión o sea dado de baja; revocarlo antes de tiempo exige mantener del lado del servidor una lista de tokens revocados, un registro de tokens de refresco activos o un identificador de versión de credenciales que se compare en cada petición. La cuarta es el método de almacenamiento en el dispositivo: los almacenes cifrados que ofrecen las plataformas móviles resultan preferibles al almacenamiento de clave-valor sin cifrar, que queda expuesto en dispositivos comprometidos.

Documentar estas cuatro decisiones es lo que permite evaluar la arquitectura de seguridad de un sistema y planificar su evolución hacia un entorno de producción. En el apartado 5.2.3 se describe la configuración adoptada en el prototipo y las decisiones que quedaron pendientes para una futura puesta en producción.

1.12.10 Criterios de selección y compromisos de la pila tecnológica

La elección de una pila tecnológica supone aceptar compromisos, y explicitarlos es parte de la justificación de diseño. Se aplicaron cuatro criterios: la adecuación al requisito de operación sin conexión, el rendimiento esperable en dispositivos de gama media, el costo de licenciamiento e infraestructura, y la disponibilidad de documentación y soporte para un equipo de un solo desarrollador.

El compromiso más relevante corresponde al frontend. React Native ejecuta la lógica de la aplicación en un entorno de JavaScript separado del hilo de interfaz, de modo que su rendimiento en animaciones complejas y en listas muy extensas es inferior al del desarrollo nativo. Para las pantallas de este prototipo (formularios de captura, listados de pocas decenas de elementos y reportes tabulares) esa diferencia no resulta perceptible, y la penalización queda compensada por la posibilidad de mantener una sola base de código. Si el sistema evolucionara hacia funciones intensivas en gráficos o en procesamiento local de imágenes, el balance habría que reconsiderarlo.

En el backend, Spring Boot impone un consumo de memoria superior al de alternativas más ligeras y un tiempo de arranque mayor, lo que encarece el alojamiento. A cambio aporta un modelo de seguridad declarativo y una capa de persistencia que reducen el código

propio en las áreas más sensibles del sistema (autorización por rol y consistencia transaccional del cierre financiero), precisamente donde un error tendría consecuencias económicas.

En la capa de datos, PostgreSQL exige un servidor en operación permanente, condición que un motor embebido no impondría. La necesidad de que varios usuarios accedan a la misma información desde dispositivos distintos descarta esa alternativa, y el control de aislamiento transaccional que ofrece el gestor es la garantía de que dos registros simultáneos sobre el mismo embarque no produzcan un estado inconsistente. El almacén local del dispositivo, en cambio, sí utiliza SQLite, que sirve de soporte a la cola de sincronización.

Ninguna de las tres tecnologías genera costos de licenciamiento, de modo que el criterio económico se resolvió sobre la infraestructura de ejecución y no sobre la adquisición del software.

1.13. Metodologías Ágiles

1.13.1 Scrum

Scrum es un marco de trabajo ágil para el desarrollo de software que organiza el trabajo en iteraciones cortas denominadas sprints, típicamente de dos semanas de duración [20]. Cada sprint produce un incremento potencialmente entregable del producto, lo que permite obtener retroalimentación temprana y adaptar los requerimientos de manera continua.

El proceso Scrum se articula en torno a tres roles principales: el Product Owner, responsable de definir y priorizar el backlog del producto; el Scrum Master, facilitador del proceso; y el equipo de desarrollo, encargado de la implementación [20]. Adicionalmente, incorpora ceremonias regulares (planificación del sprint, revisión y retrospectiva) que garantizan la transparencia, la inspección y la adaptación continua del proceso, así como artefactos como el Product Backlog, el Sprint Backlog y el incremento.

1.13.2 Usabilidad y Evaluación SUS

La usabilidad se relaciona con la efectividad, la eficiencia y la satisfacción con que usuarios específicos alcanzan objetivos en un contexto determinado [21]. Constituye una dimensión fundamental de la calidad, especialmente en contextos donde los usuarios tienen perfiles técnicos variados o limitada experiencia con herramientas digitales, como ocurre en el entorno de una hacienda bananera.

El System Usability Scale (SUS) es un cuestionario estandarizado de 10 ítems con escala de respuesta de 1 a 5, que permite evaluar la percepción de usabilidad de un sistema de manera rápida y confiable [22]. La puntuación final oscila entre 0 y 100 puntos. Para la interpretación se adoptó como referencia el promedio convencional de aproximadamente 68 puntos reportado en estudios normativos de la escala [23]. Por su brevedad y validez demostrada en múltiples contextos, el SUS es ampliamente utilizado en evaluaciones de usabilidad de sistemas de información.

1.14. Trabajos Relacionados

1.14.1 Aplicación móvil para gestión institucional

Pimple y Gupta [24] presentaron el diseño y desarrollo de una aplicación móvil para la gestión de eventos institucionales, orientada a centralizar la información en una plataforma de acceso rápido. La relevancia de este trabajo radica en que ilustra el uso de aplicaciones móviles para centralizar la gestión de información institucional, un enfoque afín al adoptado en el presente proyecto.

1.14.2 React Native para desarrollo multiplataforma

Shevtsiv et al. [15] analizaron las características, ventajas y limitaciones de React Native como alternativa para el desarrollo multiplataforma, destacando el uso de una base de código compartida para Android e iOS. Este trabajo respalda la elección de React Native como tecnología frontend del presente proyecto.

1.14.3 Digitalización de PYMES agroindustriales

Martínez, Ramírez y Sotomayor [5] analizaron los procesos de digitalización y cambio tecnológico en las mipymes agrícolas y agroindustriales de América Latina. El estudio examina las brechas de adopción tecnológica y la importancia de estas tecnologías para fortalecer la productividad y la competitividad del sector. Este antecedente contextualiza la necesidad de desarrollar soluciones adaptadas a pequeñas organizaciones agrícolas con recursos y conectividad limitados.

1.14.4 Alternativas ERP de Código Abierto y Justificación del Desarrollo a Medida

Antes de optar por un desarrollo propio corresponde examinar las soluciones ERP de código abierto disponibles en el mercado, que ofrecen módulos de inventario, facturación y gestión administrativa susceptibles de personalización y que, en principio, podrían cubrir parte de los requerimientos de la hacienda.

Odoo es la alternativa más extendida. Su arquitectura modular, su comunidad amplia y la existencia de localizaciones para Ecuador, incluida la facturación electrónica ante el Servicio de Rentas Internas, lo convierten en un candidato serio. Sus principales inconvenientes en este contexto son el peso de la infraestructura que requiere, la dependencia de una conexión permanente en su cliente web y el hecho de que sus aplicaciones móviles reproducen la lógica del escritorio, sin un modo de operación sin conexión adecuado para zonas sin cobertura.

ERPNext ofrece una cobertura funcional comparable con una interfaz más ligera y una curva de instalación menos exigente, pero su localización tributaria para Ecuador no está consolidada y su modelo de datos no contempla figuras propias de la operación bananera. Dolibarr destaca por su simplicidad y su bajo consumo de recursos, aunque su alcance funcional es menor y su enfoque se orienta a la pequeña empresa comercial más que a la producción agrícola.

La evaluación se realizó sobre cuatro criterios: la cobertura de las reglas de negocio específicas de la hacienda, la capacidad de operar sin conexión, el esfuerzo de adaptación estimado y el costo total, considerando infraestructura y mantenimiento.

El criterio determinante fue el primero. Las reglas que estructuran la operación de la Hacienda San Diego (el reparto de los gastos del embarque en proporción a las cajas aportadas por cada dueño, el cálculo del ratio de conversión por propietario dentro de una finca de tenencia compartida, la liquidación de la cuadrilla al cierre del embarque y la separación de la información entre dueños, mayordomos y facturadores) no corresponden a procesos estándar de ninguna de las tres plataformas. Implementarlas exigiría desarrollar módulos a medida sobre el marco de trabajo de cada producto, con lo cual el esfuerzo de adaptación se aproximaría al de un desarrollo propio, pero condicionado además por las convenciones internas de la plataforma.

El segundo criterio reforzó la decisión. El requerimiento de registrar operaciones en zonas sin cobertura de red es estructural en este caso y ninguna de las tres alternativas lo resuelve de forma nativa en su cliente móvil; incorporarlo supondría construir igualmente una aplicación móvil propia con su capa de sincronización.

Sobre el costo, si bien las tres opciones carecen de licenciamiento, exigen un servidor permanente y tareas de administración y actualización que la hacienda no está en condiciones de sostener. A ello se añade una consideración académica: el objetivo del trabajo incluye el diseño y la construcción de la arquitectura del sistema, actividad que la configuración de una plataforma existente no permitiría desarrollar.

Debe precisarse el alcance de esta comparación. Se apoya en la documentación pública de los tres productos y en el análisis de requerimientos de la hacienda, y no en implementaciones piloto ni en mediciones comparativas realizadas sobre cada plataforma.

Se trata, por tanto, de una justificación argumentada de la decisión de diseño y no de un estudio comparativo experimental.

Metodología

El desarrollo del trabajo se organizó en seis fases encadenadas: diagnóstico de la operación, especificación de los requisitos, diseño de la solución, construcción en cuatro sprints, validación empírica y análisis de los resultados. Cada fase entrega un producto que alimenta la siguiente, y la última cierra el ciclo contrastando lo medido con las metas fijadas en el objetivo general. La Figura 1 presenta gráficamente ese flujo y la Tabla 1 resume esa secuencia con sus técnicas, instrumentos y productos.

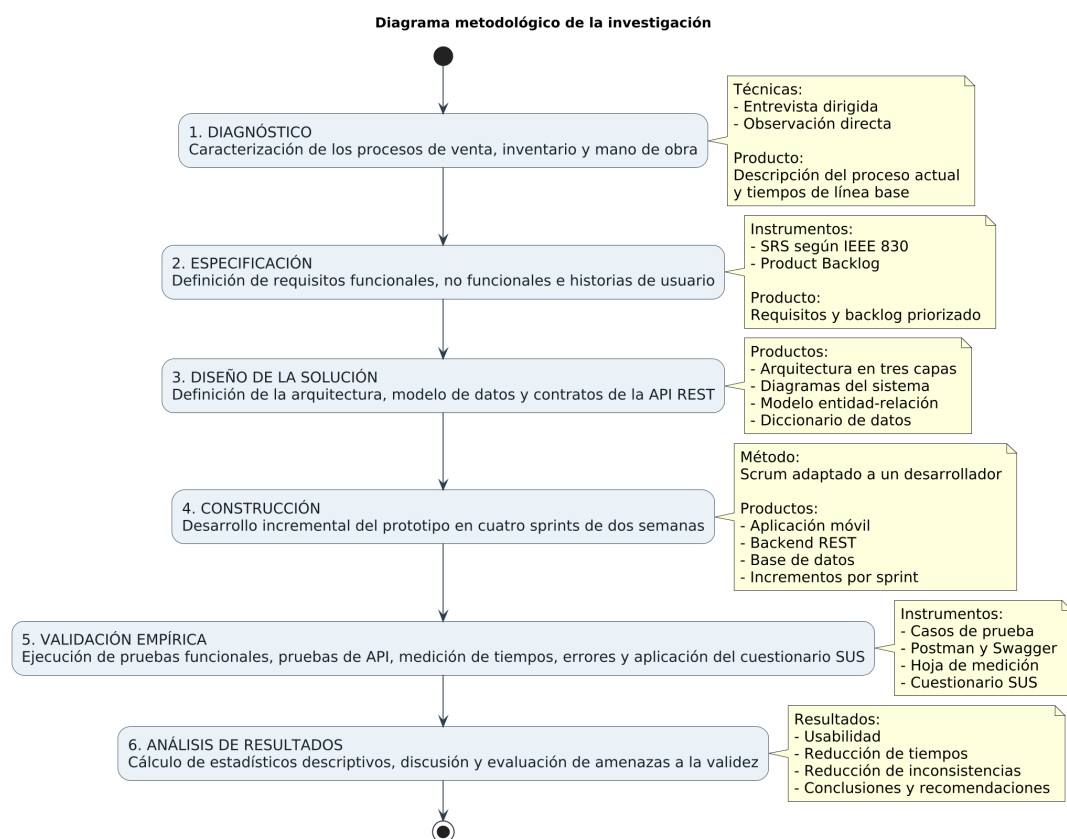


Figura 1 Diagrama metodológico de la investigación

Tabla 1 Flujo metodológico de la investigación

Fase	Actividades	Técnicas e instrumentos	Producto
1. Diagnóstico	Caracterización de los procesos de venta, inventario y mano de obra	Observación directa (ficha, Anexo C) y entrevista dirigida al propietario (Anexo B)	Descripción del proceso actual y tiempos de línea base
2. Especificación	Definición de requisitos funcionales y no funcionales,	SRS según IEEE 830, Product Backlog (Anexo D)	Especificación de requisitos y backlog priorizado

Fase	Actividades	Técnicas e instrumentos	Producto
	historias de usuario y criterios de aceptación		
3. Diseño	Arquitectura en tres capas, modelo entidad-relación, contratos de la API REST	Diagramas de capas, componentes, despliegue y modelo de datos	Diseño arquitectónico y diccionario de datos
4. Construcción	Cuatro sprints de dos semanas con revisión al cierre de cada uno	Scrum adaptado a un desarrollador, Sprint Backlog (Anexo E), control de versiones en GitLab	Prototipo funcional y registro de cambios de alcance (Anexo A)
5. Validación	Pruebas funcionales y de API, medición de tiempos y errores, aplicación del cuestionario SUS	Casos de prueba (Anexos G y H), hoja de medición (Anexo J), instrumento SUS (Anexo I)	Resultados de usabilidad, eficiencia y calidad del registro
6. Análisis	Cálculo de estadísticos descriptivos, discusión de resultados y evaluación de amenazas a la validez	Estadística descriptiva; diseño preexperimental de un solo grupo (apartado 3.6)	Discusión, conclusiones y líneas de trabajo futuro

1.15. Tipo de Investigación

El presente proyecto corresponde a una investigación de tipo aplicada y descriptiva [25]. Es aplicada porque busca resolver un problema real y concreto (la gestión manual de la facturación e inventario en la Hacienda Bananera San Diego) mediante el desarrollo de una solución tecnológica funcional, en lugar de generar conocimiento teórico abstracto. Su finalidad es producir un prototipo de software que aporte un beneficio tangible a la organización estudiada.

Asimismo, es descriptiva porque, antes de proponer la solución, requiere caracterizar detalladamente los procesos actuales de venta, control de inventario y administración de la mano de obra de la hacienda. Esta caracterización permite identificar las deficiencias existentes y, a partir de ellas, definir los requerimientos que el sistema debe satisfacer.

1.16. Enfoque de Investigación

El proyecto adopta un enfoque mixto, combinando elementos cualitativos y cuantitativos a lo largo de sus distintas fases.

El componente cualitativo se aplica durante el levantamiento y análisis de requerimientos. Mediante la observación directa de las operaciones de la hacienda y la realización de una entrevista dirigida con el propietario, se recopila información sobre la forma en que se desarrollan actualmente los procesos, las dificultades que enfrentan y las necesidades que el sistema debe atender. Este enfoque permite

comprender el contexto y la lógica de negocio particular de una finca de propiedad compartida.

El componente cuantitativo se aplica durante la fase de validación. A través del cuestionario System Usability Scale (SUS) se obtiene una medición numérica de la percepción de usabilidad del prototipo, y mediante la comparación de los tiempos de registro de operaciones antes y después de la implementación se evalúa de manera objetiva la mejora en la eficiencia de los procesos.

Esta fase cuantitativa corresponde, en términos de diseño, a un estudio preexperimental de un solo grupo con medición antes y después, sin grupo de control ni asignación aleatoria. Sus características, su protocolo de medición y las amenazas a la validez que de él se derivan se detallan en el apartado 3.6.

1.17. Población y Muestra

La población del estudio está constituida por las personas vinculadas a la operación administrativa y productiva de la Hacienda Bananera San Diego, incluyendo a los dueños, el personal encargado del registro de información y los trabajadores que participan en las labores de la finca y en los embarques.

Dado el tamaño reducido de la organización, se emplea un muestreo no probabilístico por conveniencia. La muestra está conformada por cinco usuarios directamente relacionados con el uso del sistema, seleccionados con el criterio de cubrir todos los perfiles que el prototipo diferencia: el propietario, que ejerce a la vez los roles de dueño y de administrador, el mayordomo, el facturador, el bodeguero y el capitán de cuadrilla. Los cinco interactuaron con el prototipo durante la fase de validación y respondieron el cuestionario SUS. El criterio de selección fue, por tanto, la cobertura de roles y no la proporcionalidad respecto de la población, de modo que la muestra incluye tanto perfiles administrativos como de campo. La evaluación tiene un alcance exploratorio y se orienta al contexto específico de la hacienda, por lo que sus resultados no pretenden generalizarse a otras organizaciones.

1.18. Técnicas de Recolección de Información

Para la recolección de la información se emplean tres técnicas complementarias, aplicadas en distintas fases del proyecto.

1.18.1 Entrevista dirigida

Se aplicó una entrevista dirigida (estructurada) al propietario de la Hacienda Bananera San Diego, conducida a partir de un guion de preguntas previamente definido y formuladas en un orden establecido, con el fin de mantener el control temático de la conversación y garantizar la cobertura de todos los aspectos de interés [25]. La entrevista se organizó en torno a cuatro ejes: el proceso de venta mediante embarques y la diferenciación de la producción por dueño; el control del inventario de productos de bodega; la administración de la mano de obra, tanto de la cuadrilla del embarque como del personal semanal; y las condiciones de conectividad de la zona. A diferencia de una entrevista abierta, en la entrevista dirigida el entrevistador guía la sesión siguiendo el guion, lo que permite obtener información comparable y enfocada en los requerimientos del sistema. Como resultado, esta técnica permitió identificar las reglas de negocio particulares de la finca, entre ellas el reparto de los gastos del embarque en proporción a las cajas aportadas, el cálculo del ratio de conversión por dueño, el pago del evaluador de la fruta, la liquidación de la cuadrilla al cierre del embarque y el pago semanal de los jornales del personal. El guion utilizado se presenta en el Anexo B.

Debe advertirse una limitación metodológica en esta técnica. El levantamiento de requerimientos se apoyó en un único informante (el propietario), complementado con la observación directa de las operaciones, mientras que el sistema desarrollado contempla seis perfiles de usuario con responsabilidades diferenciadas. La ingeniería de requisitos recomienda contrastar la información con los distintos actores involucrados en el proceso de negocio; al no haberse aplicado esa triangulación, los requisitos funcionales provienen en buena medida de una interpretación centralizada en la visión del propietario y de la lectura que el investigador hizo de las operaciones observadas.

Esta situación reduce la confianza en que los requisitos representen fielmente las necesidades de todos los perfiles operativos, en particular de aquellos que no participaron en el levantamiento. Como medida parcial de contraste, las funcionalidades resultantes fueron revisadas con el propietario al cierre de cada sprint y validadas al final del proyecto mediante el acta que consta en el Anexo L. Se reconoce, no obstante, que dicho mecanismo no sustituye la consulta directa a cada perfil, y se sugiere que investigaciones posteriores incorporen entrevistas o actas de aceptación firmadas por representantes de todos los roles del sistema.

1.18.2 Observación directa

La observación directa se utiliza para examinar de primera mano el desarrollo de las operaciones en la hacienda, especialmente durante los días de embarque. Esta técnica complementa la información obtenida en la entrevista y permite validar

cómo se ejecutan realmente los procesos, identificando detalles que podrían no surgir en una conversación, como los tiempos de registro y las condiciones de conectividad en el campo. La ficha empleada para documentar estas mediciones se incluye en el Anexo C.

1.18.3 Encuesta de usabilidad (SUS)

Para la fase de validación se aplica el cuestionario System Usability Scale (SUS), un instrumento estandarizado de 10 ítems con escala de respuesta de 1 a 5 que mide la percepción de usabilidad del sistema [22]. Los resultados se procesan mediante estadística descriptiva para obtener una puntuación global sobre 100, estableciéndose como criterio de aceptación un valor mínimo de 68 puntos. El instrumento aplicado se incluye en el Anexo I.

1.19. Metodología de Desarrollo de Software

1.19.1 Selección de Scrum

Entre los distintos modelos del ciclo de vida del desarrollo de software, los enfoques ágiles destacan por su capacidad de adaptación frente a requisitos cambiantes [12]. Para el desarrollo del sistema se emplea la metodología ágil Scrum, debido a su flexibilidad para adaptarse a cambios en los requerimientos y a su enfoque incremental en la entrega del producto [20]. Esta metodología resulta apropiada para el proyecto porque permite construir el prototipo de manera progresiva, obteniendo retroalimentación del propietario de la hacienda en cada iteración y ajustando el sistema a las necesidades reales de la finca. De hecho, varias funcionalidades del sistema (como el cierre financiero del embarque, la planilla semanal de jornales y el rol de dueño) surgieron precisamente de la retroalimentación recibida durante las revisiones de sprint, lo que respalda la pertinencia del enfoque iterativo.

1.19.2 Adaptación de Scrum a un equipo de un solo desarrollador

Scrum está concebido como un marco de trabajo colaborativo con responsabilidades diferenciadas de Product Owner, Scrum Master y equipo de desarrollo [20]. Dado que el presente trabajo de titulación fue desarrollado por un único estudiante, se aplicó una adaptación al contexto individual, conservando la planificación, la revisión de incrementos, la retrospectiva y los artefactos de seguimiento, y simplificando las actividades que requieren la participación de un equipo numeroso.

En esta adaptación, el estudiante asume de forma simultánea los roles técnicos de Scrum Master y equipo de desarrollo, mientras que el propietario de la hacienda cumple el rol de Product Owner, aportando y priorizando los requerimientos y validando cada incremento. El director del trabajo de titulación ejerce una función de supervisión y orientación metodológica. En cuanto a las ceremonias, se conservan la planificación de cada sprint, la revisión del incremento con el propietario y una retrospectiva personal de mejora; la reunión diaria (daily standup), pensada para sincronizar a varios miembros, se sustituye por un registro breve de avance y pendientes llevado por el propio desarrollador. Los artefactos (Product Backlog, Sprint Backlog e incremento) se mantienen sin cambios, pues son igualmente útiles en un proyecto individual.

1.19.3 Organización en sprints

El proyecto se organiza en cuatro sprints de dos semanas cada uno. Cada sprint comprende las ceremonias adaptadas de planificación, desarrollo, revisión y retrospectiva, garantizando la mejora continua del proceso. La distribución final de las actividades ejecutadas por sprint se presenta en la Tabla 2 Esta planificación fue actualizada de manera incremental con los requerimientos identificados durante las revisiones de cada sprint. El Product Backlog consolidado y el detalle de cada sprint, con sus revisiones y cambios, se presentan en los Anexos D y E.

Tabla 2 Distribución final de actividades ejecutadas por sprint

Sprint	Actividad	Resultado alcanzado
Sprint 1	Levantamiento de requerimientos y análisis. Entrevista dirigida, observación y diseño de arquitectura.	Especificación de requisitos (SRS), historias de usuario y diseño de la arquitectura.
Sprint 2	Diseño de base de datos e implementación de los módulos de autenticación, seguridad e inventario de productos.	Modelo de base de datos, autenticación con JWT, control de roles e inventario funcional con categorías, proveedores y existencias por dueño.
Sprint 3	Desarrollo del módulo de embarques con su cierre financiero, gastos, mano de obra con planilla semanal, facturación y reportes.	Módulos de venta, cierre de embarque, gastos, jornales, comprobantes y reportes implementados.
Sprint 4	Integración, modo sin conexión, ejecución de pruebas funcionales y evaluación de usabilidad.	Prototipo integrado, sincronización offline operativa, pruebas funcionales ejecutadas y evaluación SUS aplicada a cinco usuarios.

1.19.4 Evolución del alcance durante el desarrollo

Durante las revisiones de los sprints se identificaron necesidades adicionales relacionadas con los procesos de facturación e inventario inicialmente planteados. A partir de la retroalimentación proporcionada por el propietario de la Hacienda Bananera San Diego, se incorporaron funcionalidades complementarias como la gestión de jornales, el cierre financiero de los embarques, el control de la cuadrilla, la auditoría de operaciones y los reportes diferenciados por dueño.

Estas funcionalidades no modificaron el título, el problema de investigación ni el objetivo general del proyecto. Por el contrario, permitieron representar de manera más completa los procesos reales vinculados con la facturación, el inventario y la gestión administrativa de la hacienda. Su incorporación se realizó de forma incremental mediante la actualización del Product Backlog durante las revisiones de los sprints (práctica de refinamiento prevista en el propio marco de trabajo [20]), manteniendo las tecnologías, la arquitectura y la metodología Scrum definidas originalmente en el anteproyecto.

La relación entre el alcance inicialmente planteado y las funcionalidades incorporadas durante el desarrollo se presenta en la Tabla 3 El registro de los cambios identificados en cada sprint se incluye en el Anexo A.

Tabla 3 Relación entre el alcance inicial y las funcionalidades incorporadas

Componente relacionado	Funcionalidades incorporadas	Relación con el objetivo del proyecto
Registro de ventas y facturación	Ciclo de embarque, cierre financiero, reparto proporcional de costos, reliquidación y comprobantes internos en PDF.	Permitió digitalizar el proceso de venta y determinar los valores correspondientes a cada dueño.
Control de inventario	Existencias diferenciadas por dueño, movimientos de entrada y salida, proveedores, stock mínimo y fechas de caducidad.	Permitió conocer la disponibilidad de productos y controlar qué existencias corresponden a cada propietario.
Costos asociados con la producción	Registro de cuadrilla, evaluador de fruta, gastos y jornales semanales.	Permitió incorporar los costos necesarios para calcular el resultado económico de cada embarque.
Reportes básicos	Reportes de ventas, gastos, inventario, mano de obra y productividad por dueño.	Facilitó el seguimiento individual de la producción, los costos y los resultados de cada propietario.
Arquitectura y seguridad del sistema	Seis roles de acceso, restricciones de información y auditoría de operaciones.	Permitió proteger la información y registrar al usuario responsable de cada operación.
Operación con conectividad limitada	Almacenamiento local en SQLite y sincronización posterior de embarques.	Permitió registrar una operación crítica en zonas rurales sin conexión permanente a internet.

1.19.5 Procedimiento de desarrollo

El desarrollo del sistema sigue un proceso iterativo e incremental, estructurado en las siguientes etapas:

5. Identificación y análisis del problema en la Hacienda Bananera San Diego.
6. Levantamiento de requerimientos mediante entrevista dirigida y observación directa.
7. Modelado del sistema y diseño de la base de datos.

8. Implementación de los módulos del sistema según los sprints definidos.
9. Integración de los componentes del frontend y el backend.
10. Ejecución de pruebas funcionales y corrección de errores.
11. Validación del sistema con los usuarios mediante el cuestionario SUS.

1.19.6 Herramientas de desarrollo

Las herramientas y tecnologías empleadas para el desarrollo del sistema se detallan a continuación: para el frontend móvil se utiliza Expo con React Native y TypeScript; para el backend, Java 17 con Spring Boot; para la base de datos, PostgreSQL; para el control de versiones, Git y GitHub; para las pruebas de la API REST, Postman y la documentación interactiva generada con Swagger (OpenAPI). Esta combinación tecnológica permite construir una aplicación multiplataforma, escalable y mantenible, acorde con los objetivos del proyecto.

1.20. Diseño de la Validación Empírica

1.20.1 Diseño preexperimental de un solo grupo

La fase de validación del prototipo corresponde a un diseño preexperimental de un solo grupo con medición antes y después (pretest–posttest sin grupo de control). Un mismo conjunto de operaciones administrativas de la hacienda se midió primero bajo el procedimiento manual vigente y luego bajo el prototipo ERP, sin asignación aleatoria, sin grupo de comparación y sin control sobre variables externas al proceso.

Esta precisión es necesaria porque delimita el tipo de afirmación que el estudio puede sostener. El diseño permite documentar la reducción observada en los tiempos de registro y en el número de inconsistencias, y establecer que esa reducción es compatible con un efecto positivo del prototipo. No permite, en cambio, atribuir la mejora exclusivamente al sistema: factores como la familiaridad del operador con la tarea, el orden en que se ejecutaron las mediciones o las condiciones particulares de la jornada observada no fueron controlados y pueden haber contribuido al resultado.

En consecuencia, los resultados de esta fase se interpretan como evidencia descriptiva propia del caso estudiado y no como una demostración de causalidad. Esta limitación no invalida el desarrollo del software ni los valores obtenidos, pero acota el alcance de las conclusiones científicas que pueden derivarse de ellos.

1.20.2 Justificación metodológica de la meta del 70 %

El objetivo general fija una reducción de al menos el 70 % en el tiempo de registro de operaciones. Este valor no constituye un umbral tomado de la literatura ni un parámetro normativo del sector, sino un criterio de aceptación acordado con el propietario de la hacienda durante la fase diagnóstica, a partir de los tiempos observados en el procedimiento manual.

El razonamiento que sustenta el acuerdo fue el siguiente: las tres operaciones administrativas evaluadas consumían en conjunto alrededor de sesenta minutos por ciclo de embarque, ejecutados por el propietario al final de una jornada de campo de aproximadamente ocho horas. El propietario estableció que una mejora inferior a dos tercios del tiempo no justificaría el costo de aprendizaje y el cambio de rutina que implica sustituir los registros en papel; el 70 % se adoptó como el redondeo operativo de ese punto de corte.

Se explicita aquí su origen para que el lector distinga entre el indicador (el porcentaje de reducción efectivamente medido) y la meta (el valor mínimo que el interesado consideró aceptable). La meta es, por tanto, un criterio de decisión definido por el usuario del sistema y no un parámetro derivado de evidencia empírica previa, y así debe interpretarse al valorar su cumplimiento.

1.20.3 Protocolo de medición de tiempos y errores

Las mediciones se realizaron el 7 de julio de 2026, durante una única jornada de observación de aproximadamente ocho horas, correspondiente a un ciclo completo de embarque (corte, selección, empaque y cierre). Se documenta a continuación el protocolo aplicado, con el fin de permitir su reproducción por parte de otros investigadores.

Operaciones evaluadas. Se seleccionaron las tres operaciones administrativas de mayor recurrencia y mayor carga de registro manual: el cierre de embarque, la elaboración de la planilla semanal de pago del personal fijo de la finca y el registro de un movimiento de inventario. Conviene precisar que esta planilla corresponde al personal de nómina semanal y no a la liquidación de la cuadrilla del embarque, que se resuelve dentro del propio cierre y por tanto queda contabilizada en la primera operación.

Criterio de cronometraje. El tiempo se contabilizó desde que el operador tomaba el soporte de registro (cuaderno y calculadora en el procedimiento manual, dispositivo móvil con el prototipo) hasta que el registro quedaba completo y verificado. Se excluyeron del cronometraje las actividades de campo, las interrupciones ajenas a la tarea y los tiempos de espera por terceros. Los valores se registraron en minutos con dos decimales.

Número de repeticiones. Cada operación se midió una sola vez por condición, es decir, una medición bajo el procedimiento manual y una medición con el prototipo. El estudio no contempló repeticiones adicionales ni sesiones en días distintos, por lo que los valores reportados son mediciones puntuales y no promedios de ensayos múltiples.

Responsable de la medición y ejecutante. Las tres operaciones fueron ejecutadas por el propietario de la hacienda, responsable habitual de estas tareas. El cronometraje y el registro de los datos estuvieron a cargo del investigador, mediante cronómetro digital y una hoja de registro cuyo consolidado se presenta en el Anexo J.

Contabilización de errores. Se consideró error toda inconsistencia detectada al verificar el registro contra la operación real: cantidad anotada que no coincide con el conteo físico, error aritmético en un total, dato obligatorio omitido, registro duplicado o registro ilegible. La verificación se realizó al cierre de cada operación, sobre el mismo conjunto de transacciones en ambas condiciones. La ficha registró el número total de inconsistencias por condición, sin desagregarlas por categoría. Esta decisión de diseño del instrumento limita el análisis posterior: permite calcular la reducción global, pero no la contribución de cada mecanismo de validación a esa reducción. Se consigna aquí como limitación del protocolo y no se reconstruye la distribución a posteriori, por cuanto la ficha original no contiene el dato.

1.20.4 Tratamiento estadístico de los datos

El análisis cuantitativo se limita a estadística descriptiva. Para el cuestionario SUS y para los porcentajes de reducción de tiempos se reportan la media, la desviación estándar, la mediana y el rango, acompañados siempre del tamaño de muestra correspondiente. En el caso del SUS se incluye además el intervalo de confianza al 95 %, calculado con la distribución t de Student para muestras pequeñas, con el propósito de hacer visible la imprecisión asociada a un n reducido.

No se aplicaron pruebas de estadística inferencial. La decisión responde al carácter exploratorio del estudio, al tamaño reducido de la muestra (cinco participantes en la evaluación de usabilidad y una sola medición por operación en la comparación de tiempos) y a que los datos provienen de un estudio de caso único. Bajo estas condiciones, una prueba de contraste de hipótesis carecería de potencia estadística y sugeriría una capacidad de generalización que el diseño no posee.

En coherencia con lo anterior, las diferencias observadas se describen como propias del caso estudiado y no se extrapolan a la población de haciendas bananeras. Los porcentajes se reportan con dos decimales por consistencia con las hojas de cálculo del Anexo J, pero esa precisión numérica corresponde al cálculo aritmético y no a la precisión del diseño de medición, que se apoya en observaciones únicas.

1.20.5 Amenazas a la validez

Validez interna. La ausencia de grupo de control impide descartar explicaciones alternativas a la mejora observada. Se identifican tres amenazas principales: el efecto de aprendizaje, dado que el operador ejecutó la misma tarea dos veces y en la segunda ocasión ya conocía el flujo; el efecto de reactividad o Hawthorne, puesto que el ejecutante sabía que estaba siendo observado y que el prototipo era objeto de evaluación; y el sesgo del instrumento, ya que la medición y el desarrollo del sistema estuvieron a cargo de la misma persona.

Validez de constructo. El tiempo de registro y el conteo de inconsistencias son indicadores parciales de la calidad de la gestión administrativa. No capturan dimensiones como la trazabilidad de la información en el tiempo, la calidad de las decisiones que se toman con esos datos ni la carga cognitiva del operador.

Validez externa. El estudio se realizó en una sola hacienda, con cinco participantes y con un ciclo de embarque. Los resultados describen ese contexto específico y no son generalizables a otras unidades productivas, aun cuando compartan giro y tamaño.

Validez de conclusión estadística. El tamaño muestral limita la estabilidad de los estadísticos reportados: con cinco observaciones, la media del SUS es sensible a la variación de un solo participante, y con una medición por operación no es posible estimar la variabilidad natural del proceso. Por esta razón los valores se presentan acompañados de medidas de dispersión y se evita todo lenguaje confirmatorio en su interpretación.

Estrategias de mitigación y trabajo futuro. Para reducir estas amenazas, futuras investigaciones deberían repetir las mediciones en varias sesiones y con distintos operadores, separar los roles de desarrollador y evaluador, ampliar el número de participantes en la evaluación de usabilidad e incorporar al menos una unidad productiva adicional que permita contrastar los resultados.

Análisis y Diseño del Sistema

1.21. Análisis del Sistema Actual

Actualmente, la Hacienda Bananera San Diego gestiona sus procesos de forma manual mediante cuadernos físicos y recibos. A continuación se describen los dos procesos principales objeto de digitalización.

1.21.1 Proceso actual de venta (embarque)

El proceso de venta se ejecuta mediante embarques que ocurren una o dos veces por semana, y cada embarque corresponde a una semana determinada del año. El día del embarque, la producción de banano de todos los dueños se procesa de manera conjunta y se vende a un comprador a un precio por caja determinado por el mercado de esa semana; con frecuencia, el precio definitivo y la cantidad de cajas efectivamente aceptadas se conocen recién al final de la jornada. Un capitán coordina a la cuadrilla de trabajadores que ese día realizan el corte, el empaque y el embarque del producto, y un evaluador designado por el comprador verifica la calidad de la fruta, servicio por el cual también recibe un pago. Cada racimo se identifica con una cinta de color que, según el calendario de enfunde, indica la semana en que fue enfundado y, por lo tanto, su edad. Los racimos que no cumplen el estándar de calidad (por venir demasiado gruesos, delgados o explosivos) se retiran y se registran como una observación del embarque. Una vez finalizado, se anota manualmente en un cuaderno cuántos racimos cortó y cuántas cajas aportó cada dueño desde cada una de sus fincas, el precio acordado y el comprador, y se calcula a mano cuánto dinero corresponde a cada dueño, el ratio de conversión de cada uno (cajas obtenidas por racimo cortado) y cómo se reparten entre los dueños los costos del día: la cuadrilla, el evaluador y los demás gastos del embarque. El flujo de este proceso se representa mediante un diagrama en la Figura 1.

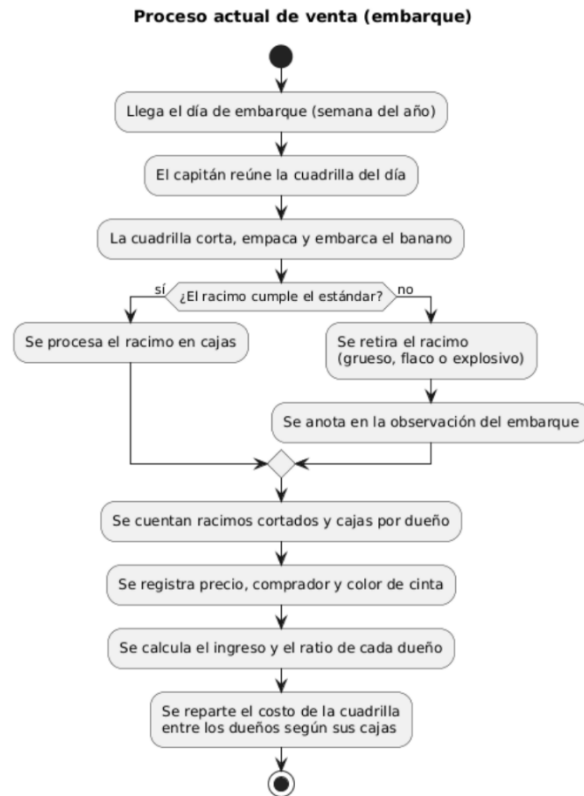


Figura 2 Proceso actual de venta (embarque)

1.21.2 Proceso propuesto de gestión del embarque

El proceso propuesto reemplaza el registro disperso en cuadernos por un flujo controlado desde la aplicación móvil. El facturador crea el embarque y registra la producción por dueño y finca; el capitán incorpora a la cuadrilla; y, al finalizar la jornada, se concilian las cajas y los racimos reales, se registra al evaluador y se confirma el precio definitivo. El sistema calcula los ingresos, distribuye los costos en proporción a las cajas aportadas, genera el cierre financiero, permite registrar los pagos y produce un comprobante interno en PDF. El proceso propuesto se muestra en la Figura 2.

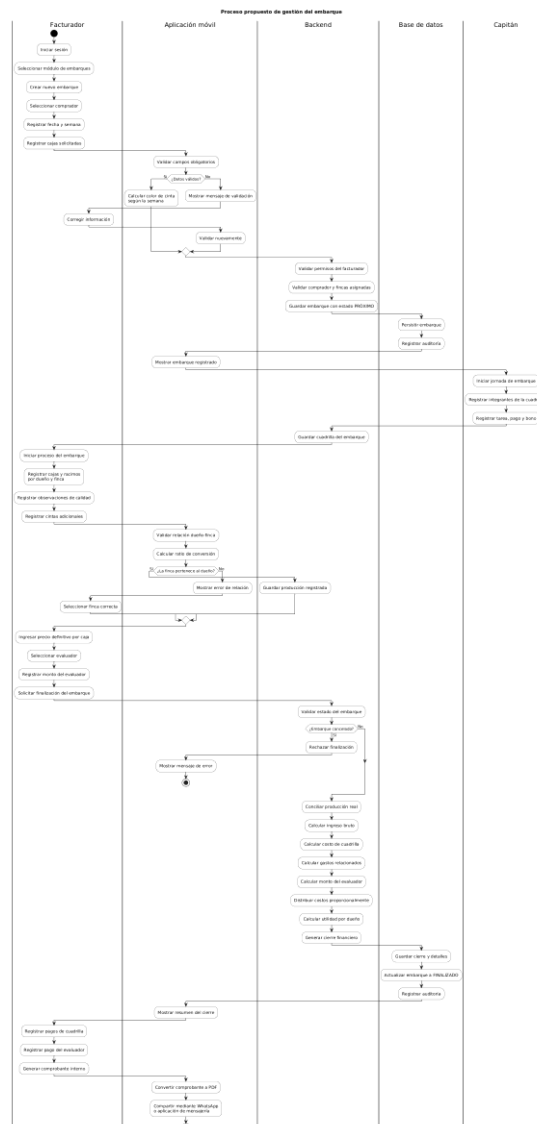


Figura 3 Proceso propuesto de gestión del embarque

1.21.3 Proceso actual de inventario

El inventario de productos de bodega (fertilizantes como la urea, químicos de fumigación y materiales como fundas y protectores) se controla de forma visual y se anota a mano en un cuaderno. No existe un registro sistemático de entradas y salidas, por lo que la disponibilidad de cada producto se conoce únicamente mediante la inspección directa de la bodega. Cuando un dueño compra productos para sus hectáreas, ese gasto se anota como propio, sin un control digital que permita conocer el

saldo en tiempo real ni distinguir con claridad qué parte de las existencias pertenece a cada dueño. El flujo de este proceso se representa en la Figura 3.

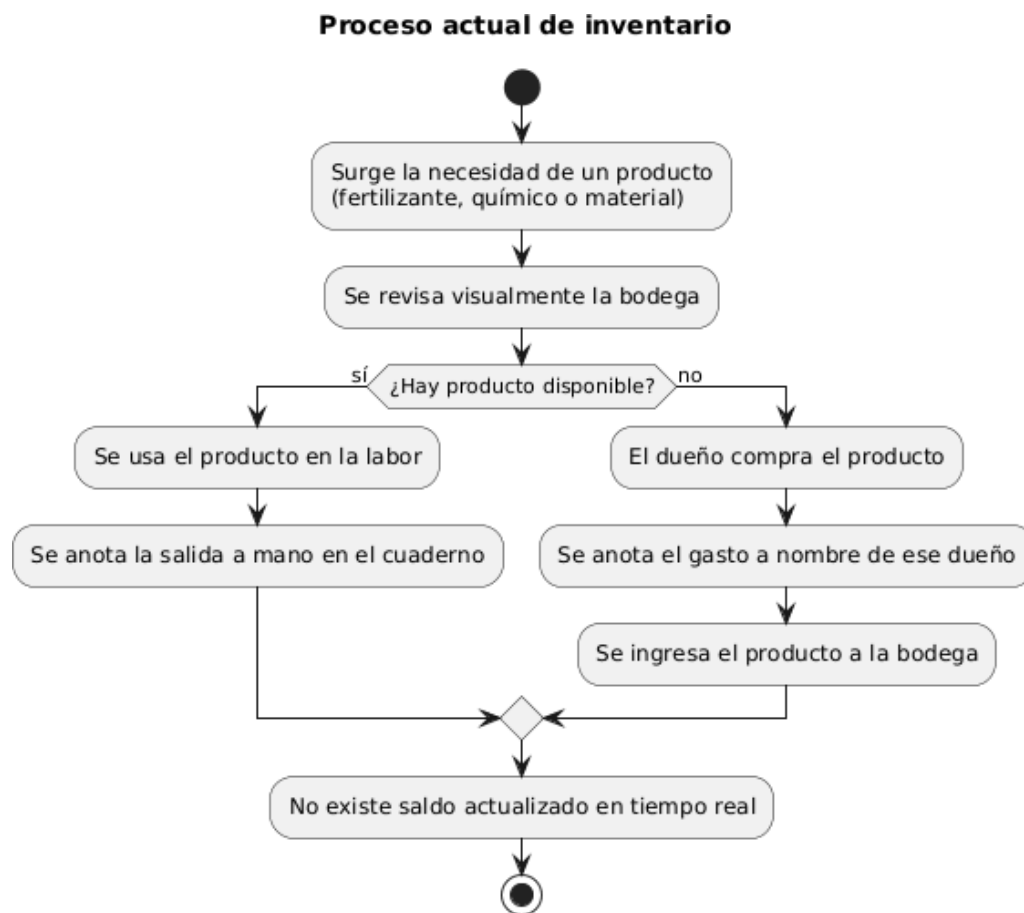


Figura 4 Proceso actual de inventario

1.21.4 Proceso propuesto de gestión del inventario

En el proceso propuesto, el bodeguero selecciona el producto y el dueño, registra una entrada o una salida y proporciona la información complementaria que corresponda, como proveedor, costo o fecha de caducidad. Antes de guardar una salida, el backend verifica que exista saldo suficiente; después de una operación válida, actualiza las existencias, evalúa el stock mínimo y conserva la trazabilidad del usuario responsable. El flujo se presenta en la Figura 4.

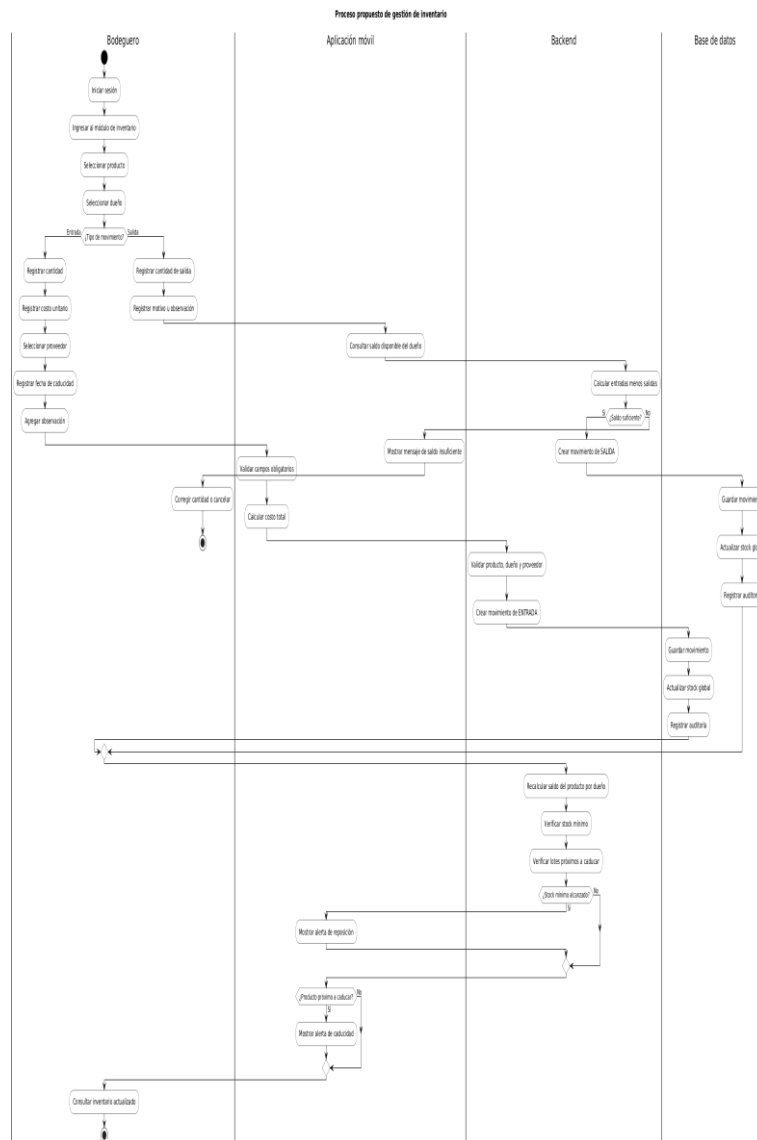


Figura 5 Proceso propuesto de gestión del inventario

1.22. Diseño de la Arquitectura

El sistema se estructura bajo una arquitectura cliente-servidor en tres capas, comunicadas mediante una API REST, y aplica el patrón Modelo-Vista-Controlador. La aplicación móvil, desarrollada en Expo/React Native, constituye la capa de presentación e incorpora un almacenamiento local en SQLite que permite operar sin conexión. El backend, desarrollado en Spring Boot, contiene la lógica de negocio y expone los servicios REST protegidos con JWT. La base de datos PostgreSQL gestiona la persistencia de la información.

1.22.1 Diagrama de capas

La organización en tres capas del sistema se representa en la Figura 5.

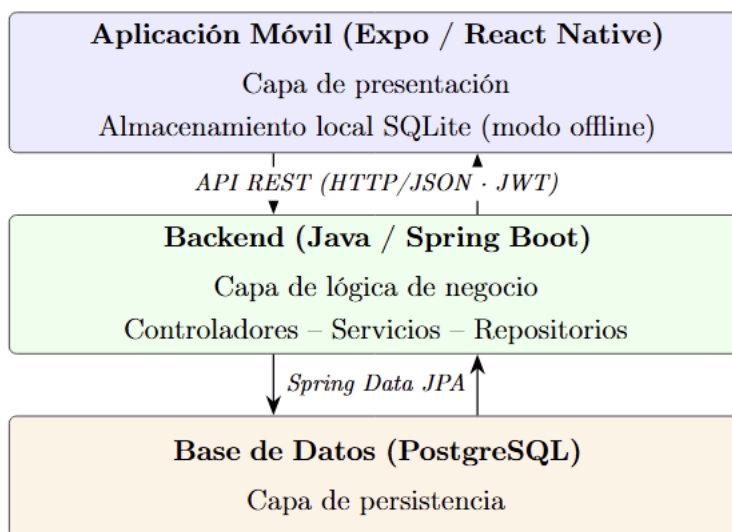


Figura 6 Arquitectura general del sistema en tres capas

1.1.2. Diagrama de componentes

El backend se organiza en módulos funcionales que exponen sus servicios a través de la API REST y comparten la capa de acceso a datos. La Figura 6 muestra los componentes principales del sistema y sus relaciones.

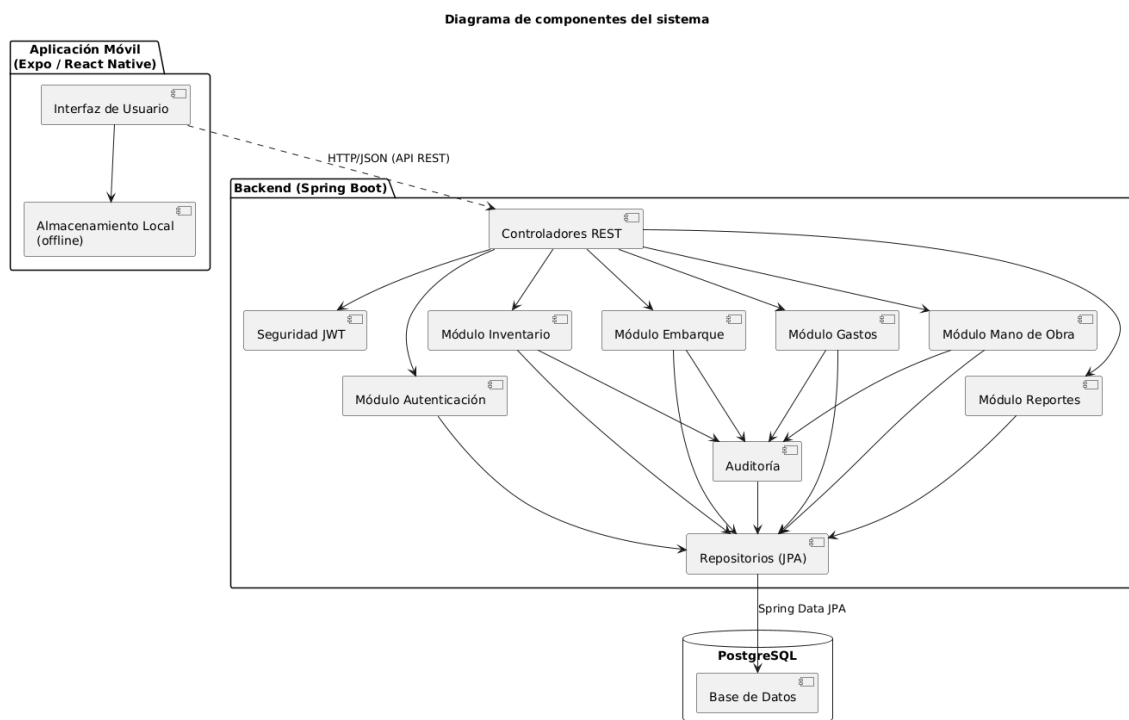


Figura 7 Diagrama de componentes del sistema

1.22.2 Diagrama de despliegue

Desde el punto de vista físico, el sistema se distribuye en el dispositivo móvil del usuario, el servidor de aplicaciones que ejecuta el backend y el servidor de base de datos. La Figura 7 representa esta distribución y los protocolos de comunicación entre los nodos.

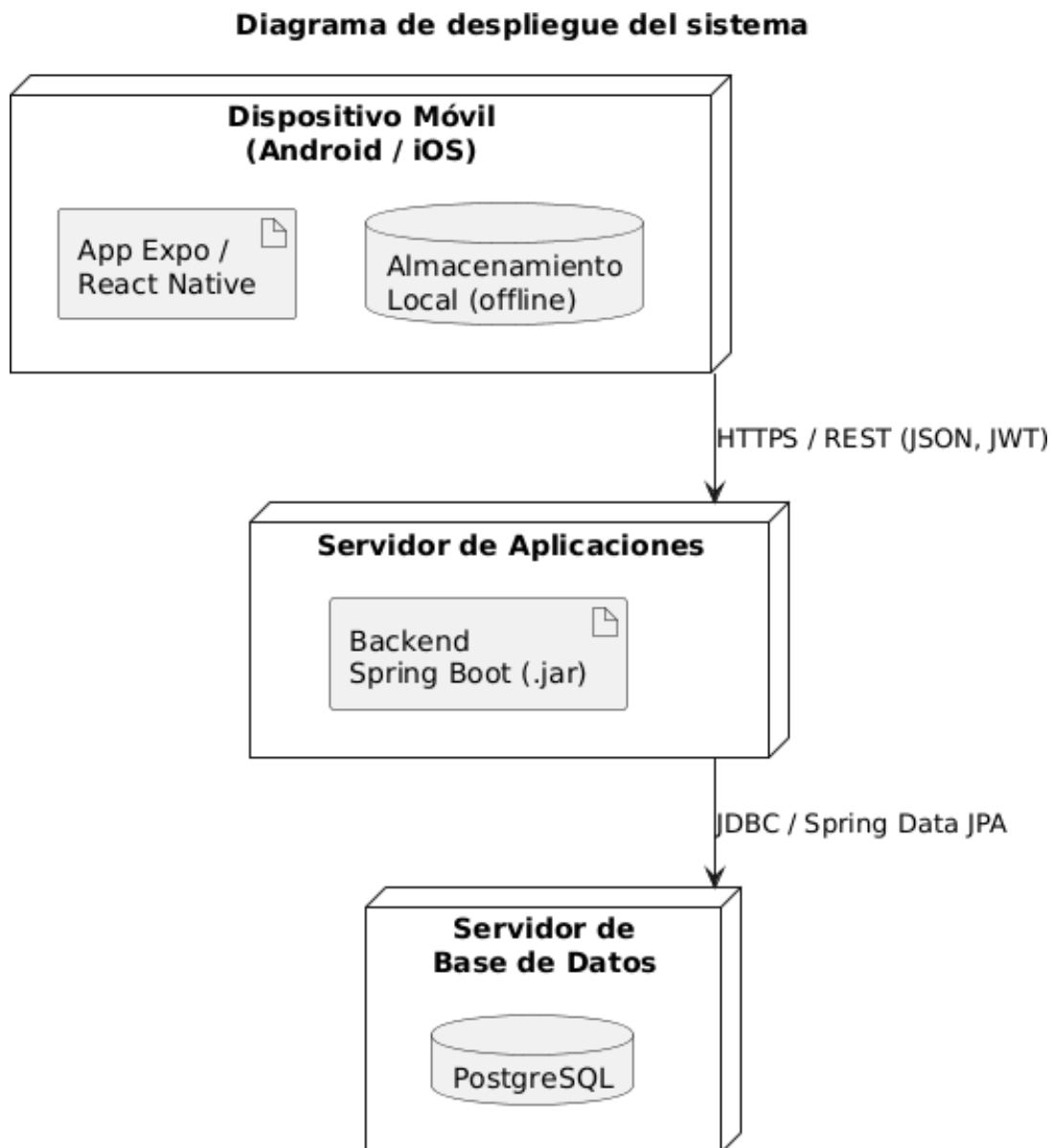


Figura 8 Diagrama de despliegue del sistema

1.23. Diseño de la Base de Datos

1.23.1 Modelo entidad-relación

El modelo de datos se compone de veintisiete tablas que representan la operación de una finca de propiedad compartida, incluida la tabla complementaria de precio histórico del banano incorporada posteriormente. Comprende la gestión de usuarios con seis roles y la auditoría de sus acciones; el registro de dueños y de sus fincas con ubicación geográfica; el registro de compradores, trabajadores, proveedores y evaluadores de fruta; el catálogo de productos de bodega clasificados por categoría y sus movimientos de inventario diferenciados por dueño; el calendario de cintas por semana; los embarques con su detalle por dueño y por finca (incluyendo cajas, racimos cortados y cintas adicionales) y la cuadrilla asociada; el cierre financiero del embarque con su detalle por dueño y la liquidación de la cuadrilla; los gastos de cuatro tipos con su estado de facturación; los jornales del personal semanal con su reparto entre dueños y la planilla de pago semanal por trabajador; y las tablas de asignación de acceso que vinculan facturadores con fincas y mayordomos con dueños. El modelo entidad-relación se presenta en la Figura 8.

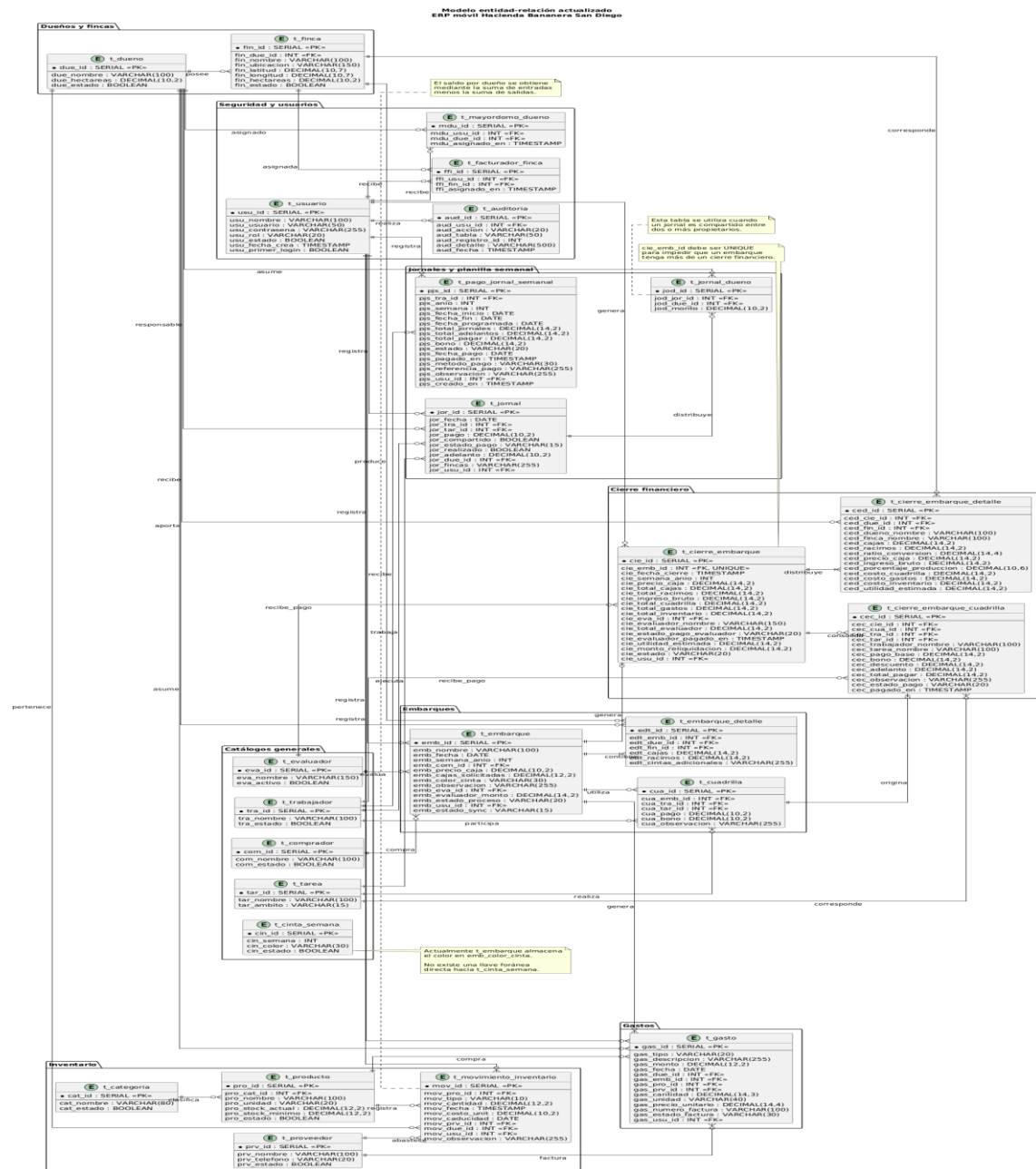


Figura 9 Modelo entidad-relación del sistema

1.23.2 Diccionario de datos

A continuación se describen las tablas del modelo de datos. El modelo se diseñó principalmente conforme a la tercera forma normal (3FN). Se incorporaron desnormalizaciones controladas para optimizar consultas frecuentes y conservar información histórica de los cierres, evitando que modificaciones posteriores en los catálogos alteren las liquidaciones ya generadas. Se emplea la convención de nombrar las tablas con el prefijo *t_* y las columnas con un prefijo de tres letras derivado del nombre de la tabla. El campo *due_hectareas* constituye un valor

derivado que el sistema calcula y almacena automáticamente a partir de la suma de las hectáreas de las fincas asociadas a cada dueño, manteniéndolo actualizado para optimizar la generación de reportes. Por su parte, las tablas de cierre de embarque almacenan de forma deliberada una copia de los nombres de dueños, fincas, trabajadores y tareas al momento del cierre, de modo que la liquidación quede congelada como evidencia histórica aunque los catálogos cambien después.

Tabla 4 Tabla t_usuario

Campo	Tipo	Descripción
usu_id	SERIAL (PK)	Identificador único del usuario.
usu_nombre	VARCHAR(100)	Nombre completo del usuario.
usu_usuario	VARCHAR(50)	Nombre de acceso (único).
usu_contrasena	VARCHAR(255)	Contraseña cifrada con BCrypt.
usu_rol	VARCHAR(20)	Rol (ADMIN, MAYORDOMO, FAC- TURADOR, BODEGUERO, CAPI- TAN, DUENO).
usu_estado	BOOLEAN	Indica si el usuario está activo.
usu_fecha_crea	TIMESTAMP	Fecha de creación del usuario.
usu_primer_login	BOOLEAN	Indica si el usuario debe cambiar su contraseña en el primer ingreso.

Tabla 5 Tabla t_dueno

Campo	Tipo	Descripción
due_id	SERIAL (PK)	Identificador único del dueño.
due_nombre	VARCHAR(100)	Nombre del dueño o socio.
due_hectareas	DECIMAL(10,2)	Total de hectáreas del dueño, calculado automáticamente como la suma de las hectáreas de sus fincas.
due_estado	BOOLEAN	Indica si el dueño está activo.

Tabla 6 Tabla t_finca

Campo	Tipo	Descripción
fin_id	SERIAL (PK)	Identificador único de la finca.
fin_due_id	INT (FK)	Dueño propietario de la finca.
fin_nombre	VARCHAR(100)	Nombre de la finca.
fin_ubicacion	VARCHAR(150)	Ubicación o referencia de la finca.
fin_latitud	DECIMAL(10,7)	Coordenada de latitud (GPS).

fin_longitud	DECIMAL(10,7)	Coordenada de longitud (GPS).
fin_hectareas	DECIMAL(10,2)	Número de hectáreas de la finca.
fin_estado	BOOLEAN	Indica si la finca está activa.

Tabla 7 Tabla t_comprador

Campo	Tipo	Descripción
com_id	SERIAL (PK)	Identificador único del comprador.
com_nombre	VARCHAR(100)	Nombre del comprador.
com_estado	BOOLEAN	Indica si el comprador está activo.

Tabla 8 Tabla t_trabajador

Campo	Tipo	Descripción
tra_id	SERIAL (PK)	Identificador único del trabajador.
tra_nombre	VARCHAR(100)	Nombre del trabajador de campo.
tra_estado	BOOLEAN	Indica si el trabajador está activo.

Tabla 9 Tabla t_tarea

Campo	Tipo	Descripción
tar_id	SERIAL (PK)	Identificador único de la tarea.
tar_nombre	VARCHAR(100)	Nombre de la labor (deshoje, enfunde, etc.).
tar_ambito	VARCHAR(15)	Ámbito de la tarea (EMBARQUE, FINCA, AMBOS).

Tabla 10 Tabla t_evaluador

Campo	Tipo	Descripción
eva_id	SERIAL (PK)	Identificador único del evaluador.
eva_nombre	VARCHAR(150)	Nombre del evaluador de la fruta (único).
eva_activo	BOOLEAN	Indica si el evaluador está activo.

Tabla 11 Tabla t_categoria

Campo	Tipo	Descripción
cat_id	SERIAL (PK)	Identificador único de la categoría.

cat_nombre	VARCHAR(80)	Nombre de la categoría (Fertilizante, Químico, Material).
cat_estado	BOOLEAN	Indica si la categoría está activa.

Tabla 12 Tabla t_proveedor

Campo	Tipo	Descripción
prv_id	SERIAL (PK)	Identificador único del proveedor.
prv_nombre	VARCHAR(100)	Nombre del proveedor.
prv_telefono	VARCHAR(20)	Teléfono de contacto.
prv_estado	BOOLEAN	Indica si el proveedor está activo.

Tabla 13 Tabla t_producto

Campo	Tipo	Descripción
pro_id	SERIAL (PK)	Identificador único del producto.
pro_cat_id	INT (FK)	Categoría a la que pertenece.
pro_nombre	VARCHAR(100)	Nombre del producto (urea, fungicida, fundas).
pro_unidad	VARCHAR(20)	Unidad (SACO, LITRO, KG, UNIDAD, CANECA, GALON).
pro_stock_actual	DECIMAL(12,2)	Cantidad total disponible en bodega. El saldo por dueño se calcula a partir de los movimientos
pro_stock_minimo	DECIMAL(12,2)	Nivel mínimo antes de reposición.
pro_estado	BOOLEAN	Indica si el producto está activo.

Tabla 14 Tabla t_movimiento_inventario

Campo	Tipo	Descripción
mov_id	SERIAL (PK)	Identificador del movimiento.
mov_pro_id	INT (FK)	Producto asociado.
mov_tipo	VARCHAR(10)	Tipo (ENTRADA o SALIDA).
mov_cantidad	DECIMAL(12,2)	Cantidad movida.
mov_fecha	TIMESTAMP	Fecha y hora del movimiento.
mov_costo_unit	DECIMAL(10,2)	Costo unitario (en entradas).
mov_caducidad	DATE	Fecha de caducidad del lote (químicos).
mov_prv_id	INT (FK)	Proveedor (en entradas).
mov_due_id	INT (FK)	Dueño al que pertenece el movimiento y que asume el gasto.
mov_usu_id	INT (FK)	Usuario que registró el movimiento.
mov_observacion	VARCHAR(255)	Detalle u observación.

Tabla 15 Tabla t_cinta_semana

Campo	Tipo	Descripción
cin_id	SERIAL (PK)	Identificador del registro.
cin_semana	INT	Número de semana del ciclo de enfunde (1–13).
cin_color	VARCHAR(30)	Color de cinta asignado a esa semana.
cin_estado	BOOLEAN	Indica si el registro está activo.

Tabla 16 Tabla t_embarque

Campo	Tipo	Descripción
emb_id	SERIAL (PK)	Identificador del embarque.
emb_nombre	VARCHAR(100)	Nombre o etiqueta del embarque.
emb_fecha	DATE	Fecha del embarque.
emb_semana_anio	INT	Número de semana del año (1–52).
emb_com_id	INT (FK)	Comprador del banano.
emb_precio_caja	DECIMAL(10,2)	Precio por caja; puede quedar pendiente y asignarse al finalizar.
emb_cajas_solicitadas	DECIMAL(12,2)	Cajas solicitadas por el comprador para ese embarque.
emb_color_cinta	VARCHAR(30)	Color de cinta del embarque (manual o automático).
emb_observacion	VARCHAR(255)	Observación de calidad y racimos retirados.

emb_eva_id	INT (FK)	Evaluador de la fruta del embarque.
emb_evaluador_monto	DECIMAL(14,2)	Monto acordado con el evaluador.
emb_estado_proceso	VARCHAR(20)	Estado del embarque (PROXIMO, EN_PROCESO, FINALIZADO, CANCELADO).-
emb_usu_id	INT (FK)	Usuario que registró el embarque.
emb_estado_sync	VARCHAR(15)	Estado de sincronización (LOCAL, SINCRONIZADO).

Tabla 17 Tabla t_embarque_detalle

Campo	Tipo	Descripción
edt_id	SERIAL (PK)	Identificador del detalle.
edt_emb_id	INT (FK)	Embarque asociado.
edt_due_id	INT (FK)	Dueño que aportó las cajas.
edt_fin_id	INT (FK)	Finca de la que proviene la producción.
edt_cajas	DECIMAL(14,2)	Número de cajas aportadas.
edt_racimos	DECIMAL(14,2)	Número de racimos cortados.
edt_cintas_adicionales	VARCHAR(255)	Colores de cinta adicionales cosechados en el mismo corte.

Tabla 18 Tabla t_cuadrilla

Campo	Tipo	Descripción
cua_id	SERIAL (PK)	Identificador del registro.
cua_emb_id	INT (FK)	Embarque asociado.
cua_tra_id	INT (FK)	Trabajador de la cuadrilla.
cua_tar_id	INT (FK)	Tarea realizada.
cua_pago	DECIMAL(10,2)	Pago del día.
cua_bono	DECIMAL(10,2)	Bono extra opcional.
cua_observacion	VARCHAR(255)	Observación (p. ej. pago parcial).

Tabla 19 Tabla t_cierre_embarque

Campo	Tipo	Descripción
cie_id	SERIAL (PK)	Identificador del cierre.
cie_emb_id	INT (FK, único)	Embarque liquidado.
cie_fecha_cierre	TIMESTAMP	Fecha y hora en que se generó el cierre.
cie_semana_anio	INT	Semana del año del embarque.
cie_precio_caja	DECIMAL(14,2)	Precio por caja definitivo.
cie_total_cajas	DECIMAL(14,2)	Total de cajas del embarque.

Campo	Tipo	Descripción
cie_total_racimos	DECIMAL(14,2)	Total de racimos cortados.
cie_ingreso_bruto	DECIMAL(14,2)	Ingreso bruto (cajas por precio).
cie_total_cuadrilla	DECIMAL(14,2)	Costo total de la cuadrilla (pagos y bonos).
cie_total_gastos	DECIMAL(14,2)	Suma de los gastos vinculados al embarque.
cie_total_inventario	DECIMAL(14,2)	Costo de inventario imputado al embarque.
cie_eva_id	INT (FK)	Evaluador del embarque.
cie_evaluador_nombre	VARCHAR(150)	Nombre del evaluador al momento del cierre.
cie_total_evaluador	DECIMAL(14,2)	Monto pactado con el evaluador.
cie_estado_pago_evaluador	VARCHAR(20)	Estado del pago al evaluador (PENDIENTE, PAGADO).
cie_evaluador_pagado_en	TIMESTAMP	Fecha y hora del pago al evaluador.
cie_utilidad_estimada	DECIMAL(14,2)	Utilidad estimada del embarque.
cie_monto_reliquidacion	DECIMAL(14,2)	Monto de la reliquidación posterior, si aplica.
cie_estado	VARCHAR(20)	Estado del cierre.
cie_usu_id	INT (FK)	Usuario que generó el cierre.

Tabla 20 Tabla t_cierre_embarque_detalle

Campo	Tipo	Descripción
ced_id	SERIAL (PK)	Identificador del detalle de cierre.
ced_cie_id	INT (FK)	Cierre asociado.
ced_due_id	INT (FK)	Dueño liquidado.
ced_fin_id	INT (FK)	Finca de origen de la producción.
ced_dueno_nombre	VARCHAR(100)	Nombre del dueño al momento del cierre.
ced_finca_nombre	VARCHAR(100)	Nombre de la finca al momento del cierre.
ced_cajas	DECIMAL(14,2)	Cajas aportadas.
ced_racimos	DECIMAL(14,2)	Racimos cortados.
ced_ratio_conversion	DECIMAL(14,4)	Ratio de conversión del dueño.
ced_precio_caja	DECIMAL(14,2)	Precio por caja aplicado.
ced_ingreso_bruto	DECIMAL(14,2)	Ingreso bruto del dueño.
ced_porcentaje_produccion	DECIMAL(10,6)	Proporción de cajas del dueño frente al total.
ced_costo_cuadrilla	DECIMAL(14,2)	Porción del costo de cuadrilla que asume.
ced_costo_gastos	DECIMAL(14,2)	Porción de gastos y evaluador que asume.

Campo	Tipo	Descripción
ced_costo_inventario	DECIMAL(14,2)	Porción de costo de inventario, si aplica.
ced_utilidad_estimada	DECIMAL(14,2)	Utilidad estimada del dueño en el embarque.

Tabla 21 Tabla t_cierre_embarque_cuadrilla

Campo	Tipo	Descripción
cec_id	SERIAL (PK)	Identificador de la liquidación.
cec_cie_id	INT (FK)	Cierre asociado.
cec_cua_id	INT (FK)	Registro de cuadrilla de origen.
cec_tra_id	INT (FK)	Trabajador liquidado.
cec_tar_id	INT (FK)	Tarea realizada.
cec_trabajador_nombre	VARCHAR(100)	Nombre del trabajador al cierre.
cec_tarea_nombre	VARCHAR(100)	Nombre de la tarea al cierre.
cec_pago_base	DECIMAL(14,2)	Pago base acordado.
cec_bono	DECIMAL(14,2)	Bono acumulado.
cec_descuento	DECIMAL(14,2)	Descuentos aplicados.
cec_adelanto	DECIMAL(14,2)	Adelantos aplicados.
cec_total_pagar	DECIMAL(14,2)	Total a pagar al trabajador.
cec_observacion	VARCHAR(255)	Observación del pago.
cec_estado_pago	VARCHAR(20)	Estado del pago (PENDIENTE, PAGADO).
cec_pagado_en	TIMESTAMP	Fecha y hora en que se pagó.

Tabla 22 Tabla t_gasto

Campo	Tipo	Descripción
gas_id	SERIAL (PK)	Identificador del gasto.
gas_tipo	VARCHAR(20)	Tipo (EMBARQUE, FINCA, COMPRA_PRODUCTO, SERVICIO).
gas_descripcion	VARCHAR(255)	Descripción del gasto.
gas_monto	DECIMAL(12,2)	Valor del gasto.
gas_fecha	DATE	Fecha del gasto.
gas_due_id	INT (FK)	Dueño asignado, cuando aplica.
gas_emb_id	INT (FK)	Embarque asociado (en gastos de embarque).
gas_pro_id	INT (FK)	Producto comprado (en compras de producto).
gas_prv_id	INT (FK)	Proveedor del gasto.
gas_cantidad	DECIMAL(14,3)	Cantidad comprada.
gas_unidad	VARCHAR(40)	Unidad de la cantidad.

gas_precio_unitario	DECIMAL(14,4)	Precio unitario.
gas_numero_factura	VARCHAR(100)	Número de la factura del proveedor.
gas_estado_factura	VARCHAR(30)	Estado del comprobante interno (LISTA_PARA_EMITIR, EMITIDA).
gas_usu_id	INT (FK)	Usuario que registró el gasto.

Tabla 23 Tabla t_jornal

Campo	Tipo	Descripción
jor_id	SERIAL (PK)	Identificador del jornal.
jor_fecha	DATE	Día trabajado.
jor_tra_id	INT (FK)	Trabajador.
jor_tar_id	INT (FK)	Labor realizada.
jor_pago	DECIMAL(10,2)	Pago del día por la tarea.
jor_compartido	BOOLEAN	Indica si la labor se reparte entre dueños.
jor_estado_pago	VARCHAR(15)	Estado del pago (PENDIENTE, PAGADO).
jor_realizado	BOOLEAN	Indica si el jornal fue efectivamente completado.
jor_adelanto	DECIMAL(10,2)	Monto adelantado, si aplica.
jor_due_id	INT (FK)	Dueño al que se asigna el jornal, cuando no es compartido.
jor_fincas	VARCHAR(255)	Fincas en las que se realizó la labor.
jor_usu_id	INT (FK)	Usuario que registró el jornal.

Tabla 24 Tabla t_jornal_dueno

Campo	Tipo	Descripción
jod_id	SERIAL (PK)	Identificador del reparto.
jod_jor_id	INT (FK)	Jornal asociado.
jod_due_id	INT (FK)	Dueño al que se carga parte del jornal.
jod_monto	DECIMAL(10,2)	Porción del pago asignada al dueño.

Tabla 25 Tabla t_pago_jornal_semanal

Campo	Tipo	Descripción
pjs_id	SERIAL (PK)	Identificador del pago semanal.
pjs_tra_id	INT (FK)	Trabajador pagado. Único por año y semana.
pjs_anio	INT	Año de la planilla.
pjs_semana	INT	Semana del año de la planilla.

pjs_fecha_inicio	DATE	Lunes de la semana liquidada.
pjs_fecha_fin	DATE	Fin de la semana liquidada.
pjs_fecha_programada	DATE	Fecha programada de pago (sábado).
pjs_total_jornales	DECIMAL(14,2)	Suma de los jornales de la semana.
pjs_total_adelantos	DECIMAL(14,2)	Suma de los adelantos entregados.
pjs_total_pagar	DECIMAL(14,2)	Total a pagar (jornales menos adelantos más bono).
pjs_bono	DECIMAL(14,2)	Bono adicional otorgado en el pago.
pjs_estado	VARCHAR(20)	Estado del pago semanal.
pjs_fecha_pago	DATE	Fecha efectiva del pago.
pjs_pagado_en	TIMESTAMP	Fecha y hora del registro del pago.
pjs_metodo_pago	VARCHAR(30)	Método de pago utilizado.
pjs_referencia_pago	VARCHAR(255)	Referencia o comprobante del pago.
pjs_observacion	VARCHAR(255)	Observación del pago.
pjs_usu_id	INT (FK)	Usuario que registró el pago.
pjs_creado_en	TIMESTAMP	Fecha de creación del registro.

Tabla 26 Tabla t_facturador_finca

Campo	Tipo	Descripción
ffi_id	SERIAL (PK)	Identificador de la asignación.
ffi_usu_id	INT (FK)	Usuario con rol facturador.
ffi_fin_id	INT (FK)	Finca a la que tiene acceso.
ffi_asignado_en	TIMESTAMP	Fecha de la asignación.

Tabla 27 Tabla t_mayordomo_dueno

Campo	Tipo	Descripción
mdu_id	SERIAL (PK)	Identificador de la asignación.
mdu_usu_id	INT (FK)	Usuario con rol mayordomo.
mdu_due_id	INT (FK)	Dueño cuyas operaciones puede gestionar.
mdu_asignado_en	TIMESTAMP	Fecha de la asignación.

Tabla 28 Tabla t_auditoria

Campo	Tipo	Descripción
aud_id	SERIAL (PK)	Identificador del registro de auditoría.
aud_usu_id	INT (FK)	Usuario que realizó la acción.
aud_accion	VARCHAR(20)	Acción (CREAR, EDITAR, ELIMINAR, FINALIZAR, LOGIN, SYNC).
aud_tabla	VARCHAR(50)	Tabla afectada.

aud_registro_id	INT	Identificador del registro afectado.
aud_detalle	VARCHAR(500)	Descripción legible de la acción.
aud_fecha	TIMESTAMP	Fecha y hora de la acción.

Tabla 29 Tabla t_precio_banano_historico

Campo	Tipo	Descripción
id	SERIAL (PK)	Identificador único del registro histórico del precio del banano.
diferencia	DECIMAL(10,2)	Variación del precio con respecto al registro anterior.
fecha_actualizacion	DATE	Fecha de la última actualización informada por la fuente consultada.
fecha_consulta	DATE	Fecha en la que el sistema consultó o capturó la información.
fecha_creacion	DATE	Fecha en la que el registro fue almacenado en la base de datos.
fecha_precio	DATE	Fecha a la que corresponde el precio registrado.
fuelle	VARCHAR(100)	Nombre de la fuente de la cual se obtuvo la información.
moneda	VARCHAR(10)	Moneda en la que se expresa el precio, por ejemplo, USD.
precio_spot	DECIMAL(10,2)	Precio spot o precio de referencia registrado para la caja de banano.
tendencia	VARCHAR(20)	Comportamiento del precio con respecto al registro anterior: al alza, a la baja o estable.
tipo_caja	VARCHAR(50)	Tipo o presentación de la caja de banano a la que corresponde el precio.

Desarrollo e Implementación del Sistema

1.24. Especificación de Requisitos de Software (SRS)

La presente sección documenta la especificación de requisitos de software del sistema siguiendo la estructura recomendada por el estándar IEEE 830 [26]: propósito, alcance, descripción general, requisitos funcionales, requisitos no funcionales e historias de usuario. La especificación consolida tanto los requerimientos identificados durante el levantamiento inicial como aquellos incorporados de forma incremental a partir de la retroalimentación del propietario en las revisiones de cada sprint.

1.24.1 Introducción del SRS

Propósito. El propósito de este documento es definir de manera detallada los requisitos funcionales y no funcionales del prototipo de sistema ERP móvil para la Hacienda Bananera San Diego, sirviendo de referencia para su desarrollo y validación.

Alcance del producto. El sistema, denominado prototipo ERP móvil, permite gestionar el ciclo completo del embarque (desde su planificación hasta su cierre financiero), el inventario de productos de bodega con existencias por dueño, los gastos, la mano de obra con su planilla semanal, la emisión de comprobantes internos en PDF y la generación de reportes, con soporte para operación sin conexión y con seis roles de acceso diferenciados. Queda fuera del alcance la facturación electrónica homologada ante el SRI y la liquidación contable completa entre socios.

Definiciones, acrónimos y abreviaturas. ERP: Enterprise Resource Planning; SRS: Software Requirements Specification; API: Application Programming Interface; REST: Representational State Transfer; JWT: JSON Web Token; SUS: System Usability Scale; embarque: venta semanal de banano; cierre de embarque: liquidación financiera generada al finalizar un embarque; ratio: cociente entre cajas y racimos cortados; evaluador: persona que verifica la calidad de la fruta el día del embarque; planilla semanal: consolidado de los jornales de un trabajador entre lunes y sábado.

1.24.2 Descripción general

Perspectiva del producto. El sistema es un producto nuevo, independiente, compuesto por una aplicación móvil cliente y un servicio backend con base de datos, comunicados mediante una API REST.

Funciones del producto. De manera general, el sistema permite la autenticación con control de roles, la gestión de usuarios y catálogos, el registro de embarques con producción por dueño y por finca, la finalización del embarque con conciliación de la producción real y generación automática del cierre financiero, el control de inventario con saldos por dueño, la gestión de gastos y de la mano de obra con su planilla semanal, la emisión de comprobantes en PDF, la auditoría de operaciones y la generación de reportes.

Características de los usuarios. Los usuarios poseen perfiles técnicos variados y, en general, limitada experiencia con herramientas digitales, por lo que la interfaz debe ser sencilla e intuitiva. Los roles previstos son administrador, mayordomo, facturador, bodeguero, capitán y dueño; este último corresponde a los socios de la hacienda, que acceden únicamente a la información de sus propias fincas.

Restricciones. El sistema debe operar en dispositivos Android e iOS, funcionar con conectividad limitada mediante un modo sin conexión, restringir la visibilidad de la información financiera según el rol del usuario y garantizar la integridad de los datos mediante una base de datos relacional.

Suposiciones y dependencias. Se asume que los usuarios disponen de un dispositivo móvil compatible y que la sincronización se realiza cuando exista conexión a internet disponible.

1.24.3 Requisitos funcionales

Tabla 30 Requisitos funcionales del sistema

Código	Descripción
RF01	El sistema permitirá la autenticación de usuarios mediante usuario y contraseña, emitiendo un token JWT para las solicitudes posteriores.
RF02	El sistema permitirá gestionar usuarios y asignarles un rol (administrador, mayordomo, facturador, bodeguero, capitán, dueño).
RF03	El sistema obligará al usuario a cambiar su contraseña en su primer inicio de sesión.
RF04	El sistema permitirá registrar y administrar a los dueños de la finca, y para cada dueño registrar una o varias fincas con su nombre, ubicación, coordenadas GPS y número de hectáreas.
RF05	El sistema permitirá al usuario con rol dueño administrar sus propias fincas, recalculando automáticamente el total de hectáreas del dueño.
RF06	El sistema permitirá gestionar categorías de productos (fertilizantes, químicos, materiales).
RF07	El sistema permitirá registrar productos de bodega indicando su categoría, unidad de medida y stock mínimo.
RF08	El sistema permitirá registrar proveedores y evaluadores de fruta.
RF09	El sistema permitirá registrar entradas de productos al inventario, incluyendo costo, proveedor, dueño responsable y fecha de caducidad del lote.
RF10	El sistema permitirá registrar salidas de productos del inventario asociadas a un dueño, impidiendo que el saldo quede negativo.
RF11	El sistema calculará el stock disponible de cada producto por dueño a partir de sus movimientos, y actualizará el stock global tras cada operación.
RF12	El sistema alertará cuando un producto alcance su stock mínimo o cuando un lote esté próximo a caducar.
RF13	El sistema permitirá gestionar un calendario de cintas que asocie cada semana del ciclo de enfunde (1 a 13) con un color.

Código	Descripción
RF14	El sistema permitirá registrar un embarque indicando nombre, fecha, semana del año, comprador, cajas solicitadas y, opcionalmente, el precio por caja, que puede definirse más adelante.
RF15	El sistema asignará el color de cinta del embarque de forma automática según la semana del año y el ciclo de trece semanas, permitiendo también su ingreso manual.
RF16	El sistema permitirá registrar, dentro de un embarque, la cantidad de cajas y de racimos cortados aportados por cada dueño desde cada finca, validando que la finca pertenezca al dueño indicado, con la posibilidad de anotar cintas adicionales.
RF17	El sistema calculará automáticamente el ingreso de cada dueño según sus cajas y el precio acordado.
RF18	El sistema calculará automáticamente el ratio de conversión de cada dueño dividiendo sus cajas entre sus racimos cortados, acotado a un máximo de 2 para evitar valores atípicos.
RF19	El sistema permitirá registrar una observación del embarque, incluyendo los racimos retirados por no cumplir el estándar de calidad.
RF20	El sistema permitirá registrar la cuadrilla del embarque, con el trabajador, la tarea, el pago, un bono opcional y observaciones.
RF21	El sistema gestionará el estado del proceso de cada embarque (Próximo, En proceso, Finalizado, Cancelado).
RF22	El sistema permitirá finalizar un embarque conciliando la producción real de cada finca, registrando el evaluador con su monto y el precio por caja definitivo; un embarque cancelado no podrá finalizarse.
RF23	Al finalizar un embarque, el sistema generará automáticamente su cierre financiero: ingreso bruto, costo de cuadrilla, gastos, monto del evaluador y utilidad estimada, tanto del embarque como de cada dueño.
RF24	El sistema repartirá los costos del embarque (cuadrilla, gastos y evaluador) entre los dueños en proporción a las cajas que aportó cada uno.
RF25	El sistema permitirá registrar el pago a cada trabajador de la cuadrilla desde el cierre del

Código	Descripción
	embarque, con la posibilidad de añadir un bono extra, controlando su estado de pago.
RF26	El sistema permitirá registrar el pago al evaluador del embarque y controlar su estado.
RF27	El sistema permitirá ajustar el precio por caja de un cierre y recalcular los ingresos y utilidades, así como registrar el monto de una reliquidación posterior del embarque.
RF28	El sistema permitirá registrar gastos de cuatro tipos: de embarque, de finca, compra de producto y servicio, indicando dueño, proveedor, cantidad, precio unitario y número de factura cuando corresponda.
RF29	Cuando el gasto sea una compra de producto, el sistema generará automáticamente la entrada correspondiente en el inventario del dueño.
RF30	El sistema permitirá registrar el jornal diario del personal semanal, indicando trabajador, tarea, pago, fecha, dueño o dueños responsables y las fincas donde se realizó la labor.
RF31	El sistema permitirá asignar un jornal a varios dueños cuando la labor sea compartida, repartiendo el pago entre ellos.
RF32	El sistema permitirá registrar adelantos de pago a los trabajadores y llevar el control de lo pendiente.
RF33	El sistema permitirá marcar cada jornal como realizado y clasificará su verificación como hecho, pendiente o pendiente de verificar cuando la semana haya concluido sin confirmación.
RF34	El sistema consolidará los jornales de cada trabajador en una planilla semanal (lunes a sábado), calculando el total a pagar como los jornales menos los adelantos más un bono opcional, con pago programado para el sábado.
RF35	El sistema permitirá registrar el pago de la planilla semanal indicando método, referencia y observación, marcando como pagados los jornales incluidos.
RF36	El sistema generará comprobantes internos en formato PDF del cierre de embarque y de la planilla semanal, que podrán imprimirse o compartirse mediante aplicaciones de mensajería como WhatsApp, controlando su estado de emisión.

Código	Descripción
RF37	El sistema permitirá al administrador asignar fincas a los usuarios facturadores y dueños a los usuarios mayordomos, restringiendo su acceso a la información asignada.
RF38	El sistema ocultará los ingresos y ganancias de los dueños a los usuarios con rol mayordomo.
RF39	El usuario con rol dueño solo podrá consultar los embarques, cierres, gastos, jornales, inventario y reportes vinculados a sus propias fincas.
RF40	El sistema registrará el usuario responsable de cada operación (embarque, gasto, movimiento de inventario, jornal y pagos).
RF41	El sistema mantendrá una bitácora de auditoría de las acciones realizadas por los usuarios.
RF42	El sistema permitirá operar sin conexión a internet, almacenando los registros de embarque localmente en el dispositivo y sincronizándolos con el servidor al recuperar la conexión.
RF43	El sistema permitirá programar recordatorios locales en el dispositivo para la verificación de los jornales del día.
RF44	El sistema permitirá registrar las coordenadas GPS de cada finca mediante un mapa interactivo, pudiendo tomar la ubicación actual del dispositivo.
RF45	El sistema generará reportes de ventas por dueño (resumidos y con detalle por embarque), gastos por tipo y dueño, mano de obra semanal, estado del inventario (productos bajo stock mínimo y próximos a caducar) y productividad por dueño y embarque con cajas, racimos y ratio de conversión.
RF46	El sistema identificará, a partir del cierre, a los dueños cuyos costos superan su ingreso en un embarque, mostrando el monto que deben aportar.

1.24.4 Requisitos no funcionales

Tabla 31 Requisitos no funcionales del sistema

Código	Atributo	Descripción
RNF01	Seguridad	El acceso al sistema se controlará mediante autenticación con JWT, contraseñas cifradas con BCrypt y permisos verificados en cada endpoint según el rol del usuario.
RNF02	Usabilidad	La interfaz será intuitiva y adecuada para usuarios con poca experiencia tecnológica, alcanzando un mínimo de 68 puntos en la escala SUS.
RNF03	Disponibilidad	El sistema permitirá registrar operaciones críticas sin conexión a internet.
RNF04	Rendimiento	Las operaciones de registro y consulta responderán en un tiempo razonable bajo condiciones normales de uso; no obstante, no se realizaron pruebas formales de rendimiento ni de carga.
RNF05	Compatibilidad	La aplicación funcionará en dispositivos Android e iOS; la validación experimental de la versión evaluada se realizó en dispositivos Android.
RNF06	Mantenibilidad	El sistema seguirá el patrón MVC y una arquitectura en capas que facilite su mantenimiento y escalabilidad, con la API documentada mediante OpenAPI.
RNF07	Integridad	La base de datos garantizará la consistencia de los datos mediante transacciones ACID y restricciones de integridad referencial.
RNF08	Trazabilidad	El sistema permitirá identificar qué usuario realizó cada operación mediante el registro de auditoría.
RNF09	Confidencialidad	La información financiera será visible únicamente para los roles autorizados; cada dueño accederá solo a sus propios datos.

1.24.5 Historias de usuario

Tabla 32 Historias de usuario

ID	Rol	Historia
HU01	Administrador	Como administrador quiero gestionar usuarios y roles para controlar quién accede al sistema.
HU02	Administrador	Como administrador quiero registrar a los dueños con sus fincas, hectáreas y ubicación para diferenciar su producción y sus gastos.
HU03	Administrador	Como administrador quiero gestionar categorías, productos, proveedores y evaluadores para organizar los catálogos de la hacienda.
HU04	Administrador	Como administrador quiero gestionar el calendario de cintas para que el color se asigne automáticamente según la semana.
HU05	Administrador	Como administrador quiero asignar fincas a los facturadores y dueños a los mayordomos para que cada uno vea solo lo que le corresponde.
HU06	Administrador	Como administrador quiero ver los reportes de ventas, gastos, mano de obra y productividad para conocer los resultados de la finca.
HU07	Administrador	Como administrador quiero consultar la bitácora de auditoría para saber qué usuario realizó cada operación.
HU08	Facturador	Como facturador quiero registrar un embarque con las cajas y los racimos de cada dueño y finca para calcular cuánto recibe y su ratio.
HU09	Facturador	Como facturador quiero finalizar el embarque conciliando la producción real, el evaluador y el precio definitivo para que se genere el cierre financiero.
HU10	Facturador	Como facturador quiero registrar los pagos a la cuadrilla y al evaluador desde el cierre, y emitir el comprobante en PDF para compartirlo por WhatsApp.
HU11	Facturador	Como facturador quiero registrar la reliquidación del embarque cuando el comprador entrega un valor adicional.
HU12	Capitán	Como capitán quiero registrar a la cuadrilla del embarque con su tarea y pago para controlar el costo del día.
HU13	Bodeguero	Como bodeguero quiero registrar las entradas y salidas de productos por dueño para mantener el inventario actualizado.
HU14	Bodeguero	Como bodeguero quiero ser alertado cuando un producto llegue al stock mínimo o esté próximo a caducar.
HU15	Mayordomo	Como mayordomo quiero registrar ventas y gastos de los dueños que se me asignen, sin acceder a sus ganancias.
HU16	Mayordomo	Como mayordomo quiero registrar el jornal diario del personal y marcarlo como realizado para controlar la mano de obra de la semana.
HU17	Mayordomo	Como mayordomo quiero pagar la planilla semanal de cada trabajador descontando los adelantos para cerrar la semana sin errores.
HU18	Dueño	Como dueño quiero iniciar sesión y ver únicamente mis embarques, cierres, gastos e inventario para conocer el estado de mis fincas.

HU19	Dueño	Como dueño quiero administrar mis fincas con su ubicación en el mapa para mantener mi información al día.
HU20	Dueño	Como dueño quiero saber cuánto me corresponde de cada embarque y cuánto debo aportar cuando los gastos superan mi ingreso.
HU21	Administrador	Como administrador quiero que el sistema funcione sin internet en el campo y sincronice después para no perder registros.
HU22	Usuario	Como usuario quiero cambiar mi contraseña en el primer ingreso para proteger mi cuenta.
HU23	Facturador	Como facturador, quiero consultar los comprobantes internos de los dueños que me hayan autorizado para gestionar la documentación asignada sin acceder a información de otros propietarios.

Historias de usuario detalladas de los roles operativos

Para precisar el alcance de los roles conforme a la implementación, se detallan tres historias representativas con su estructura completa. Se incorpora la historia HU23, que explicita el acceso del facturador restringido a los dueños que lo hayan autorizado; las historias HU13 (bodeguero) y HU15 (mayordomo), ya registradas en la Tabla 32, se presentan con su estructura ampliada. Los valores de prioridad, Story Points y sprint se corresponden con el Product Backlog consolidado (Anexo D), y el estado «Implementada» refleja su disponibilidad verificada en el backend y el frontend.

HU23: Facturador

Como facturador, quiero consultar los comprobantes internos de los dueños que me hayan autorizado para gestionar la documentación asignada sin acceder a información de otros propietarios. Prioridad: Media. Story Points: 5. Sprint: 3. Estado: Implementada.

Criterios de aceptación: el facturador solo visualiza comprobantes y documentos internos de los dueños que lo han autorizado mediante una asignación válida; el sistema impide el acceso a documentos de dueños no autorizados; y los documentos generados son comprobantes internos de respaldo, no facturas electrónicas autorizadas por el SRI.

HU13: Bodeguero

Como bodeguero, quiero registrar las entradas y salidas de productos para mantener actualizado el inventario correspondiente a cada dueño. Prioridad: Alta. Story Points: 5. Sprint: 2. Estado: Implementada.

Criterios de aceptación: cada movimiento actualiza automáticamente el saldo del producto; se distingue el inventario por dueño; el sistema alerta cuando un producto alcanza el stock mínimo o se aproxima a su caducidad; y el bodeguero no accede a cierres ni información financiera privada.

HU15: Mayordomo

Como mayordomo, quiero registrar jornales, gastos y operaciones de los dueños asignados para controlar las actividades de la finca sin visualizar sus ganancias. Prioridad: Alta. Story Points: 8. Sprint: 3. Estado: Implementada.

Criterios de aceptación: el mayordomo solo opera con los dueños que le fueron asignados; puede registrar jornales, gastos y operaciones autorizadas; el sistema oculta ingresos, ganancias y utilidades de los dueños; y no permite gestionar información de dueños no asignados.

1.24.6 Diagrama de casos de uso

El comportamiento del sistema desde la perspectiva de sus actores se resume en el diagrama general de casos de uso de la Figura 9, que relaciona cada rol (administrador, facturador, bodeguero, capitán, mayordomo y dueño) con las funcionalidades que tiene permitido ejecutar. Para evitar que el diagrama principal resulte ilegible, los casos de uso detallados de embarques, inventario y mano de obra se incluyen en el Anexo M.

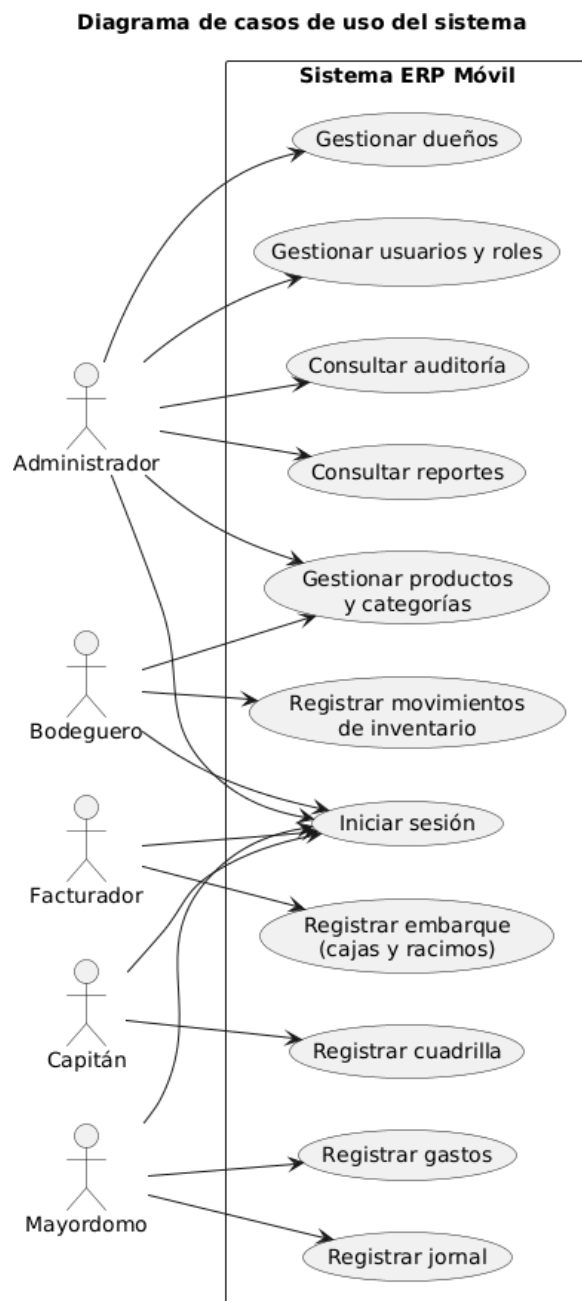


Figura 10 Diagrama de casos de uso del sistema

1.25. Implementación del Sistema

1.25.1 Ambiente de desarrollo

El desarrollo se realizó en un equipo personal con los siguientes componentes: JDK 17 y Maven para el backend, Node.js y el CLI de Expo para el frontend, PostgreSQL como motor de base de datos local, Visual Studio Code e IntelliJ IDEA como entornos de edición, Postman y la interfaz de Swagger para probar la API, y Git con GitLab para el control de versiones, con el código organizado en dos proyectos dentro de un grupo institucional privado: sanDiego_Back para la API

REST y sanDiego_Front para la aplicación móvil. Las pruebas en dispositivo se ejecutaron mediante la aplicación Expo Go sobre teléfonos Android físicos, complementadas con el emulador de Android Studio.

1.25.2 Estructura del backend

El backend es un proyecto Maven de Spring Boot organizado en paquetes que reflejan la arquitectura en capas: controller expone los endpoints REST; service concentra la lógica de negocio; repository define las interfaces de acceso a datos con Spring Data JPA; entity contiene las veintiséis entidades mapeadas a las tablas de PostgreSQL junto con sus enumeraciones de rol, tipo de gasto, tipo de movimiento, estado de pago, ámbito de tarea y estado de sincronización; dto agrupa los objetos de transferencia organizados por módulo; security implementa la autenticación con JWT; exception centraliza el manejo de errores de negocio y de recursos no encontrados mediante un manejador global; y config incluye la configuración de CORS, la documentación OpenAPI, la política de seguridad y una clase de inicialización que, al primer arranque, crea los usuarios de cada rol y siembra el calendario de cintas con sus trece colores.

El modelo físico de datos está compuesto por 27 tablas. De ellas, 26 tienen su entidad JPA correspondiente en el backend; la restante, t_precio_banano_historico, es una tabla complementaria que almacena el histórico de precios y se consulta directamente, sin entidad asociada.

Los principales controladores corresponden a autenticación, catálogos, embarques, cierre de embarque, gastos, inventario, jornales, pago semanal de jornales, asignaciones de facturador y mayordomo, fincas del dueño, personal, evaluadores, reportes, auditoría y sincronización.

1.25.3 Seguridad y control de acceso

La seguridad se implementó con Spring Security y JWT. Al iniciar sesión, el servidor valida las credenciales, registra el evento en la auditoría y emite un token firmado que la aplicación móvil adjunta en el encabezado de cada solicitud; un filtro lo verifica antes de permitir el acceso. Las contraseñas se almacenan cifradas con BCrypt y cada endpoint declara los roles autorizados mediante anotaciones @PreAuthorize, de modo que la restricción se aplica en el servidor y no depende de la interfaz.

Sobre esa base se añadieron dos niveles adicionales de control. El primero es el filtrado de datos por usuario: cuando quien consulta es un dueño, los servicios devuelven únicamente los embarques, cierres, gastos, jornales y movimientos

vinculados a sus fincas; cuando es un mayordomo, solo la información de los dueños que el administrador le asignó, ocultando además los ingresos, ganancias o utilidades privadas; y cuando es un facturador, únicamente los comprobantes y documentos internos de las fincas y dueños que le fueron autorizados mediante la asignación registrada en `t_facturador_finca`, sin acceso a la información de otros propietarios. El segundo es el flujo de primer inicio de sesión: si el usuario fue creado recientemente, la aplicación lo dirige a una pantalla donde debe establecer una nueva contraseña antes de continuar.

1.25.4 Ciclo de vida del token de sesión implementado

En coherencia con lo expuesto en el apartado 2.7.9, se documenta a continuación la configuración adoptada en el prototipo para las cuatro decisiones que definen el ciclo de vida de un token de sesión.

Expiración. El token de acceso se emite con una validez de veinticuatro horas contadas desde el inicio de sesión. La duración se fijó atendiendo al patrón de uso de la hacienda, donde el registro se concentra en la jornada de embarque y una expiración corta obligaría a reautenticar en pleno campo, con conectividad intermitente.

Renovación. No se implementó un token de refresco. Si el token caduca mientras la aplicación no está en uso, la sesión se cierra y el usuario debe autenticarse de nuevo con sus credenciales. La consecuencia práctica es que la sesión no se prolonga de forma indefinida, pero también que la reautenticación depende de disponer de conexión en ese momento.

Invalidación. Por su naturaleza autocontenida, el token firmado conserva su validez hasta cumplir las veinticuatro horas. El prototipo no mantiene una lista de tokens revocados ni un identificador de versión de credenciales, de modo que dar de baja a un usuario o cambiar su contraseña no invalida de inmediato un token ya emitido. Esta es una limitación conocida del diseño y no un comportamiento deseado: en un entorno de producción, donde la baja de un mayordomo o de un facturador debe surtir efecto al instante, corresponde incorporar un registro de revocación en el servidor.

Almacenamiento en el dispositivo. El token se conserva mediante `AsyncStorage`, el almacén clave-valor de React Native, que no aplica cifrado. Contrastado con las prácticas derivadas del listado OWASP Mobile Top 10 [32], que recomiendan el uso del almacén cifrado que provee el sistema operativo, esta decisión constituye la principal debilidad de la implementación: en un dispositivo comprometido o con acceso de superusuario, el token queda expuesto. La migración a `SecureStore` de Expo, que emplea `Keystore` en Android, es una tarea de bajo esfuerzo y alta prioridad para la puesta en producción.

El balance de estas cuatro decisiones es coherente con la condición de prototipo del sistema: los mecanismos implementados son suficientes para la validación funcional realizada, y las carencias identificadas en la invalidación anticipada y en el almacenamiento del token quedan registradas como requisitos previos a un despliegue real, según se recoge en las recomendaciones del apartado 7.2.

1.25.5 Módulo de inventario

El módulo de inventario administra las categorías, los productos y sus movimientos. Cada entrada registra el costo unitario, el proveedor, la fecha de caducidad del lote y el dueño que asume la compra; cada salida descuenta del saldo del dueño correspondiente.

El saldo de un producto para un dueño no se guarda como un número suelto: se calcula sumando sus entradas y restando sus salidas, lo que evita inconsistencias, y el sistema rechaza cualquier salida que dejaría el saldo en negativo. Las alertas de stock mínimo y de lotes próximos a caducar se exponen como reportes propios del módulo.

1.25.6 Módulo de embarques y cierre financiero

El embarque atraviesa un ciclo de estados: nace como *Próximo*, pasa a *En proceso* el día de su ejecución y termina en *Finalizado* o *Cancelado*. Al crearlo se registran el comprador, la semana del año, las cajas solicitadas y la producción estimada por dueño y finca; el precio por caja puede quedar pendiente, porque en la práctica suele conocerse recién al final de la jornada. El color de cinta se asigna automáticamente a partir de la semana del año y del ciclo de trece semanas del calendario de enfunde, como muestra el Código 1, aunque el usuario puede fijarlo manualmente.

```
private String resolveColor(EmbarqueRequest request) {
    if (request.colorCinta() != null && !request.colorCinta().isBlank()) {
        return request.colorCinta();
    }
    int semanaCinta = ((request.semanaAnio() - 1) % 13) + 1;
    return cintaSemanaRepository.findBySemanaAndEstadoTrue(semanaCinta)
        .orElseThrow(() -> new ResourceNotFoundException("No existe cinta activa para la semana " + semanaCinta))
        .getColor();
}
```

Código 1 Asignación automática del color de cinta según la semana del año

Al finalizar el embarque, el facturador concilia la producción real de cada finca, indica el evaluador con su monto y el precio definitivo, y el sistema genera de forma automática el cierre financiero: calcula el ingreso bruto, el costo total de la cuadrilla (pagos y bonos), los gastos vinculados al embarque y el monto del evaluador, y obtiene la utilidad estimada. Ese total de costos se distribuye entre los dueños en proporción a las cajas que aportó cada uno, respetando la regla de negocio levantada en la entrevista: quien aporta más cajas asume una parte proporcionalmente mayor del costo del día. El fragmento del Código 2 muestra el cálculo del reparto por detalle.

```
BigDecimal proporcion = totalCajas.signum() == 0
    ? BigDecimal.ZERO
    : nz(detalle.getCajas()).divide(
        totalCajas, 6, RoundingMode.HALF_UP);

BigDecimal costoCuadrilla = totalCuadrilla
    .multiply(proporcion)
    .setScale(2, RoundingMode.HALF_UP);

BigDecimal costoGastos = totalGastos
    .add(totalEvaluador)
    .multiply(proporcion)
    .setScale(2, RoundingMode.HALF_UP);

BigDecimal ingresoDetalle = nz(detalle.getCajas())
    .multiply(precioCaja);

BigDecimal utilidadDetalle = ingresoDetalle
    .subtract(costoCuadrilla)
    .subtract(costoGastos);
```

Código 2 Reparto proporcional de costos en el cierre del embarque

El cierre almacena una copia del nombre del dueño, de la finca, del trabajador y de la tarea al momento de generarse, de manera que la liquidación quede congelada como

evidencia aunque los catálogos se modifiquen después. Desde el cierre, el facturador registra el pago de cada trabajador de la cuadrilla (pudiendo añadir un bono extra) y el pago al evaluador, cada uno con su estado y su marca de tiempo. Si más adelante el comprador entrega un valor adicional por el embarque, el sistema permite registrar esa reliquidación sobre el mismo cierre, y también recalculan todo el cierre cuando el precio por caja definitivo cambia.

La Figura 10 representa el ciclo de estados del embarque. Un registro cancelado no puede finalizarse y un embarque finalizado no puede generar un segundo cierre. La reliquidación se aplica sobre el cierre existente sin crear un nuevo embarque.

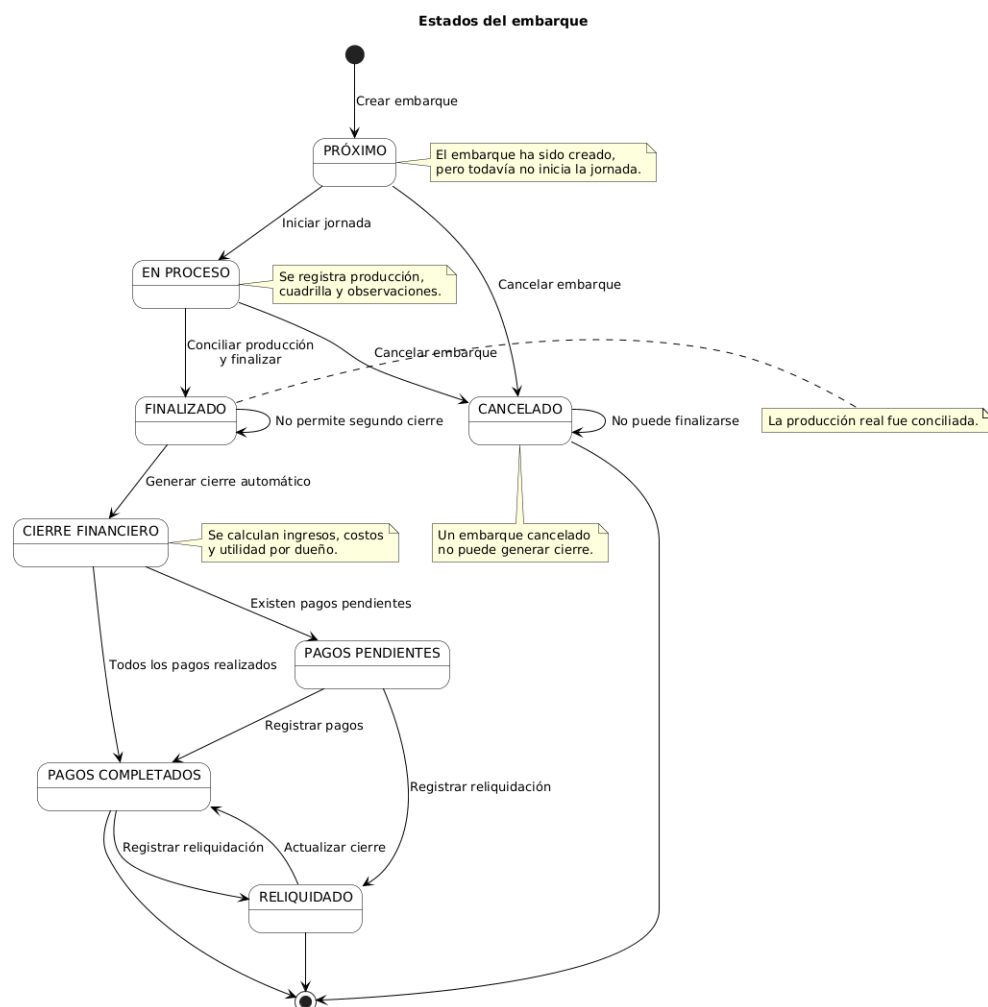


Figura 11 Diagrama de estados del embarque

La interacción técnica para finalizar el embarque y generar el cierre se resume en la

Figura 11. El flujo incluye la validación del token y del rol, la comprobación del estado del embarque, el cálculo proporcional, la persistencia del cierre y el registro de auditoría.

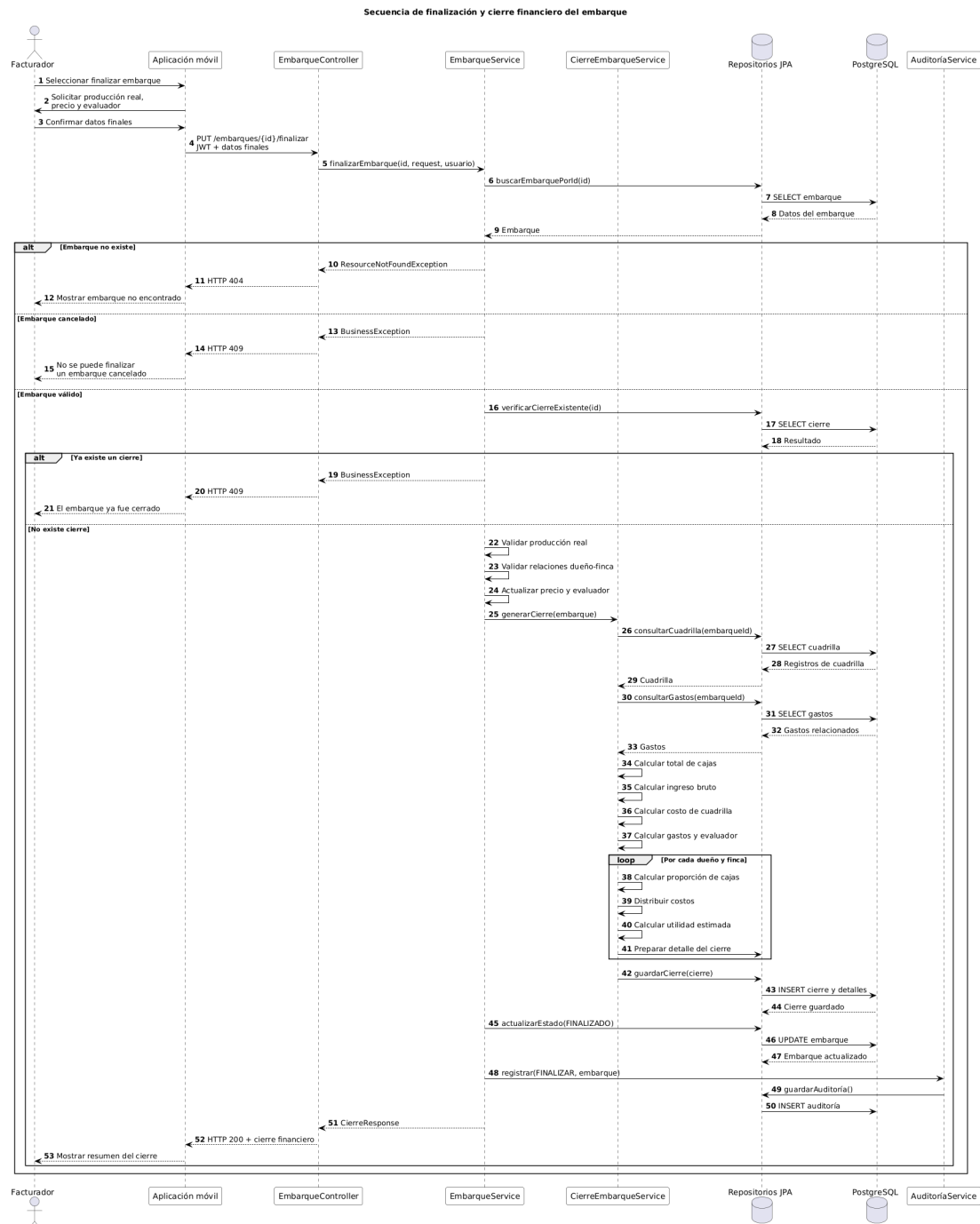


Figura 12 Diagrama de secuencia de la finalización y cierre del embarque

1.25.7 Módulo de gastos

Los gastos se clasifican en cuatro tipos: de embarque, de finca, compra de producto y servicio. Las compras de producto y los servicios exigen indicar el dueño responsable y admiten proveedor, cantidad, unidad, precio unitario y número de la factura del proveedor.

Una decisión de diseño relevante es que la compra de producto genera automáticamente la entrada correspondiente en el inventario del dueño, de modo que registrar el gasto y actualizar la bodega sea una sola operación y no dos, eliminando una fuente habitual de olvidos. Cada gasto mantiene además un estado de comprobante (listo para emitir o emitido) que alimenta el módulo de facturación.

1.25.8 Módulo de mano de obra y planilla semanal

El jornal diario registra al trabajador, la labor, el pago, la fecha, las fincas donde se trabajó y el dueño responsable; cuando la labor es compartida, el pago se reparte entre varios dueños. Cada jornal puede marcarse como realizado, y el sistema clasifica su verificación en tres estados: hecho, pendiente mientras la semana está en curso, y pendiente de verificar cuando ya pasó el sábado de esa semana sin confirmación.

La aplicación móvil complementa este control con recordatorios locales configurables (uno por la mañana y otro por la tarde) que avisan al mayordomo los jornales del día que faltan por verificar. Sobre los jornales se construye la planilla semanal: el sistema agrupa los jornales de cada trabajador entre lunes y sábado, suma los pagos, descuenta los adelantos entregados durante la semana, admite un bono adicional y calcula el total a pagar, con el pago programado para el sábado.

Al registrarse el pago (con su método, referencia y observación) los jornales incluidos quedan marcados como pagados, y el registro semanal es único por trabajador, año y semana, lo que impide pagos duplicados.

La Figura 12 resume el flujo desde el registro diario hasta la liquidación semanal y la emisión del comprobante.

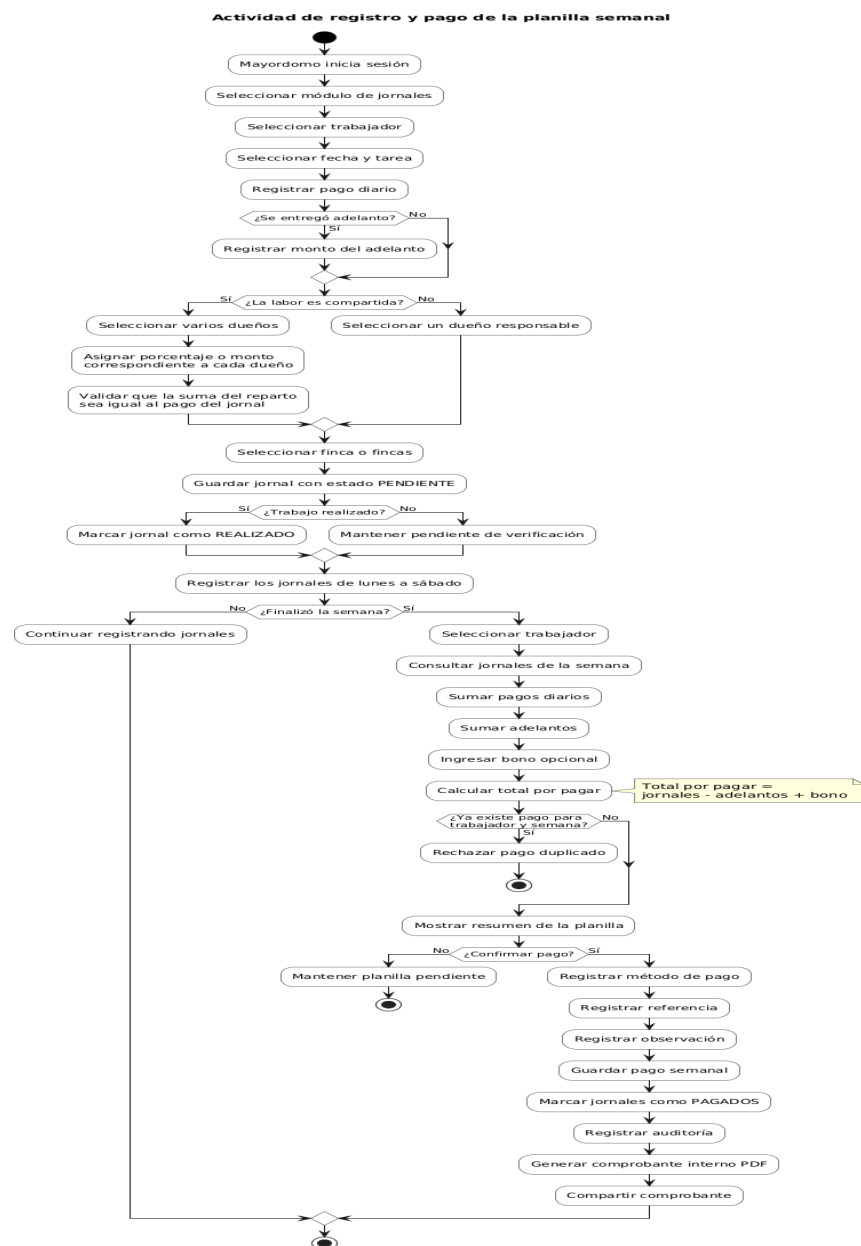


Figura 13 Diagrama de actividades de la planilla semanal

1.25.9 Módulo de facturación y comprobantes

Aunque el prototipo no emite facturación electrónica homologada, sí resuelve la necesidad práctica que la motivaba: dejar constancia formal de las liquidaciones. El módulo de facturación reúne los cierres de embarque y las planillas semanales y genera para cada uno un comprobante en HTML que la aplicación convierte a PDF en el propio dispositivo mediante expo-print; el documento puede imprimirse o compartirse a través de la hoja nativa del sistema con expo-sharing, lo que en

la práctica se utiliza para enviarlo a los dueños por WhatsApp. El comprobante incluye el detalle por dueño, los pagos de la cuadrilla, el evaluador y espacios de firma, y su emisión queda registrada con el estado correspondiente. Si al momento de emitir aún existen pagos pendientes, el sistema advierte que el documento podría cambiar y pide confirmación. En todos los casos se trata de comprobantes internos de respaldo: el prototipo no implementa facturación electrónica homologada, no genera XML tributario autorizado, no firma electrónicamente los documentos y no se conecta con el Servicio de Rentas Internas (SRI); los PDF son documentos internos que pueden compartirse mediante WhatsApp u otras aplicaciones.

1.25.10 Estructura del frontend y navegación por roles

El frontend es un proyecto de Expo con React Native y TypeScript, organizado en carpetas de componentes reutilizables, contextos (autenticación, estado sin conexión y preferencias), navegación, pantallas, servicios de acceso a la API con Axios y utilidades. La navegación principal es una barra de pestañas cuyo contenido depende del rol autenticado: una matriz de acceso define qué módulos ve cada rol, de modo que, por ejemplo, el bodeguero solo encuentra inventario y reportes, el capitán trabaja sobre embarques y productividad, y el dueño dispone además de la pestaña “Mi finca”. Esta restricción visual es solo la primera barrera, pues el backend vuelve a validar cada operación.

Para el registro de la ubicación de las fincas se integró un mapa interactivo basado en Leaflet dentro de un WebView, sobre el cual el usuario arrastra un marcador o toma su posición actual mediante expo-location; las coordenadas seleccionadas se guardan como latitud y longitud de la finca. El mapa de navegación de la Figura 13 muestra las pantallas disponibles para cada perfil y facilita la comparación entre las restricciones visuales del frontend y las autorizaciones aplicadas por el backend.

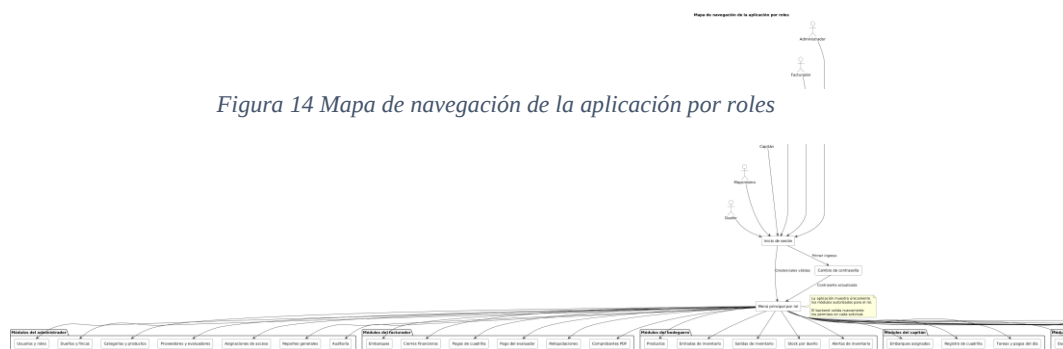


Figura 14 Mapa de navegación de la aplicación por roles

1.25.11 Modo sin conexión y sincronización

La operación sin conexión se apoya en una base de datos SQLite local creada con expo-sqlite, que mantiene una cola de operaciones pendientes. Cuando el dispositivo no tiene internet, el registro del embarque se guarda en esa cola con su carga útil completa y la aplicación muestra un aviso de modo sin conexión; al detectarse nuevamente la red, un proceso envía los registros pendientes al endpoint de sincronización del backend, que los crea marcados con el estado SINCRONIZADO, y los elimina de la cola local. El Código 3 muestra la estructura de la cola local.

```
CREATE TABLE IF NOT EXISTS offline_queue (
  id TEXT PRIMARY KEY NOT NULL,
  entity TEXT NOT NULL,
  payload TEXT NOT NULL,
  created_at TEXT NOT NULL,
  synced INTEGER NOT NULL DEFAULT 0
);
```

Código 3 Cola local de operaciones pendientes en SQLite

La secuencia completa de almacenamiento local y sincronización se representa en la Figura 14. El servidor confirma la creación antes de que la aplicación elimine el elemento de la cola local, evitando la pérdida del registro pendiente. Si la sincronización falla (por pérdida de conectividad o por un error del servidor), la operación permanece en la cola local y se reintenta en el siguiente evento de sincronización; solo se elimina de la cola cuando el servidor confirma su creación. El prototipo no implementa una resolución avanzada y automática de conflictos: la sincronización se limita al reenvío de las operaciones encoladas y a su registro en el servidor. En la versión evaluada, el almacenamiento sin conexión se aplicó principalmente al registro de embarques mediante un endpoint de sincronización

especializado; las demás operaciones se registran en línea a través de su endpoint normal.

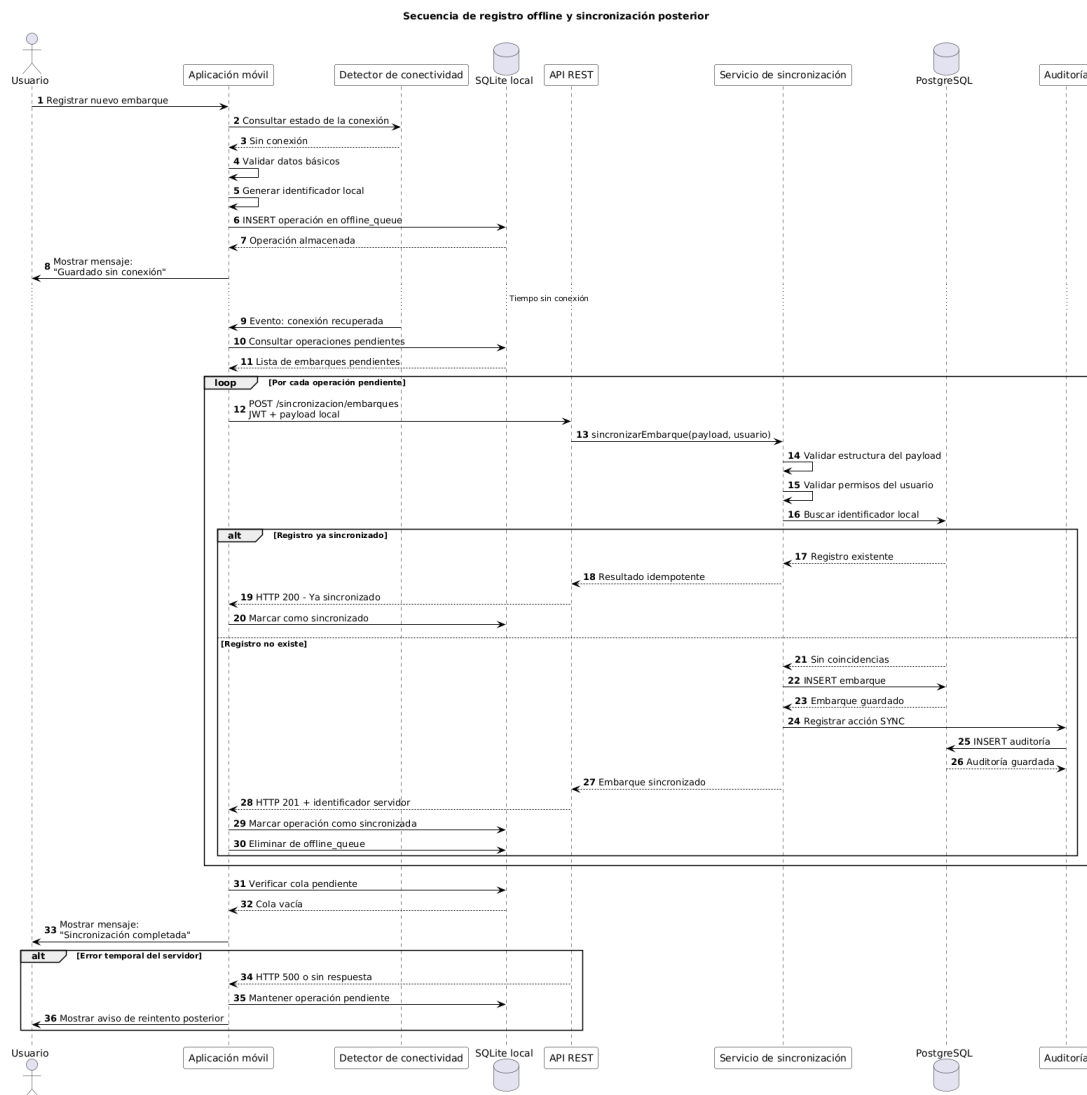


Figura 15 Diagrama de secuencia de la sincronización sin conexión

1.25.12 Reportes

El módulo de reportes ofrece, según el rol, ventas por dueño con su detalle por embarque, gastos filtrados por tipo y dueño, mano de obra semanal, inventario (productos bajo stock mínimo y lotes próximos a caducar) y productividad por dueño y embarque con cajas, racimos y ratio de conversión. Los reportes pueden visualizarse en pantalla y exportarse a PDF para compartirse con los dueños. A partir del cierre de embarque, el sistema identifica además los casos en que los costos asignados a un dueño superan su ingreso, presentando el monto que debe aportar para cubrir la diferencia.

1.25.13 Documentación y despliegue de la API

La API quedó documentada de forma interactiva con Swagger (OpenAPI), lo que facilitó tanto las pruebas manuales durante el desarrollo como la validación de los contratos entre frontend y backend. El despliegue del prototipo para las pruebas con usuarios se realizó en un entorno controlado, con el backend y la base de datos ejecutándose en un equipo servidor accesible desde la red local de los dispositivos móviles. Este despliegue tuvo carácter demostrativo y sirvió para la sesión de validación con los usuarios. Con posterioridad a esas pruebas, el backend y la base de datos se trasladaron a Railway, una plataforma como servicio que los expone a través de internet y elimina la dependencia de la red local de la finca. Ese traslado amplía la accesibilidad del prototipo, pero tampoco constituye un entorno productivo en sentido estricto: no se definió una política de respaldos, no se instrumentó monitoreo ni alertas, no se estableció un plan de recuperación ante fallos y la infraestructura no fue dimensionada a partir de una medición de carga real. Por ese motivo el trabajo describe el estado del sistema como prototipo desplegado para validación y no como sistema en producción. En el Anexo K se incluye un manual breve de usuario con las pantallas de los principales flujos del sistema, y en el Anexo L consta el formato del acta de validación funcional, prevista para su suscripción por el propietario de la hacienda.

Pruebas y Resultados

1.26. Pruebas funcionales

Se ejecutaron 19 casos de prueba funcional (CP01–CP19) sobre la aplicación móvil y la API REST, realizados de forma manual al cierre de los sprints y complementados con validaciones de los servicios mediante Postman y la documentación interactiva de Swagger. Para cada caso se verificó la regla de negocio, el resultado devuelto por el backend y la respuesta mostrada en la interfaz. Estas pruebas funcionales no constituyen una batería de pruebas unitarias automatizadas: la cobertura automatizada del proyecto se limita a una prueba básica de contexto de la aplicación Spring Boot, mientras que la evaluación de usabilidad se aborda por separado mediante el cuestionario System Usability Scale (sección 6.3). La Tabla 33 resume los principales casos ejecutados, y la correspondencia entre requisitos, historias de usuario y casos de prueba se detalla en la matriz de trazabilidad del Anexo F.

ID	RF	Escenario	Resultado obtenido	Estado
CP01	RF01	Inicio de sesión con credenciales válidas.	El sistema autenticó al usuario, generó el token JWT y mostró las opciones correspondientes a su rol.	Aprobado
CP02	RF01	Inicio de sesión con credenciales inválidas.	El sistema rechazó la solicitud y mostró un mensaje de credenciales incorrectas.	Aprobado
CP03	RF03	Primer inicio de sesión de un usuario nuevo.	La aplicación solicitó el cambio de contraseña antes de permitir el acceso a los módulos.	Aprobado
CP04	RF10	Registro de una salida superior al saldo disponible del dueño.	La operación fue rechazada y el inventario conservó el saldo anterior.	Aprobado
CP05	RF12	Consulta de productos bajo stock mínimo y lotes próximos a caducar.	El sistema identificó y presentó los productos que cumplían las condiciones de alerta.	Aprobado
CP06	RF16	Selección de una finca que no pertenece al dueño indicado.	El backend rechazó la combinación y evitó guardar un detalle inconsistente.	Aprobado
CP07	RF15	Creación de un embarque sin ingresar manualmente el color de cinta.	El sistema calculó la semana del ciclo y asignó el color configurado.	Aprobado
CP08	RF22	Intento de finalizar un embarque cancelado.	La operación fue rechazada y no se generó el cierre financiero.	Aprobado
CP09	RF23– RF24	Finalización de un embarque con producción de varios dueños.	Se generó un único cierre y los costos se distribuyeron en proporción a las cajas aportadas.	Aprobado
CP10	RF25– RF26	Registro de pagos de cuadrilla y evaluador.	Los pagos actualizaron su estado y registraron la fecha de la operación.	Aprobado
CP11	RF29	Registro de un gasto de tipo compra de producto.	La operación creó el gasto y generó la entrada correspondiente en el inventario del dueño.	Aprobado
CP12	RF34– RF35	Generación y pago de la planilla semanal.	El sistema consolidó los jornales, descontó adelantos, agregó el bono registrado y evitó duplicar el pago de la misma semana.	Aprobado
CP13	RF37– RF39	Consulta de información con perfiles de facturador, mayordomo y dueño.	La aplicación y el backend limitaron los módulos y registros de acuerdo con las asignaciones y el rol autenticado.	Aprobado
CP14	RF36	Generación de un comprobante interno.	La aplicación produjo el documento PDF y permitió compartirlo mediante la hoja nativa del dispositivo.	Aprobado
CP15	RF41	Ejecución de una operación auditable.	La bitácora registró el usuario, la acción, la entidad afectada y la fecha.	Aprobado

ID	RF	Escenario	Resultado obtenido	Estado
CP16	RF42	Registro de un embarque sin conexión y sincronización posterior.	La aplicación almacenó la operación en SQLite y la envió al servidor al recuperar la conectividad.	Aprobado
CP17	RF43	Configuración de los recordatorios de verificación de jornales.	El dispositivo emitió las notificaciones locales en los horarios configurados con los jornales del día pendientes.	Aprobado
CP18	RF44	Registro de la ubicación de una finca sobre el mapa interactivo.	La aplicación permitió arrastrar el marcador y tomar la posición actual, y guardó las coordenadas de la finca.	Aprobado
CP19	RF45– RF46	Consulta de reportes por dueño y del detalle de aportes pendientes.	Los reportes presentaron la información filtrada según el rol y mostraron el monto a aportar cuando los costos superaron el ingreso.	Aprobado

Tabla 33 Resultados de las pruebas funcionales

Los casos ejecutados permitieron verificar las reglas de negocio más críticas del prototipo. Las evidencias visuales de las pruebas se incorporan en el Anexo G mediante capturas de la aplicación, Postman o Swagger, conservando el mismo identificador utilizado en la Tabla 33.

1.27. Pruebas de la API REST

Las pruebas de la API se organizaron por módulo y consideraron solicitudes válidas, datos incompletos, ausencia de token, permisos insuficientes y violaciones de reglas de negocio. La Tabla 34 resume los criterios de respuesta comprobados; el registro detallado de endpoints y capturas se presenta en el Anexo H.

Tabla 34 Criterios verificados en las pruebas de la API REST

Tipo de prueba	Criterio verificado	Resultado
Autenticación	Las credenciales válidas generan un token y las inválidas producen una respuesta de rechazo.	Aprobado
Autorización	Las solicitudes sin token o con un rol no permitido son rechazadas antes de ejecutar la operación.	Aprobado
Validación de datos	Los campos obligatorios, rangos y relaciones entre entidades son validados antes de persistir información.	Aprobado
Reglas de negocio	Se rechazaron operaciones como saldos negativos, cierres duplicados, embarques cancelados y relaciones dueño–finca inválidas.	Aprobado
Persistencia	Las operaciones aceptadas se almacenaron y pudieron consultarse mediante los servicios correspondientes.	Aprobado
Manejo de errores	Las respuestas de error incluyeron un código HTTP coherente y un mensaje comprensible para el cliente móvil.	Aprobado

1.27.1 Consideraciones sobre el desempeño técnico

Las pruebas descritas verifican el comportamiento funcional de la API, pero no su desempeño bajo carga. Dado que se trata de una arquitectura móvil cliente–servidor, corresponde precisar qué aspectos de rendimiento fueron evaluados y cuáles quedaron fuera del alcance de este trabajo.

Respecto de la latencia de respuesta, no se ejecutó una medición instrumentada de los tiempos de la API. Durante las pruebas funcionales se comprobó que las respuestas se producían dentro de márgenes que no afectaban la fluidez de la interfaz, pero esta observación es cualitativa y no sustituye una medición sistemática. Una caracterización adecuada exigiría registrar la distribución de los tiempos de respuesta por endpoint, con sus percentiles, bajo condiciones de red controladas.

En cuanto al comportamiento ante baja conectividad, el prototipo incorpora un mecanismo de registro sin conexión y sincronización diferida para el módulo de embarques, cuyo funcionamiento se verificó desactivando la conectividad del dispositivo, registrando operaciones y restableciendo la red para comprobar la sincronización posterior. Estas pruebas confirman que el flujo previsto opera, pero no cubren escenarios de conmutación de red durante una transacción en curso, de conectividad intermitente ni de

sincronización parcial interrumpida, situaciones que son las que realmente ponen a prueba la robustez de este tipo de mecanismos.

Tampoco se ejecutaron pruebas de carga ni de concurrencia. El sistema atiende a una sola hacienda, con seis perfiles diferenciados y veinte usuarios registrados, aunque su actividad no se distribuye de manera uniforme: se concentra en los dos días de embarque de la semana y, dentro de ellos, en el momento del cierre. Esta forma de uso hace previsible que la carga simultánea permanezca en un orden de magnitud que una API construida con Spring Boot y PostgreSQL atiende sin dificultad con su configuración por omisión.

Este razonamiento, sin embargo, tiene un límite que conviene explicitar, y por ello se presenta como una expectativa fundada en el perfil de uso y no como una propiedad verificada. El argumento sostiene que la concurrencia esperada es baja, no que el sistema haya sido probado bajo concurrencia. La distinción importa porque el sistema tiene veinte usuarios registrados y la carga real nunca fue medida. Operaciones como el cierre simultáneo de un mismo embarque desde dos dispositivos, o el registro concurrente de salidas sobre el mismo ítem de inventario, dependen del control de aislamiento transaccional y no fueron sometidas a prueba. Se recomienda incorporar pruebas de concurrencia sobre estas operaciones críticas antes de un despliegue en producción.

Debe señalarse, por último, una limitación en la estrategia de pruebas. La verificación funcional se ejecutó de forma manual al término de cada sprint, mientras que la cobertura automatizada del backend se limita a una prueba de carga del contexto de la aplicación Spring Boot. Las reglas de negocio que involucran cálculos económicos (el reparto proporcional de gastos, el ratio de conversión por dueño y la liquidación semanal de la cuadrilla) carecen de pruebas unitarias que protejan su comportamiento frente a modificaciones futuras del código. Dado que se trata de cálculos que afectan pagos a terceros, la incorporación de pruebas unitarias sobre estas funciones críticas constituye una tarea prioritaria para la evolución del prototipo.

1.27.2 Evaluación de usabilidad mediante SUS

La evaluación de usabilidad se aplicó el 7 de julio de 2026, al término de la jornada de embarque, a cinco usuarios de la hacienda que entre ellos cubren los seis roles diferenciados por el sistema. La correspondencia no es de uno a uno porque el propietario ejerce simultáneamente los perfiles de dueño y de administrador: es él quien impulsó el desarrollo del sistema con el propósito de dar transparencia al reparto de ingresos y gastos frente a sus seis hermanos, que son los demás propietarios de la hacienda. Los cuatro participantes restantes corresponden, uno por perfil, a los roles de mayordomo, facturador, bodeguero y capitán, de modo que la valoración recoge tanto la experiencia de quienes ejecutan el cierre financiero y

la planilla como la de quienes registran operaciones durante la jornada de embarque. Esta composición es una decisión de muestreo por conveniencia orientada a la cobertura de roles y no a la representatividad estadística: privilegia la amplitud de perfiles sobre el número de casos, con la contrapartida de que cada rol queda representado por una sola persona. Después de una sesión guiada, los participantes ejecutaron tareas relacionadas con autenticación, embarques, inventario, jornales, cierres y consulta de reportes, y posteriormente respondieron los diez ítems del cuestionario System Usability Scale, cuyo instrumento completo y matriz de respuestas se presentan en el Anexo I.

Para calcular el puntaje individual, en los ítems impares se restó 1 a la respuesta seleccionada y, en los ítems pares, la respuesta se restó de 5. La suma de las contribuciones se multiplicó por 2,5, como se expresa en la Ecuación 2.

Ecuación 2 Cálculo SUS

$$SUS = \left[\sum_{(i \in \{1,3,5,7,9\})} (R_i - 1) + \sum_{(i \in \{2,4,6,8,10\})} (5 - R_i) \right] \times 2,5$$

Tabla 35 Resumen de la evaluación de usabilidad

Indicador	Resultado
Número de participantes	5
Puntaje promedio SUS	82,5 puntos
Valor de referencia	68 puntos
Interpretación	Usabilidad aceptable

El puntaje promedio obtenido fue de 82,50 puntos ($n = 5$; $DE = 2,50$), superior al valor de referencia de 68 puntos habitualmente empleado como umbral de

usabilidad aceptable. Los estadísticos descriptivos completos de la distribución se presentan a continuación.

Tabla 36 Estadísticos descriptivos del puntaje SUS (n = 5)

Estadístico	Valor
Tamaño de muestra (n)	5
Media	82,50
Desviación estándar	2,50
Mediana	82,50
Mínimo – Máximo	80,00 – 85,00
Rango	5,00
Intervalo de confianza al 95 % de la media	79,40 – 85,60

La dispersión observada es reducida: los cinco puntajes se concentran entre 80,00 y 85,00, con una desviación estándar de 2,50 puntos. Resulta llamativo que esa homogeneidad se mantenga a pesar de que cada participante ocupa un rol distinto, con tareas, permisos y niveles de familiaridad tecnológica dispares. Cabe una lectura favorable, que la navegación diferenciada por rol funciona de manera equivalente para perfiles heterogéneos, y otra más cautelosa: los cinco comparten el mismo contexto de trabajo y recibieron la misma sesión de inducción, factores que por sí solos reducen la variabilidad esperable. Con cinco observaciones no es posible discriminar entre ambas explicaciones.

En términos de referencia académica, la media obtenida se sitúa por encima del promedio de 68 puntos reportado en estudios que consolidan cientos de evaluaciones SUS, y en la escala de adjetivos propuesta por Bangor y colaboradores [23] se ubica en la franja comprendida entre las calificaciones «bueno» y «excelente». De acuerdo con la curva de calificación de Sauro [33], un puntaje de 82,50 corresponde aproximadamente al percentil 90 y a una calificación de grado A. Estas referencias permiten situar el resultado, pero no lo convierten en un valor generalizable.

El intervalo de confianza al 95 % de la media abarca de 79,40 a 85,60 puntos. Su amplitud, superior a seis puntos con solo cinco observaciones, ilustra la limitada estabilidad del promedio: con una muestra de este tamaño, la incorporación de un único participante adicional podría desplazar la media de forma apreciable. El intervalo se calcula, además, bajo el supuesto de normalidad, que no es verificable con cinco casos.

Por lo anterior, el resultado describe exclusivamente la experiencia de los cinco participantes evaluados en la Hacienda Bananera San Diego y no se plantea como estimación de la usabilidad percibida por una población más amplia de usuarios de

sistemas ERP móviles. La lectura apropiada es que, dentro del grupo evaluado, la percepción de usabilidad fue favorable y homogénea.

Conviene precisar además el alcance que impone tener un solo participante por rol. Los puntajes individuales no admiten lectura por perfil: una diferencia entre el valor del mayordomo y el del bodeguero es indistinguible de la variación entre dos personas cualesquiera, porque no existe una segunda observación en ninguno de los dos grupos con la cual contrastarla. A esto se añade que la sesión se realizó el mismo día del embarque, inmediatamente después de la inducción guiada, condición que concentra la valoración en un momento de primera impresión y no permite observar cómo evoluciona la percepción de usabilidad con el uso sostenido. La cobertura de roles refuerza la validez de contenido de la evaluación, ya que las diez preguntas fueron respondidas por usuarios que ejercen funciones distintas dentro del sistema, pero no habilita comparaciones entre perfiles. Para eso se requerirían varios usuarios por rol, condición que la plantilla de la hacienda no permite satisfacer.

1.28. Comparación de tiempos de registro

Para evaluar el objetivo de eficiencia se cronometraron las mismas operaciones ejecutadas primero con el procedimiento manual de la hacienda y luego con el prototipo, empleando la ficha de observación del Anexo C; las hojas consolidadas de medición constan en el Anexo J. La reducción porcentual se calcula mediante la Ecuación 3.

Ecuación 3 Cálculo de la reducción porcentual del tiempo

$$\text{Reducción del tiempo} = \frac{T_{\text{manual}} - T_{\text{sistema}}}{T_{\text{manual}}} \times 100$$

Tabla 37 Comparación de tiempos de registro

Operación	Tiempo manual	Tiempo con sistema	Reducción
Cierre de embarque	30,00 min	7,00 min	76,67 %
Liquidación semanal de jornales	25,00 min	6,00 min	76,00 %
Registro de movimiento de inventario	5,00 min	1,50 min	70,00 %

Promedio	–	–	74,22 %
----------	---	---	---------

La reducción promedio de las tres operaciones cronometradas fue de 74,22 % ($n = 3$ operaciones; $DE = 3,67$), valor que supera el mínimo de 70 % fijado en el objetivo general. Debe subrayarse que cada operación se midió una sola vez, durante una única jornada de observación, por lo que estos porcentajes constituyen una medición puntual y no un promedio de repeticiones.

Antes de los porcentajes conviene presentar los tiempos absolutos, que son el dato primario de la comparación: para cada operación, el tiempo con el procedimiento manual, el tiempo con el prototipo y la diferencia en minutos.

Tabla 38 Tiempos absolutos de ejecución por operación

Operación	Manual (min)	Prototipo (min)	Diferencia (min)	Reducción
Cierre de embarque	30,00	7,00	23,00	76,67 %
Liquidación semanal de jornales	25,00	6,00	19,00	76,00 %
Registro de movimiento de inventario	5,00	1,50	3,50	70,00 %
Total del ciclo	60,00	14,50	45,50	74,22 % (promedio simple)

Al disponer de una sola medición por operación y condición, estos tiempos no admiten el cálculo de una desviación estándar: cada valor es una observación única y no la media de varios ensayos. Los estadísticos de dispersión que se reportan a continuación describen la distribución de los tres porcentajes de reducción entre operaciones, no la variabilidad de cada medición, que quedó sin estimar. Por eso el apartado 3.6.3 consigna el número de repeticiones como una limitación del protocolo y no como un parámetro del diseño.

Tabla 39 Estadísticos descriptivos de la reducción porcentual del tiempo

Estadístico	Valor
Operaciones medidas (n)	3
Media	74,22 %
Desviación estándar	3,67 %
Mediana	76,00 %

Estadístico	Valor
Mínimo – Máximo	70,00 % – 76,67 %
Rango	6,67 %
Tiempo total ahorrado por ciclo completo	45,50 min

El desglose por operación muestra que la magnitud de la mejora no es uniforme y que su distribución obedece a una razón técnica identificable. El cierre de embarque concentra el mayor ahorro absoluto, con 23,00 minutos por evento, seguido de la liquidación semanal de jornales con 19,00 minutos y del registro de un movimiento de inventario con 3,50 minutos.

La explicación de esta jerarquía se encuentra en la proporción de cálculo derivado que contiene cada operación. El cierre de embarque exige, en el procedimiento manual, repartir los gastos entre los propietarios en proporción a las cajas aportadas, calcular el ratio de conversión de cada dueño y liquidar el pago de la cuadrilla, operaciones aritméticas que se resuelven con calculadora y se transcriben entre varias hojas. En el prototipo, todos esos valores se derivan de datos ya capturados en el momento del registro, de modo que el cierre se reduce a verificar y confirmar. La liquidación semanal responde al mismo patrón: consolida jornales previamente registrados y su ahorro proviene de suprimir la reagrupación manual por trabajador y por dueño.

El registro de movimiento de inventario, en cambio, presenta la reducción relativa más baja (70,00 %) porque casi no contiene cálculo derivado: consiste esencialmente en identificar el producto, indicar la cantidad y confirmar. Su tiempo está dominado por la interacción física y la digitación, componentes que el sistema no puede eliminar. Este resultado sugiere un techo de mejora para las operaciones de captura simple y explica por qué la reducción observada disminuye a medida que baja la carga de cálculo de la tarea.

A partir de los ahorros unitarios puede estimarse el impacto acumulado en horas-hombre. La proyección que se presenta a continuación no constituye una medición, sino una extrapolación lineal de los tiempos observados durante la jornada de observación, aplicada a la frecuencia operativa que declaró el propietario.

La frecuencia declarada por el propietario es de dos embarques por semana y una liquidación de planilla por semana, lo que arroja un ahorro semanal de 65,00 minutos, equivalentes a 1,08 horas-hombre. Conviene señalar que el apartado 1.2.1 describe los embarques como un evento que ocurre una o dos veces por semana, de manera que la proyección adopta el extremo superior de ese rango y constituye por tanto una estimación optimista: en una semana de un solo embarque el ahorro se reduce a 42,00 minutos. El registro de movimientos de inventario queda fuera del cálculo acumulado por una razón operativa: no responde a una periodicidad fija. Durante el embarque apenas se registran

salidas de bodega, a veces ninguna más allá del combustible, mientras que el grueso de los movimientos se concentra en las labores culturales de la semana, principalmente la fertilización y la fumigación, cuya cadencia varía con el ciclo del cultivo. Por ese motivo su aporte se expresa en términos unitarios: cada movimiento registrado ahorra 3,50 minutos, y el ahorro acumulado depende del número de labores ejecutadas en el período que se considere.

Tabla 40 Proyección del ahorro de horas-hombre según la frecuencia operativa declarada

Periodo	Ahorro estimado (min)	Ahorro estimado (h)
Semana	65,00	1,08
Mes (4,33 semanas)	281,45	4,69
Año (52 semanas)	3 380,00	56,33

En términos económicos, el ahorro anual estimado de 56,33 horas-hombre corresponde a tiempo del propietario y del mayordomo que deja de destinarse a tareas de transcripción y cálculo. A ese total se suma el ahorro de los movimientos de inventario, que no se acumula aquí por carecer de una periodicidad fija.

Para expresar ese ahorro en unidades monetarias se tomó como base la remuneración del mayordomo, que percibe 150,00 dólares por una semana de lunes a sábado. Sobre seis jornadas de ocho horas, el costo asciende a 25,00 dólares por día y 3,13 dólares por hora. Se eligió esta base y no la del jornalero, que gana 20,00 dólares por jornada equivalentes a 2,50 dólares por hora, porque las operaciones cronometradas son tareas administrativas que ejecutan el propietario y el mayordomo, no el personal de campo.

Aplicado ese costo horario, el ahorro semanal de 1,08 horas equivale a 3,39 dólares; el mensual, de 4,69 horas, a 14,66 dólares; y el anual, de 56,33 horas, a 176,04 dólares. La cifra es deliberadamente conservadora: el tiempo del propietario, que no percibe una remuneración declarada, se valoró al mismo costo que el del mayordomo, cuando su costo de oportunidad es previsiblemente superior.

Conviene ser explícito sobre lo que este resultado significa. Ciento setenta y seis dólares anuales no constituyen por sí solos una justificación económica del desarrollo del sistema; el ahorro de tiempo administrativo, en una explotación de este tamaño, es un beneficio real pero modesto en términos monetarios. El valor del prototipo se concentra en la reducción de las inconsistencias de registro y en la trazabilidad de la información, cuyo impacto económico no es directamente cuantificable con los datos recogidos, pero se manifiesta en la exactitud del reparto de gastos entre propietarios y de la liquidación de la cuadrilla, es decir, en pagos a terceros donde un error aritmético tiene consecuencias inmediatas. Un error en el reparto de un solo embarque puede superar con facilidad el ahorro administrativo de un año entero.

Corresponde reiterar el alcance de esta estimación. Se apoya en una medición única por operación y en una frecuencia operativa declarada por el propietario, no verificada contra el registro histórico de la hacienda. Cualquier variación en el número de embarques modifica el resultado de forma proporcional. La cifra debe leerse, por tanto, como un orden de magnitud del beneficio operativo y no como un valor exacto.

1.29. Comparación de errores de registro

Además del tiempo, se comparó la cantidad de inconsistencias detectadas al ejecutar las operaciones seleccionadas. Se consideraron errores de cálculo, duplicidad, campos obligatorios omitidos y relaciones inválidas entre dueño y finca.

Tabla 41 Comparación de errores de registro

Indicador	Resultado
Errores detectados con el procedimiento manual	12
Errores detectados con el prototipo	1
Reducción de errores	91,67 %

La reducción de 91,67 % corresponde al paso de 12 inconsistencias observadas con el procedimiento manual a una sola con el prototipo, en el mismo conjunto de operaciones y durante la misma jornada de observación. Para interpretar este resultado conviene distinguir entre los errores que el sistema erradica por diseño y aquellos que permanecen fuera de su alcance, distinción que permite valorar el grado de tolerancia a fallos alcanzado.

Tabla 42 Categorización de los errores según el mecanismo de control del prototipo

Categoría de error	Mecanismo de control en el prototipo	Condición
Errores de cálculo en el reparto de gastos, el ratio por dueño y la liquidación de la cuadrilla	El cálculo se deriva automáticamente de los datos capturados; no existe aritmética manual intermedia	Erradicado por diseño
Duplicidad de cierres o de registros	Restricciones de unicidad en la base de datos y control de estado del embarque	Erradicado por diseño
Campos obligatorios omitidos	Validación en el formulario y en el servicio antes de persistir	Erradicado por diseño

Categoría de error	Mecanismo de control en el prototipo	Condición
Relaciones inválidas entre dueño y finca	Integridad referencial y filtrado de las opciones disponibles según el contexto	Erradicado por diseño
Salos negativos de inventario	Validación de existencias previa al registro de la salida	Erradicado por diseño
Digitación de un valor numérico incorrecto pero sintácticamente válido	Sin control automático; solo detectable por revisión humana	Residual
Selección de una opción válida pero equivocada	Sin control automático; solo detectable por revisión humana	Residual

La categorización muestra que los errores erradicados comparten una característica común: son detectables mediante una regla formal verificable en el momento del registro, sea una restricción de integridad, una validación de obligatoriedad o un cálculo determinista. El prototipo los previene porque traslada esa regla del criterio del operador al código del sistema.

Los errores residuales, en cambio, son semánticamente correctos y sintácticamente válidos: un peso mal digitado o un dueño seleccionado por equivocación producen registros que ninguna validación puede rechazar sin conocer la intención del usuario. Esta es la frontera de la tolerancia a fallos del prototipo, y explica por qué la reducción se detiene en 91,67 % y no alcanza el total. Mitigarla exigiría controles de otro orden, como reglas de rango sobre valores atípicos, pantallas de confirmación antes de operaciones irreversibles o una revisión cruzada al cierre.

Debe precisarse una limitación en el registro de estos datos: la ficha de observación consignó el número total de inconsistencias, pero no su desagregación por categoría. La clasificación anterior se sustenta, por tanto, en el mecanismo de control implementado para cada tipo de error y no en un recuento por categoría. Se recomienda que futuras mediciones registren esa desagregación, ya que permitiría cuantificar la contribución de cada mecanismo de validación a la reducción observada.

1.30. Discusión de resultados

La comparación de tiempos de registro mostró una reducción promedio de 74,22 % ($n = 3$ operaciones; $DE = 3,67$) al sustituir el procedimiento manual por el prototipo, valor superior al umbral mínimo de 70 % fijado en el objetivo general. El resultado es compatible con un efecto positivo de la digitalización sobre las operaciones administrativas evaluadas

(cierre financiero, planilla semanal e inventario), aunque el diseño empleado no permite atribuir la mejora exclusivamente al sistema.

En cuanto a la calidad del registro, la comparación de inconsistencias mostró una disminución del 91,67 %, al pasar de doce errores en el procedimiento manual a uno con el prototipo. Este resultado constituye un hallazgo propio del presente estudio. De manera general, la literatura sobre digitalización de mipymes agrícolas y agroindustriales destaca el potencial de las tecnologías digitales para fortalecer la productividad, la organización y la competitividad del sector [5]. La disminución observada se explica por la incorporación de validaciones automáticas y cálculos verificables; no obstante, se trata de una reducción significativa y no de una eliminación absoluta, pues persiste la posibilidad de errores de digitación o de uso.

La evaluación de usabilidad mediante el System Usability Scale arrojó un puntaje promedio de 82,50 puntos ($n = 5$; $DE = 2,50$; $IC\ 95\ \% \ 79,40\text{--}85,60$), superior al valor de referencia de 68 puntos. El resultado sugiere que las decisiones de diseño de la interfaz y de la navegación por roles fueron pertinentes para el grupo evaluado, sin que ello permita extender la valoración a otros perfiles de usuario.

Cabe preguntarse por qué el prototipo superó al método tradicional en la magnitud observada. El desglose por operación ofrece una respuesta: la ventaja no proviene de acelerar la digitación, sino de suprimir el cálculo derivado y la retranscripción entre soportes. En el procedimiento manual, el dato se anota una vez en la libreta de campo, se traslada a una hoja de consolidación y se vuelve a operar con calculadora para obtener el reparto de gastos, el ratio por dueño y la liquidación de la cuadrilla. Cada traslado consume tiempo y añade una oportunidad de error. El prototipo captura el dato una sola vez y deriva de él todos los valores subordinados, de modo que el ahorro es mayor cuanto más cálculo contiene la tarea. Esta relación explica que la reducción alcance 76,67 % en el cierre de embarque y descienda a 70,00 % en el registro de inventario, operación en la que apenas hay cálculo que suprimir.

Un segundo factor concurre en el mismo sentido: la eliminación de la espera entre la captura en campo y el procesamiento en oficina. En el procedimiento manual, el cierre financiero no podía iniciarse hasta que las hojas de campo llegaban físicamente al escritorio. El registro móvil con sincronización diferida elimina ese desplazamiento del flujo de trabajo. Conviene señalar, no obstante, que este componente del ahorro no fue aislado en la medición y que su contribución no puede cuantificarse por separado con los datos disponibles.

En cuanto al alto nivel de usabilidad registrado, tres factores permiten explicarlo de forma plausible. El primero es la correspondencia entre la estructura de la aplicación y la estructura del trabajo real: las pantallas reproducen la secuencia del embarque tal como la ejecutan los usuarios, sin exigirles reordenar sus tareas para adaptarse al sistema. El segundo es la navegación diferenciada por rol, que reduce la carga cognitiva al presentar a cada perfil únicamente las funciones que le competen. El tercero es el uso de la terminología propia de la hacienda (dueño, ratio, cuadrilla, enfunde) en lugar de vocabulario genérico de sistemas ERP.

A estos factores debe añadirse una consideración crítica. Los cinco participantes intervinieron en la validación de requerimientos y en las revisiones de sprint, y la sesión de evaluación fue precedida de una inducción guiada. Ambas circunstancias favorecen puntuaciones altas: se trata de usuarios familiarizados con el sistema y con expectativas ya formadas sobre él, no de usuarios nuevos. La baja dispersión observada ($DE = 2,50$) es coherente con esta interpretación. El puntaje obtenido debe leerse, por tanto, como la valoración de un grupo implicado en el desarrollo y no como una estimación de la usabilidad que percibiría un usuario ajeno al proyecto.

Los resultados anteriores deben interpretarse a la luz de las amenazas a la validez que introduce el diseño empleado. Al tratarse de un diseño preexperimental de un solo grupo con medición antes y después, sin grupo de control ni asignación aleatoria, la mejora observada no puede atribuirse de forma exclusiva al prototipo.

En cuanto a la validez interna, cuatro amenazas resultan pertinentes. La primera es el efecto de aprendizaje: las mediciones con el prototipo se tomaron después de las manuales y sobre las mismas operaciones, de modo que parte de la mejora puede deberse a la repetición de la tarea. La segunda es el efecto del observador, ya que el investigador cronometró personalmente ambas condiciones y ese conocimiento pudo influir en el registro. La tercera es la ausencia de repeticiones: una sola medición por operación impide separar la diferencia atribuible al sistema de la variabilidad natural del proceso. La cuarta es la variación de las condiciones entre ambas mediciones (volumen del embarque, disposición del personal, hora de la jornada), que no fueron controladas.

Respecto de la validez externa, las restricciones son igualmente claras. La evaluación se realizó con cinco participantes en una sola hacienda, con un régimen de tenencia compartida que no es el habitual en el sector, y los participantes habían intervenido previamente en el desarrollo. Los valores obtenidos describen ese caso y no constituyen estimaciones aplicables a otras haciendas, ni siquiera de tamaño y actividad similares.

Existe además una amenaza a la validez de constructo. El indicador de eficiencia se definió como el tiempo de ejecución de tres operaciones administrativas, lo que deja fuera

dimensiones que también integran la noción de eficiencia, como el esfuerzo de corrección posterior, el tiempo de capacitación o la carga de mantenimiento del sistema. La reducción reportada corresponde estrictamente a la dimensión medida.

Reconocer estas amenazas no invalida los hallazgos, pero sí acota lo que puede afirmarse a partir de ellos. La formulación defendible es que, en el caso estudiado, se observó una reducción sustancial de los tiempos de registro y de las inconsistencias, y que estos resultados resultan compatibles con un efecto positivo del prototipo. Establecer causalidad requeriría un diseño con grupo de control, mediciones repetidas y un número mayor de participantes.

El modo sin conexión constituye un aporte relevante para el entorno rural en el que opera la hacienda, caracterizado por una conectividad limitada. La posibilidad de registrar los embarques localmente y sincronizarlos posteriormente con el servidor permite mantener la continuidad operativa durante la jornada de embarque, cuando el acceso a internet puede ser intermitente. Esta capacidad reduce la dependencia de la conectividad en el momento crítico de la captura de datos y contribuye a la trazabilidad de la información.

Entre las limitaciones del prototipo se identifican las siguientes: la muestra de evaluación fue reducida; el funcionamiento sin conexión se aplicó principalmente al registro de embarques y no a la totalidad de los módulos; el sistema genera comprobantes internos en PDF, pero no incorpora facturación electrónica homologada por el Servicio de Rentas Internas (SRI); y el prototipo, aunque quedó desplegado en Railway y accesible por internet, no cuenta con las condiciones propias de un entorno productivo, como una política de respaldos, monitoreo o un plan de recuperación ante fallos. Estas limitaciones acotan el alcance de las conclusiones y definen líneas de trabajo futuro orientadas a ampliar la cobertura funcional, incrementar la muestra de usuarios y avanzar hacia la homologación tributaria y la puesta en producción.

En conjunto, los resultados guardan correspondencia con los objetivos planteados. El análisis de los procesos mediante observación directa y entrevista permitió identificar los requerimientos críticos previstos en el primer objetivo específico; el diseño de la arquitectura cliente-servidor, el modelo de datos y la API REST dio respuesta al segundo objetivo; el desarrollo del prototipo funcional con Expo/React Native, Spring Boot y PostgreSQL satisfizo el tercer objetivo; y la evaluación de usabilidad, con un puntaje SUS de 82,50 puntos ($n = 5$; $DE = 2,50$), superó el mínimo de 68 puntos previsto en el cuarto

objetivo. La reducción del tiempo de registro superior a 70 % y la disminución de las inconsistencias observadas son consistentes con el cumplimiento del objetivo general, en los términos ajustados de reducir (y no eliminar por completo) los errores derivados del registro manual, y dentro de los límites que impone el diseño de validación empleado.

Conclusiones y Recomendaciones

1.31. Conclusiones

Primer objetivo específico: analizar los procesos actuales de facturación e inventario mediante observación directa y entrevistas, identificando al menos cinco requerimientos funcionales críticos en un plazo máximo de dos semanas. El objetivo se cumplió. La observación directa y la entrevista dirigida permitieron caracterizar los procesos de venta, inventario y mano de obra de la Hacienda Bananera San Diego, y el levantamiento identificó como necesidades críticas el registro diferenciado por dueño y finca, el reparto proporcional de los costos del embarque, el control de existencias por propietario y la consolidación semanal de jornales, además de la operación sin conexión. El cumplimiento debe matizarse en un punto: el levantamiento se apoyó en un único informante, lo que constituye la limitación metodológica reconocida en el apartado 3.4.1.

Segundo objetivo específico: diseñar la arquitectura del sistema, definiendo los módulos de facturación e inventario, el modelo de datos y la API REST, documentando los requerimientos antes del inicio del segundo sprint. El objetivo se cumplió. Se diseñó una solución cliente–servidor en tres capas, comunicada mediante una API REST y respaldada por un modelo relacional de veintisiete tablas. La especificación de requisitos, las historias de usuario, los diagramas y el diccionario de datos permitieron representar las reglas de negocio de una hacienda con propiedad compartida, incluidas las que no admiten cobertura mediante plataformas ERP existentes, según el análisis del apartado 2.11.

Tercer objetivo específico: desarrollar un prototipo funcional con Expo/React Native en el frontend y Spring Boot con PostgreSQL en el backend, completando las funcionalidades básicas en cuatro sprints de dos semanas bajo Scrum. El objetivo se cumplió. El sistema integra autenticación con seis roles, embarques, cierre financiero, inventario por dueño, gastos, cuadrilla, jornales, planilla semanal, comprobantes internos, reportes, auditoría y registro sin conexión de embarques. Las funcionalidades adicionales se incorporaron mediante la actualización incremental del Product Backlog, sin modificar el problema ni el objetivo general aprobados. Queda pendiente, como se indica en el apartado 6.2, la cobertura mediante pruebas unitarias de las funciones de cálculo económico.

Cuarto objetivo específico: evaluar la usabilidad mediante el cuestionario SUS aplicado a un mínimo de cinco usuarios, obteniendo una puntuación mínima de 68 puntos. El objetivo se cumplió. La evaluación aplicada a cinco usuarios obtuvo una media de 82,50 puntos ($n = 5$; $DE = 2,50$; $IC\ 95\ \% \ 79,40\text{--}85,60$), superior a la referencia de 68 puntos. El resultado describe la percepción de los cinco participantes evaluados y no se plantea como estimación generalizable a otros usuarios o contextos.

Las pruebas funcionales verificaron las reglas de negocio principales, entre ellas el control de saldos, la relación entre dueños y fincas, la generación del cierre financiero, la consolidación de jornales, las restricciones por rol y la sincronización posterior de embarques registrados sin conexión.

Objetivo general: meta de eficiencia. La comparación de tiempos de registro arrojó una reducción promedio de 74,22 % ($n = 3$ operaciones; $DE = 3,67$), por encima del mínimo de 70 % establecido, y la comparación de inconsistencias registró 12 errores con el procedimiento manual frente a 1 con el prototipo, equivalente a una reducción de 91,67 %. Ambos valores corresponden a una medición puntual realizada en una sola jornada, bajo un diseño preexperimental de un solo grupo, por lo que son compatibles con un efecto positivo del prototipo sin permitir atribuirle la mejora de forma exclusiva. El aporte del sistema consiste en reducir las inconsistencias mediante validaciones automáticas y cálculos verificables, y no en eliminarlas: subsisten los errores de digitación y de selección, que ninguna validación formal puede interceptar.

En síntesis, el prototipo responde al problema planteado al centralizar la información administrativa y adaptar sus funciones a la operación real de la hacienda, sustituyendo los cuadernos físicos por registros trazables, cálculos automáticos y comprobantes verificables para cada dueño. La arquitectura y el enfoque de desarrollo adoptados podrían adaptarse a otros negocios agrícolas con procesos administrativos similares; sin embargo, su implementación requeriría un proceso previo de levantamiento y adecuación de los requerimientos específicos de cada organización, por lo que no cabe afirmar que la solución sea directamente replicable.

1.2. Recomendaciones

A partir de la experiencia obtenida durante el desarrollo del proyecto, se plantean las siguientes recomendaciones para trabajos futuros:

- Integrar la facturación electrónica homologada ante el Servicio de Rentas

Internas (SRI), incorporando la generación de comprobantes en formato XML con firma digital, de modo que los documentos que hoy emite el sistema como respaldo interno adquieran validez tributaria.

- Incorporar un módulo de liquidación contable entre socios que consolide los cierres de embarque, los gastos y los jornales en estados de cuenta periódicos por dueño, ampliando el reparto que hoy se realiza por embarque.
- Desarrollar un módulo de trazabilidad de enfunde por trabajador y lote de cinta, que permita registrar cuántos racimos enfunda cada trabajador y vincular esa información con la edad de la fruta cosechada.
- Desarrollar reportes avanzados y tableros de indicadores que faciliten el análisis histórico de la producción, los gastos, el ratio de conversión y la rentabilidad de la finca.
- Fortalecer el mecanismo de sincronización sin conexión, ampliándolo a más entidades además del embarque e incorporando la resolución automática de conflictos de datos para entornos con conectividad intermitente.
- Incorporar un módulo de gestión de compras que vincule las alertas de stock mínimo con órdenes de compra a los proveedores registrados, cerrando el ciclo de reposición del inventario.
- Repetir la medición de tiempos y errores en varias sesiones, con más de un operador y en distintos volúmenes de embarque, de modo que los valores reportados puedan expresarse como promedios con su dispersión y no como observaciones únicas.
- Modificar la ficha de observación para que registre las inconsistencias desagregadas por categoría (cálculo, duplicidad, dato omitido, relación inválida y digitación), de modo que en una medición posterior pueda cuantificarse el aporte de cada mecanismo de validación a la reducción de errores.
- Incorporar pruebas unitarias automatizadas sobre las funciones de cálculo económico (reparto proporcional de gastos, ratio de conversión por dueño y liquidación de la cuadrilla), dado que un error en ellas afecta pagos a terceros y hoy solo está cubierto por verificación manual.
- Ejecutar pruebas de concurrencia sobre las operaciones críticas antes de un despliegue en producción, en particular el cierre simultáneo de un mismo embarque desde dos dispositivos y el registro concurrente de salidas sobre el mismo ítem de inventario.
- Instrumentar la medición de la latencia de la API por endpoint y evaluar el comportamiento del cliente móvil ante conmutación de red, conectividad intermitente y sincronización parcial interrumpida.

- Ampliar el levantamiento de requerimientos a los seis perfiles de usuario mediante entrevistas o actas de aceptación firmadas, con el fin de triangular la información que en este trabajo provino de un único informante.
- Ampliar la evaluación de usabilidad a un número mayor de participantes y a usuarios ajenos al desarrollo, y contrastar los resultados en al menos una segunda unidad productiva que permita valorar la transferibilidad de la solución.

3. REFERENCIAS BIBLIOGRÁFICAS

- [1] Corporación Financiera Nacional, “Ficha sectorial: sector bananero,” Quito, Ecuador, 2024.
- [2] Ministerio de Producción, Comercio Exterior, Inversiones y Pesca, “Rendición de cuentas 2024,” Quito, Ecuador, 2024.
- [3] D. Ayora Recalde, I. Illescas Rendon y A. Reigosa Lara, “La Revolución Digital en las Pymes Ecuatorianas: Nuevos Modelos de Negocio y Oportunidades de Crecimiento,” *Investigación, Tecnología e Innovación*, vol. 16, no. 22, pp. 1–10, 2024, doi: 10.53591/iti.v16i22.1647.
- [4] Statista, “Cuota de mercado mundial de sistemas operativos móviles por trimestre,” 2025. [En línea]. Disponible en: <https://es.statista.com/estadisticas/635543/>. [Consulta: 16 de julio de 2026].
- [5] O. Sotomayor, E. Ramírez y H. Martínez, coords., *Digitalización y cambio tecnológico en las mipymes agrícolas y agroindustriales en América Latina*, Documentos de Proyectos, LC/TS.2021/65. Santiago, Chile: CEPAL/FAO, 2021. [En línea]. Disponible en: <https://www.cepal.org/es/publicaciones/46965>. [Consulta: 21 de julio de 2026].
- [6] K. C. Laudon y J. P. Laudon, *Management Information Systems: Managing the Digital Firm*, 16.^a ed. Pearson, 2020.
- [7] M. A. Rashid, L. Hossain y J. D. Patrick, “The evolution of ERP systems: A historical perspective,” en *Enterprise Resource Planning: Global Opportunities and Challenges*, Hershey, PA, USA: Idea Group Publishing, 2002, pp. 1–16.
- [8] IBM, “¿Qué es la gestión de inventario?,” 2025. [En línea]. Disponible en: <https://www.ibm.com/mx-es/think/topics/inventory-management>. [Consulta: 16 de julio de 2026].
- [9] Procesos IQ, “¿Por qué controlar el peso de cajas de banano?,” 2019. [En línea]. Disponible en: <https://www.procesos-iq.com/blog/peso-cajas-banano/>. [Consulta: 16 de julio de 2026].
- [10] BANABIO S.A., “El calendario de enfunde del banano,” 2018. [En línea]. Disponible en: <https://www.banabiosa.com/es/calendario-de-enfunde-edad-de-un-racimo-de-banano/>. [Consulta: 16 de julio de 2026].
- [11] Servicio de Rentas Internas, “Facturación electrónica,” 2025. [En línea]. Disponible en: <https://www.sri.gob.ec/facturacion-electronica>. [Consulta: 16 de julio de 2026].
- [12] S. Shylesh, “A study of software development life cycle process models,” SSRN, 2017. [En línea]. Disponible en: <https://doi.org/10.2139/ssrn.2988291>.
- [13] Spring, “Spring Boot,” [En línea]. Disponible en:

<https://spring.io/projects/spring-boot>. [Consulta: 16 de julio de 2026].

[14] Android Developers, “About Android,” 2025. [En línea]. Disponible en: <https://developer.android.com/about>. [Consulta: 16 de julio de 2026].

[15] N. Shevtsiv, D. Shvets y N. Karabut, “Prospects for using React Native for developing cross-platform mobile applications,” *Central Ukrainian Scientific Bulletin. Technical Sciences*, vol. 2, no. 33, pp. 208–213, 2019, doi: 10.32515/2664-262X.2019.2(33).208-213.

[16] Expo, “Expo documentation,” 2025. [En línea]. Disponible en: <https://docs.expo.dev/>. [Consulta: 16 de julio de 2026].

[17] M. Mythily, A. Samson Arun Raj e I. Thanakumar Joseph, “An Analysis of the Significance of Spring Boot in the Market,” en *Proceedings of the 5th International Conference on Inventive Computation Technologies (ICICT)*, pp. 1277–1281, 2022, doi: 10.1109/ICICT54344.2022.9850910.

[18] PostgreSQL Global Development Group, “PostgreSQL,” 2025. [En línea]. Disponible en: <https://www.postgresql.org/>. [Consulta: 16 de julio de 2026].

[19] M. Stonebraker y L. A. Rowe, “The design of POSTGRES,” en *Proceedings of the 1986 ACM SIGMOD International Conference on Management of Data*, pp. 340–355, 1986.

[20] K. Schwaber y J. Sutherland, *La Guía Scrum: La guía definitiva de Scrum*, 2020. [En línea]. Disponible en: <https://scrumguides.org/>. [Consulta: 16 de julio de 2026].

[21] International Organization for Standardization, *ISO 9241-11:2018 Ergonomics of human-system interaction, Part 11: Usability: Definitions and concepts*. Ginebra, Suiza: ISO, 2018.

[22] J. Brooke, “SUS: A quick and dirty usability scale,” en P. W. Jordan, B. Thomas, B. A. Weerdmeester y A. L. McClelland, eds., *Usability Evaluation in Industry*. Londres, Reino Unido: Taylor & Francis, 1996, pp. 189–194.

[23] A. Bangor, P. T. Kortum y J. T. Miller, “An Empirical Evaluation of the System Usability Scale,” *International Journal of Human–Computer Interaction*, vol. 24, no. 6, pp. 574–594, 2008, doi: 10.1080/10447310802205776.

[24] P. N. Pimple y S. K. Gupta, “Design and development of mobile based college event management system,” en *2016 International Conference on Computing, Analytics and Security Trends (CAST)*, Pune, India, pp. 165–168, 2016, doi: 10.1109/CAST.2016.90.

[25] R. Hernández-Sampieri y C. P. Mendoza, *Metodología de la investigación: las rutas cuantitativa, cualitativa y mixta*. Ciudad de México, México: McGraw-Hill, 2018.

[26] IEEE, *IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications*. Nueva York, NY, USA: Institute of Electrical and Electronics Engineers, 1998.

- [27] A. S. Tanenbaum y D. J. Wetherall, Computer Networks, 5.a ed. Boston, MA, USA: Pearson, 2011.
- [28] Google Developers, “Build an offline-first app,” Android Developers, 2025. [En línea]. Disponible en: <https://developer.android.com/topic/architecture/data-layer/offline-first>. [Consulta: 21 de julio de 2026].
- [29] S. Chacon y B. Straub, Pro Git, 2.a ed. Berkeley, CA, USA: Apress, 2014. [En línea]. Disponible en: <https://git-scm.com/book>. [Consulta: 21 de julio de 2026].
- [30] R. C. Martin, Clean Architecture: A Craftsman’s Guide to Software Structure and Design. Boston, MA, USA: Prentice Hall, 2017.
- [31] E. Evans, Domain-Driven Design: Tackling Complexity in the Heart of Software. Boston, MA, USA: Addison-Wesley, 2003.
- [32] OWASP Foundation, “OWASP Mobile Top 10,” 2024. [En línea]. Disponible en: <https://owasp.org/www-project-mobile-top-10/>. [Consulta: 31 de julio de 2026].
- [33] J. Sauro, A Practical Guide to the System Usability Scale: Background, Benchmarks & Best Practices. Denver, CO, USA: Measuring Usability LLC, 2011.

Anexos

En esta sección se presentan los instrumentos, registros, evidencias y documentos complementarios utilizados durante el levantamiento de información, el desarrollo, las pruebas y la validación del prototipo ERP móvil para la Hacienda Bananera San Diego.

3.1. Anexo A. Registro de cambios identificados durante los sprints

El presente anexo resume los requerimientos que surgieron durante las revisiones de los incrementos. Los cambios se incorporaron al Product Backlog y se desarrollaron dentro de los mismos cuatro sprints, sin modificar el título, el problema de investigación ni el objetivo general.

Tabla 43 Registro de cambios del alcance durante los sprints

Sprint	Necesidad identificada	Funcionalidad incorporada	Forma de validación
Sprint 2	Diferenciar los productos pertenecientes a cada propietario.	Inventario y movimientos diferenciados por dueño.	Revisión funcional con el propietario.
Sprint 3	Conocer cuánto recibe cada dueño después de descontar los costos del embarque.	Cierre financiero y reparto proporcional de costos.	Validación de los cálculos mediante casos representativos.
Sprint 3	Controlar los pagos de los trabajadores que participan en el embarque y en las labores semanales.	Registro de cuadrilla, jornales y planilla semanal.	Revisión del flujo de trabajo y pago con el propietario.
Sprint 3	Registrar valores adicionales entregados posteriormente por el comprador.	Reliquidación del cierre de embarque.	Comprobación del recálculo de ingresos por dueño.
Sprint 4	Permitir que cada socio consulte únicamente la información de sus fincas.	Rol de dueño, restricciones de acceso y reportes diferenciados.	Pruebas de acceso con perfiles de usuario.
Sprint 4	Mantener el registro de embarques cuando no exista conexión a internet.	Cola local en SQLite y sincronización posterior.	Prueba sin conexión y sincronización al recuperar la red.

3.2. Anexo B. Entrevista dirigida y evidencia de aplicación

Datos de aplicación

Dato	Información
Fecha	11/07/2026
Lugar	Hacienda Bananera San Diego, La Troncal,
Cañar. Entrevistado	Propietario o representante de la hacienda.
Entrevistador	Víctor Alfonso Castro Tomala.
Duración	40 minutos
Forma de evidencia Acta, fotografía autorizada, audio autorizado o firma de validación.	

Guion de preguntas

1. ¿Cómo se registra actualmente cada embarque y qué información se anota de cada dueño y finca?
2. ¿Cómo se determina el precio por caja y en qué momento se conoce el valor definitivo?
3. ¿Cómo se reparten los costos de cuadrilla, evaluador y demás gastos?
4. ¿Cómo se controla la entrada, salida y disponibilidad de productos?
5. ¿Cómo se identifica a qué dueño pertenece cada producto o gasto?
6. ¿Cómo se registran los jornales, adelantos y pagos semanales?
7. ¿Qué reportes necesita consultar cada rol?
8. ¿Qué operaciones deben funcionar cuando no existe conexión?
9. ¿Qué restricciones requieren los diferentes perfiles de usuario?
10. ¿Qué documentos internos deben generarse y compartirse?

Resumen de respuesta

Tabla 44 Resumen de la entrevista dirigida

N.º	Tema	Respuesta resumida o hallazgo
1	Registro de embarques	El embarque se registra manualmente en un cuaderno, anotando el comprador, la fecha, las cajas y los racimos aportados por cada dueño y finca, la cuadrilla participante y las observaciones de calidad. La jornada de embarque observada tuvo una duración aproximada de 8 horas.
2	Precio y producción definitiva	El precio por caja y la cantidad definitiva de cajas aceptadas se conocen generalmente al finalizar la jornada, después de conciliar la producción real entregada por cada dueño.
3	Reparto de costos	Los costos de la cuadrilla, el evaluador y los demás gastos del embarque se distribuyen entre los dueños de manera proporcional a la cantidad de cajas aportadas por cada uno.
4	Control de inventario	El inventario se controla mediante inspección visual de la bodega y anotaciones manuales de las entradas y salidas, por lo que no se dispone de un saldo actualizado de cada producto.
5	Propiedad de insumos	Cada dueño adquiere y utiliza sus propios insumos, por lo que es necesario diferenciar las existencias y los movimientos de inventario correspondientes a cada propietario.
6	Jornales y pagos	Los jornales se registran diariamente indicando el trabajador, la labor y el valor acordado. También se anotan los adelantos y, al finalizar la semana, se calcula manualmente el total que debe pagarse a cada trabajador.
7	Reportes	Se requiere consultar reportes de ventas y cierres por dueño, gastos, existencias de inventario, mano de obra semanal y productividad mediante la relación entre cajas elaboradas y racimos cortados.
8	Conectividad	La hacienda se encuentra en una zona con conectividad limitada o inexistente, por lo que el sistema debe permitir registrar operaciones sin conexión y sincronizarlas posteriormente cuando se recupere el acceso a internet.
9	Roles y restricciones	El sistema debe manejar los roles de administrador, facturador, bodeguero, capitán, mayordomo y dueño. Cada usuario debe acceder únicamente a las

N.º	Tema	Respuesta resumida o hallazgo
		funciones y datos autorizados; el dueño solo puede consultar su información y el mayordomo no debe visualizar las ganancias de los propietarios.
10	Comprobantes internos	Se requieren comprobantes internos en formato PDF para respaldar los cierres de embarque y los pagos semanales, con la posibilidad de imprimirlos o compartirlos mediante WhatsApp. Estos documentos no corresponden a facturas electrónicas autorizadas por el SRI.

3.3. Anexo C. Ficha de observación directa

La observación del proceso de embarque se realizó durante una jornada aproximada de 8 horas. Este tiempo corresponde al desarrollo completo del embarque, desde el inicio de las actividades de corte y empaque hasta la finalización de la jornada. Para la comparación de eficiencia se midió únicamente el tiempo requerido para registrar y procesar cada operación administrativa.

Tabla 45 Registro de tiempos y observaciones de las operaciones evaluadas

Operación	Modalidad	Inicio	Fin	Tiempo, errores y observaciones
Cierre de embar-que	Manual	00:00	00:30	30 minutos. Se realizaron manualmente los cálculos de ingresos, costos y distribución por dueño, requiriendo revisión de los resultados.

Cierre de embar-que	Sistema	00:00	00:07	7 minutos. El prototipo calculó automáticamente los ingresos, los costos y la utilidad correspondiente a cada dueño.
Planilla semanal	Manual	00:00	00:25	25 minutos. Se sumaron los jornales, adelantos y bonos de cada trabajador mediante registros escritos y cálculos manuales.
Planilla semanal	Sistema	00:00	00:06	6 minutos. El sistema consolidó automáticamente los jornales de la semana, descontó los adelantos y agregó los bonos registrados.
Movimiento de inventario	Manual	00:00	00:05	5 minutos. Fue necesario revisar el cuaderno y comprobar visualmente las existencias antes de registrar el movimiento.

Movimiento de inventario	Sistema	00:00	00:01:30	1 minuto y 30 segundos. La aplicación per-mitió seleccionar el producto, el dueño y el tipo de movimiento, actualizando automáticamente el saldo.
--------------------------	---------	-------	----------	---

3.4. Anexo D. Product Backlog consolidado

El Product Backlog consolidado presenta las historias implementadas, su prioridad, el sprint asignado y el estado alcanzado al finalizar el desarrollo del prototipo.

La estimación del esfuerzo de cada historia se expresó en Story Points siguiendo una escala de Fibonacci (1, 2, 3, 5, 8, 13 y 21). La asignación de puntos consideró la complejidad de la funcionalidad, la cantidad de trabajo requerida, el riesgo, las dependencias con otras historias, la incertidumbre técnica y la necesidad de pruebas; a mayor valor, mayor esfuerzo relativo estimado.

El Product Backlog distingue, además, las historias que integran el producto mínimo viable (MVP) (el núcleo funcional de autenticación por roles, embarques, cierre financiero, inventario, jornales, planilla semanal y comprobantes internos) de las funcionalidades complementarias incorporadas posteriormente, como la consulta del precio histórico del banano, que no formaban parte del alcance inicial.

Tabla 46 Estimación de Story Points y alcance del MVP

ID	Story Points	MVP
HU01	5	Sí
HU02	5	Sí
HU03	5	Sí
HU04	3	Sí
HU05	5	Sí
HU06	8	Sí
HU07	5	Sí
HU08	13	Sí
HU09	13	Sí
HU10	8	Sí
HU11	5	Sí
HU12	5	Sí
HU13	5	Sí
HU14	3	Sí

HU15	8	Sí
HU16	5	Sí
HU17	8	Sí
HU18	5	Sí
HU19	5	Sí
HU20	5	Sí
HU21	13	Sí
HU22	2	Sí
HU23	5	Sí

ID	Historia resumida	Prioridad	Sprint	Estado
HU01	Gestionar usuarios y roles.	Alta	2	Completada
HU02	Registrar dueños, fincas, hectáreas y ubicación.	Alta	2	Completada
HU03	Gestionar categorías, productos, proveedores y evaluadores.	Alta	2	Completada
HU04	Configurar el calendario de cintas.	Media	2	Completada
HU05	Asignar fincas a facturadores y dueños a mayordomos.	Alta	4	Completada
HU06	Consultar reportes administrativos y productivos.	Media	4	Completada
HU07	Consultar la bitácora de auditoría.	Media	4	Completada
HU08	Registrar un embarque y su producción por dueño y finca.	Alta	3	Completada
HU09	Finalizar el embarque y generar el cierre financiero.	Alta	3	Completada
HU10	Registrar pagos y generar el comprobante PDF.	Alta	3	Completada
HU11	Registrar una reliquidación.	Media	3	Completada
HU12	Registrar la cuadrilla del embarque.	Alta	3	Completada
HU13	Registrar entradas y salidas de inventario por dueño.	Alta	2	Completada
HU14	Consultar alertas de stock y caducidad.	Media	2	Completada
HU15	Registrar operaciones de los dueños asignados.	Alta	3	Completada

ID	Historia resumida	Prioridad	Sprint	Estado
HU16	Registrar y verificar jornales.	Alta	3	Completada
HU17	Liquidar y pagar la planilla semanal.	Alta	3	Completada
HU18	Consultar únicamente la información propia.	Alta	4	Completada
HU19	Administrar las fincas propias y su ubicación.	Media	4	Completada
HU20	Consultar resultados y aportes por embarque.	Alta	4	Completada
HU21	Registrar embarques sin conexión y sincronizar después.	Alta	4	Completada
HU22	Cambiar la contraseña durante el primer ingreso.	Alta	2	Completada
HU23	Consultar comprobantes internos autorizados por los dueños.	Media	3	Completada

Tabla 47 Product Backlog consolidado

3.5. Anexo E. Sprint Backlog y revisiones

El presente anexo resume los objetivos, actividades, incrementos, revisiones, cambios identificados y evidencias correspondientes a cada sprint del proyecto. Las evidencias comprenden actas, capturas del prototipo, resultados de pruebas y registros del repositorio de código fuente.

Sprint 1: análisis y diseño inicial

Tabla 48 Resumen del Sprint 1

Elemento	Descripción
Objetivo	Comprender los procesos administrativos de la hacienda y definir la solución inicial del prototipo ERP móvil.
Actividades	Aplicación de la entrevista dirigida, observación directa del embarque, identificación de requisitos, elaboración de historias de usuario, diseño de la arquitectura inicial y modelado de los procesos actuales.
Incremento	Documento inicial de requisitos, historias de usuario, arquitectura general, diagramas de procesos actuales y modelo preliminar de la base de datos.
Revisión	Los procesos de embarque, inventario y mano de obra fueron revisados con el propietario de la hacienda para verificar que las necesidades identificadas correspondieran con la operación real.
Cambio identificado	Se determinó la necesidad de diferenciar la producción, los gastos y las existencias de inventario según el dueño y la finca correspondiente.
Evidencia	Guion y resumen de la entrevista dirigida, ficha de observación de la jornada de embarque de aproximadamente ocho horas, diagramas iniciales y primer registro del proyecto en el repositorio de código fuente.

Sprint 1 - Análisis y diseño

Edit





↳Figura 16 Evidencia del análisis y levantamiento de requisitos del Sprint 1

Sprint 1 — Análisis y diseño inicial

Objetivo

Comprender los procesos administrativos de la Hacienda Bananera San Diego y definir la solución inicial del prototipo ERP móvil.

Duración

- Inicio: 13 de abril de 2026
- Fin: 26 de abril de 2026
- Duración: 2 semanas

Actividades realizadas

- Entrevista dirigida al propietario.
- Observación de la jornada de embarque.
- Identificación de requisitos.
- Definición de historias de usuario.
- Diseño de los procesos actuales y propuestos.
- Diseño inicial de la arquitectura.
- Elaboración del modelo de datos.

Evidencias

Entrevista

Entrevista con el propietario

Observación del embarque

Observación de la jornada de embarque

Arquitectura inicial

Arquitectura inicial

Repositorios

- [Backend Spring Boot](#)
- [Frontend Expo React Native](#)

Resultado

Se obtuvo la especificación inicial de requisitos, las historias de usuario, los modelos de procesos y la arquitectura general del sistema.

Cambio identificado

Durante la revisión se determinó la necesidad de diferenciar la producción, los gastos y el inventario según el dueño y la finca.

Sprint 2: seguridad e inventario

Tabla 49 Resumen del Sprint 2

Elemento	Descripción
Objetivo	Implementar la autenticación, el control de acceso, los catálogos generales y el módulo de inventario.
Actividades	Implementación de autenticación con JWT, cifrado de contraseñas con BCrypt, gestión de usuarios y roles, registro de dueños y fincas, categorías, productos, proveedores, entradas y salidas de inventario.
Incremento	Módulo de seguridad operativo e inventario funcional con movimientos y existencias diferenciadas por dueño.
Revisión	Se probaron el inicio de sesión, el cambio de contraseña, las restricciones por rol, las entradas y salidas de productos y el rechazo de movimientos que generaran saldos negativos.
Cambio identificado	Se incorporó el control de existencias por propietario, debido a que cada dueño compra y utiliza sus propios productos agrícolas.
Evidencia	Capturas del inicio de sesión, cambio de contraseña, listado de productos, registro de entradas y salidas, validación de saldo insuficiente y commits correspondientes al módulo de inventario.

Funcionalidades desarrolladas

Backend — Spring Boot

- Autenticación mediante JWT.
- Cifrado de contraseñas mediante BCrypt.
- Cambio obligatorio de contraseña en el primer ingreso.
- Gestión de usuarios.
- Gestión de los seis roles del sistema.
- Registro de dueños y fincas.
- Gestión de categorías.
- Gestión de productos.
- Gestión de proveedores.
- Registro de entradas de inventario.
- Registro de salidas de inventario.
- Cálculo de existencias por dueño.
- Validación de saldo insuficiente.
- Alertas de stock mínimo.
- Control de productos próximos a caducar.
- Registro de auditoría de las operaciones.

Frontend — Expo/React Native

- Pantalla de inicio de sesión.
- Pantalla de cambio obligatorio de contraseña.
- Navegación según el rol autenticado.
- Pantalla de usuarios y roles.
- Pantalla de dueños y fincas.
- Pantalla de categorías.
- Pantalla de productos.
- Formulario de entrada de inventario.
- Formulario de salida de inventario.
- Consulta del saldo por propietario.
- Mensajes de validación.
- Visualización de alertas de inventario.

Figura 17 Evidencia de autenticación e inventario correspondiente al Sprint 2

Sprint 3: embarques, costos y mano de obra

Tabla 50 Resumen del Sprint 3

Elemento	Descripción
Objetivo	Digitalizar el proceso de embarque y los costos relacionados con la producción y la mano de obra.
Actividades	Implementación del registro de embarques, producción por dueño y finca, cuadrilla, evaluador, cierre financiero, distribución proporcional de costos, gastos, jornales, adelantos, planilla semanal, pagos y comprobantes internos en PDF.
Incremento	Flujo administrativo principal integrado, desde la creación del embarque hasta su cierre financiero, junto con el control de los jornales y la liquidación semanal de los trabajadores.
Revisión	Se verificaron los cálculos de cajas, racimos, ingresos, costos y utilidad por dueño. También se validó la consolidación de jornales, el descuento de adelantos y la generación de comprobantes.
Cambios identificados	Se incorporaron la reliquidación del embarque, el registro de pagos de cuadrilla y evaluador, la planilla semanal y la generación de comprobantes internos en PDF.
Evidencia	Capturas del registro y finalización del embarque, cierre financiero, distribución proporcional de costos, pagos de cuadrilla, planilla semanal, comprobante PDF y registros del repositorio correspondientes a estos módulos.

Funcionalidades desarrolladas

Backend — Embarques y cierre financiero

- Registro del embarque.
- Registro del comprador.
- Registro de la semana y el color de cinta.
- Asignación automática del color de cinta.
- Registro de cajas y racimos por dueño.
- Registro de la finca de origen.
- Validación de la relación entre dueño y finca.
- Registro de la cuadrilla.
- Registro del evaluador.
- Finalización del embarque.
- Conciliación de la producción definitiva.
- Registro del precio definitivo por caja.
- Generación automática del cierre financiero.
- Cálculo del ingreso bruto.
- Cálculo de la utilidad por dueño.
- Distribución proporcional de los costos.
- Registro del pago a la cuadrilla.
- Registro del pago al evaluador.
- Registro de la reliquidación posterior.

Backend — Gastos y mano de obra

- Registro de gastos de embarque.
- Registro de gastos de finca.
- Registro de compra de productos.
- Generación automática de una entrada de inventario por compra.
- Registro de trabajadores.
- Registro de tareas.
- Registro de jornales.
- Registro de adelantos.
- Distribución de jornales entre varios dueños.
- Consolidación de la planilla semanal.
- Cálculo de bonos y descuentos.
- Registro del pago semanal.
- Prevención de pagos duplicados.

||

Figura 18 Evidencia del módulo de embarques y cierre financiero del Sprint 3

Funcionalidades desarrolladas

Backend — Embarques y cierre financiero

- Registro del embarque.
- Registro del comprador.
- Registro de la semana y el color de cinta.
- Asignación automática del color de cinta.
- Registro de cajas y racimos por dueño.
- Registro de la finca de origen.
- Validación de la relación entre dueño y finca.
- Registro de la cuadrilla.
- Registro del evaluador.
- Finalización del embarque.
- Conciliación de la producción definitiva.
- Registro del precio definitivo por caja.
- Generación automática del cierre financiero.
- Cálculo del ingreso bruto.
- Cálculo de la utilidad por dueño.
- Distribución proporcional de los costos.
- Registro del pago a la cuadrilla.
- Registro del pago al evaluador.
- Registro de la reliquidación posterior.

Backend — Gastos y mano de obra

- Registro de gastos de embarque.
- Registro de gastos de finca.
- Registro de compra de productos.
- Generación automática de una entrada de inventario por compra.
- Registro de trabajadores.
- Registro de tareas.
- Registro de jornales.
- Registro de adelantos.
- Distribución de jornales entre varios dueños.
- Consolidación de la planilla semanal.
- Cálculo de bonos y descuentos.
- Registro del pago semanal.
- Prevención de pagos duplicados.

Figura 19 Evidencia del módulo de jornales y planilla semanal del Sprint 3

Sprint 4: integración, acceso, modo offline y validación

Tabla 51 Resumen del Sprint 4

Elemento	Descripción
Objetivo	Integrar los módulos del sistema, completar los controles de acceso y validar el funcionamiento general del prototipo.
Actividades	Implementación de los roles finales, restricciones de información, reportes, auditoría, almacenamiento local en SQLite, sincronización posterior, pruebas funcionales, pruebas de la API REST y aplicación del cuestionario SUS.
Incremento	Prototipo integrado con control de acceso por roles, reportes, auditoría, registro de embarques sin conexión, sincronización posterior y evaluación de usabilidad.
Revisión	Se realizaron pruebas con los perfiles de administrador, facturador, bodeguero, capitán, mayordomo y dueño. También se verificó el registro sin conexión, la sincronización con el servidor y las principales reglas de negocio.
Cambios identificados	Se incorporó el rol de dueño con acceso restringido a sus propias fincas, embarques, gastos, inventario y reportes. También se completó el mecanismo de sincronización de embarques registrados sin conexión.
Evidencia	Capturas de las pruebas por rol, bitácora de auditoría, almacenamiento local, sincronización de registros pendientes, resultados de Postman o Swagger, formularios SUS de los cinco participantes y matriz consolidada de resultados.

Backend — Seguridad por roles

- Restricciones para el administrador.
- Restricciones para el facturador.
- Restricciones para el bodeguero.
- Restricciones para el capitán.
- Restricciones para el mayordomo.
- Restricciones para el dueño.
- Filtrado de información por propietario.
- Asignación de fincas a facturadores.
- Asignación de dueños a mayordomos.
- Protección de la información financiera.
- Bitácora de auditoría.

Figura 20 Evidencia de las pruebas de acceso por roles del Sprint 4

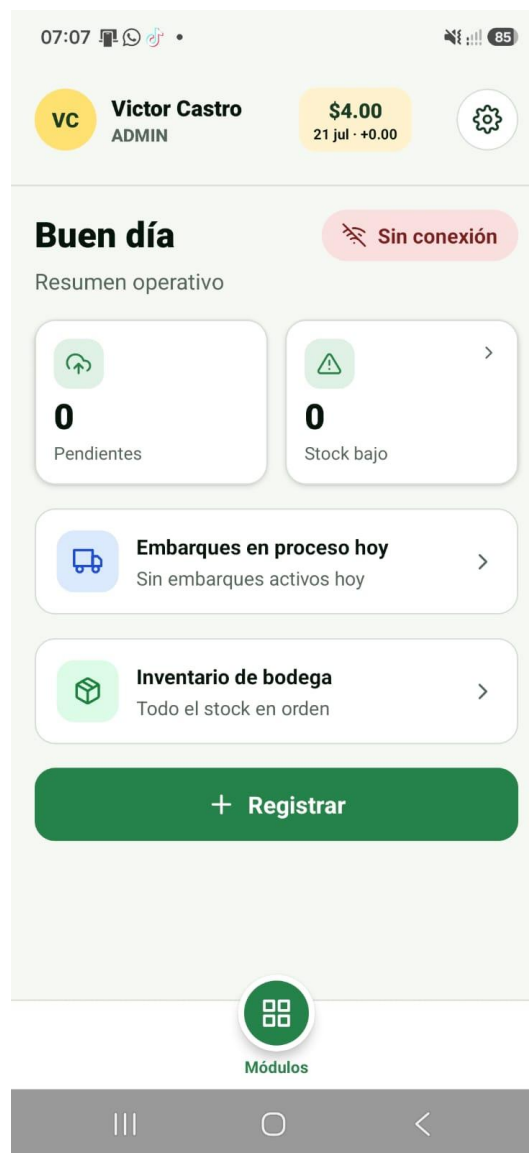


Figura 21 Evidencia del registro sin conexión y sincronización del Sprint 4

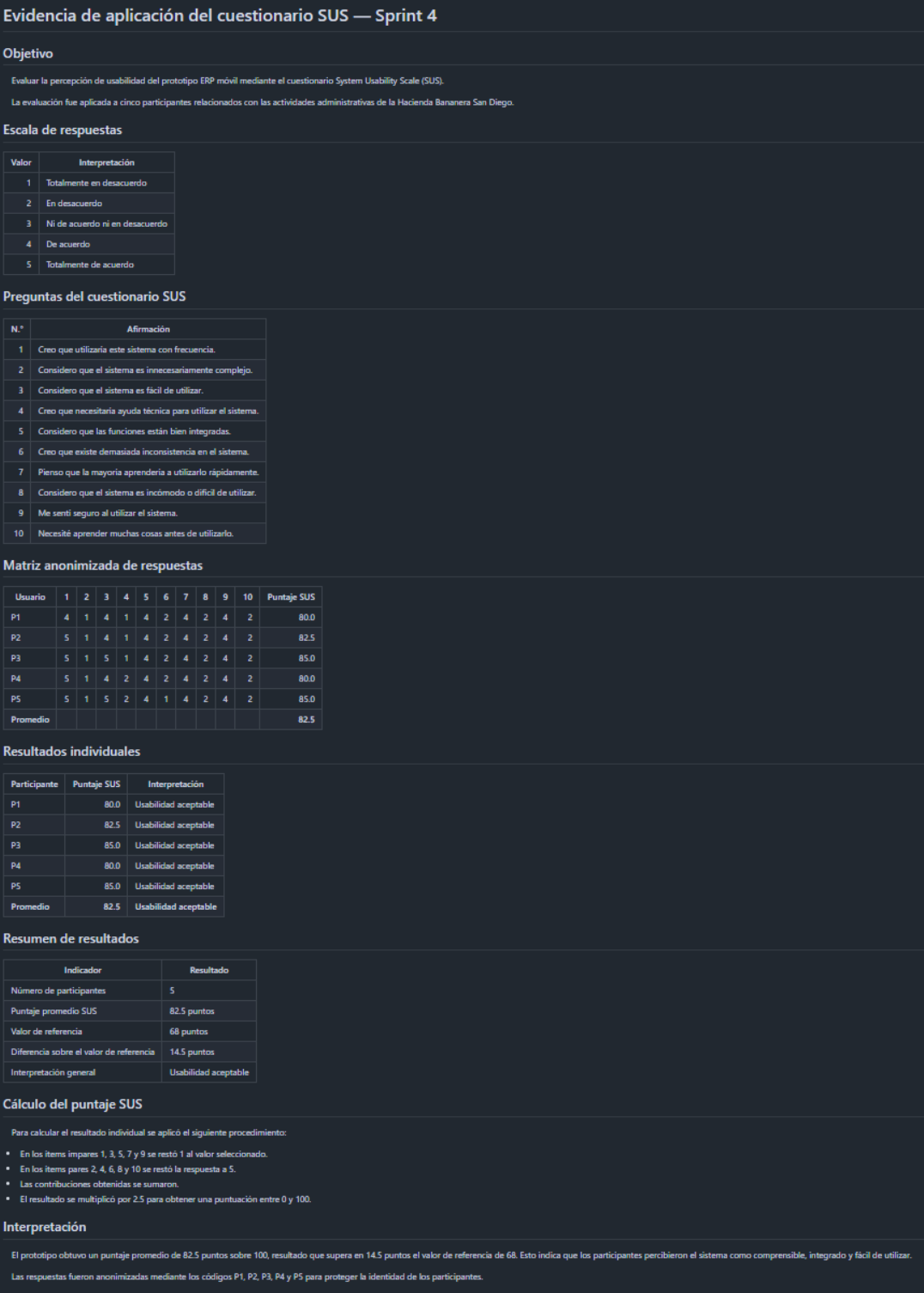


Figura 22 Evidencia de la aplicación del cuestionario SUS en el Sprint 4

3.6. Anexo F. Matriz de trazabilidad

Tabla 52 Matriz de trazabilidad entre requisitos, historias y pruebas

Requisito	Historia	Módulo	Prueba	Evidencia esperada
RF01– RF03	HU01, HU22	Autenticación	CP01– CP03	Capturas de login y cambio de clave.
RF04– RF05	HU02, HU19	Dueños y fincas	CP13, CP18	Perfil de dueño y mapa.
RF06– RF12	HU03, HU13, HU14	Inventario	CP04, CP05, CP11	Entradas, salidas y alertas.
RF13– RF19	HU04, HU08	Embarques	CP06, CP07	Validaciones y cinta.
RF20– RF27	HU09– HU12	Cierre y cuadrilla	CP08– CP10	Cierre, pagos y reliquidación.
RF28– RF29	HU15	Gastos	CP11	Gasto y entrada automática.
RF30– RF35	HU16, HU17	Jornales y planilla	CP12	Planilla semanal.
RF36	HU10	Comprobantes	CP14	PDF generado.
RF37– RF39	HU05, HU18, HU20, HU23	Seguridad por rol	CP13	Consultas restringidas.
RF40– RF41	HU07	Auditoría	CP15	Registro de bitácora.
RF42	HU21	Sincronización	CP16	Cola SQLite y confirmación.
RF43	HU16	Recordatorios	CP17	Notificación local.
RF44	HU02, HU19	Geolocalización	CP18	Mapa y coordenadas.
RF45– RF46	HU06, HU20	Reportes	CP19	Reportes por dueño.

3.7. Anexo G. Evidencias de pruebas funcionales

Las evidencias de las pruebas funcionales deben conservar el mismo identificador utilizado en el Capítulo VI. Cada archivo debe mostrar el escenario ejecutado, el resultado obtenido y, cuando corresponda, el mensaje de validación presentado por la aplicación.

Tabla 53 Matriz de evidencias de pruebas funcionales

Caso	Evidencia requerida	Archivo recomendado	Estado
CP01– CP03	Inicio de sesión válido e inválido y cambio obligatorio de contraseña.	<u>captura_cp01_login.png</u>	Adjuntado
CP04– CP05	Rechazo de una salida sin saldo suficiente y consulta de alertas de inventario.	<u>captura_cp04_inventario.png</u>	Adjuntado
CP06– CP10	Validación dueño–finca, asignación de cinta, finalización, cierre y pagos.	<u>captura_cp09_cierre.png</u>	Adjuntado
CP11– CP12	Compra de producto, entrada automática de inventario y pago de planilla.	<u>captura_cp12_planilla.png</u>	Adjuntado
CP13	Restricciones de acceso según rol y propietario.	<u>captura_cp13_roles.png</u>	Adjuntado
CP14	Generación y compartición del comprobante interno en PDF.	<u>captura_cp14_pdf.png</u>	Adjuntado
CP15	Registro de usuario, acción, entidad y fecha en la bitácora de auditoría.	<u>captura_cp15_auditoria.png</u>	Adjuntado
CP16	Registro local del embarque y sincronización posterior con el servidor.	<u>captura_cp16_offline.png</u>	Adjuntado
CP17	Notificación local de jornales pendientes de verificación.	<u>captura_cp17_recordatorio.png</u>	Adjuntado
CP18	Selección de coordenadas de la finca mediante el mapa interactivo.	<u>captura_cp18_mapa.png</u>	Adjuntado
CP19	Consulta de reportes por dueño y visualización de aportes pendientes.	<u>captura_cp19_reportes.png</u>	Adjuntado

3.8. Anexo H. Evidencias de pruebas de la API REST

Las pruebas de la API REST deben documentarse mediante capturas de Postman o Swagger. En cada evidencia se debe mostrar el método HTTP, la ruta, el cuerpo de la solicitud cuando corresponda, el código de respuesta y el mensaje devuelto por el backend. Los tokens, contraseñas y demás datos sensibles deben ocultarse antes de incorporar las imágenes al documento.

Tabla 54 Registro de evidencias de Postman o Swagger

Módulo	Endpoint	Método	Respuesta esperada	Evidencia
Autenticación	/api/auth/login	POST	200 / 401	Swagger UI
Inventario	/api/inventario/movimientos	POST	201 / 400	Swagger UI
Embarques	/api/embarques	POST	201 / 400	Swagger UI
Cierre	/api/embarques/{id}/finalizar	POST	200 / 409	Swagger UI
Planilla	/api/pagos-jornales-semanales/pagar	POST	200 / 409	Swagger UI
Auditoría	/api/auditoria	GET	200 / 403	Swagger UI
Sincronización	/api/sync/embarques	POST	200 / 400	Swagger UI

Nota: las rutas y códigos HTTP deben contrastarse con la documentación Swagger de la versión final del backend. Cuando la respuesta observada sea diferente, debe registrarse el valor real devuelto por la API.

3.9. Anexo I. Instrumento y resultados System Usability Scale

Los participantes calificaron cada afirmación en una escala de 1 a 5, donde 1 corresponde a “totalmente en desacuerdo” y 5 a “totalmente de acuerdo”.

Tabla 55: Ítems del cuestionario SUS

N.º	Afirmación
1	Creo que utilizaría este sistema con frecuencia.
2	Considero que el sistema es innecesariamente complejo.
3	Considero que el sistema es fácil de utilizar.
4	Creo que necesitaría ayuda técnica para utilizar el sistema.
5	Considero que las funciones están bien integradas.
6	Creo que existe demasiada inconsistencia en el sistema.
7	Pienso que la mayoría aprendería a utilizarlo rápidamente.
8	Considero que el sistema es incómodo o difícil de utilizar.
9	Me sentí seguro al utilizar el sistema.
10	Necesité aprender muchas cosas antes de utilizarlo.

Tabla 55 Matriz anonimizada de respuestas SUS

Usuario	1	2	3	4	5	6	7	8	9	10	SUS
P1	4	1	4	1	4	2	4	2	4	2	80,0
P2	5	1	4	1	4	2	4	2	4	2	82,5
P3	5	1	5	1	4	2	4	2	4	2	85,0
P4	5	1	4	2	4	2	4	2	4	2	80,0
P5	5	1	5	2	4	1	4	2	4	2	85,0

El promedio obtenido por los cinco participantes fue de 82,5 puntos sobre 100. Las respuestas fueron anonimizadas mediante los códigos P1, P2, P3, P4 y P5 para proteger la identidad de los participantes. Los formularios aplicados se incorporan como evidencia del proceso de evaluación de usabilidad.

3.10. Anexo J. Medición de tiempos y errores

La observación del proceso de embarque se desarrolló el 7 de julio de 2026, durante una jornada aproximada de ocho horas. Este tiempo corresponde a la ejecución completa de las actividades de corte, selección, empaque y cierre. La comparación de eficiencia considera únicamente el tiempo empleado en las operaciones administrativas evaluadas.

Tabla 56 Hoja consolidada para el cálculo de reducción de tiempos

Operación	Manual	Sistema	Reducción
Cierre de embarque	30,00 min	7,00 min	76,67 %
Planilla semanal	25,00 min	6,00 min	76,00 %
Movimiento de inventario	5,00 min	1,50 min	70,00 %
Promedio	—	—	74,22 %

Tabla 57 Hoja consolidada para la comparación de errores

Indicador	Valor
Errores observados en el procedimiento manual	12
Errores observados con el prototipo	1
Reducción calculada	91,67 %

Los resultados muestran que el cierre de embarque pasó de 30 minutos con el procedimiento manual a 7 minutos con el prototipo, lo que representa una reducción del 76,67 %. La planilla semanal pasó de 25 a 6 minutos, con una reducción del 76,00 %, mientras que el movimiento de inventario pasó de 5 a 1,5 minutos, equivalente a una reducción del 70,00 %.

El promedio simple de reducción de las tres operaciones fue del 74,22 %, superando el mínimo del 70 % establecido en el objetivo general. En la comparación de errores se registraron 12 inconsistencias con el procedimiento manual y una con el prototipo, lo que representa una reducción del 91,67 %.

3.11. Anexo K. Manual breve de usuario

El manual completo de usuario con capturas en alta resolución se encuentra disponible en el siguiente enlace:

[Ver manual completo en MediaFire](#)

3.12. Anexo L. Acta de validación del propietario

En la ciudad de La Troncal, a los 11 días del mes de julio de 2026, el propietario o representante de la Hacienda Bananera San Diego declara haber revisado el prototipo ERP móvil desarrollado como parte del presente trabajo de titulación.

Durante la revisión se presentaron los módulos de embarques, cierre financiero, inventario, gastos, cuadrilla, jornales, planilla semanal, comprobantes, reportes, auditoría y registro sin conexión.

El representante deja constancia de que las funcionalidades descritas corresponden a necesidades identificadas durante el levantamiento de información y las revisiones de los sprints. Asimismo, manifiesta que el prototipo representa los principales procesos administrativos observados en la hacienda y que su funcionamiento fue presentado para validación.

Esta acta constituye una evidencia de validación funcional y no reemplaza las pruebas técnicas, las mediciones de eficiencia ni la evaluación de usabilidad mediante el cuestionario SUS.

Victor Hugo Castro Raiban

Cargo o relación con la hacienda: **Dueño**

Fecha: 11/07/2026

3.13.

3.14. **Anexo M. Diagramas complementarios**

Los diagramas en formato fuente y las imágenes en alta resolución están disponibles en el repositorio de documentación del proyecto. El acceso está restringido a los propietarios del proyecto, dado que se trata de un repositorio privado.

[Ver repositorio en GitLab](#)

3.15. **Anexo N. Acta de aceptación de requerimientos por perfil de usuario**

Este anexo contiene el modelo de acta empleado para validar los requerimientos funcionales con cada perfil de usuario del sistema. Se aplicó un acta independiente por rol, con el fin de verificar que las funciones asignadas a cada perfil correspondan a las tareas que esa persona ejecuta realmente en la operación de la hacienda.

El instrumento se estructura en cuatro bloques. El primero identifica al participante y su rol dentro del sistema. El segundo enumera los requerimientos funcionales asociados a ese rol, tomados de la especificación del apartado 5.1.3, y solicita valorar cada uno según su correspondencia con la tarea real, con tres opciones: corresponde, corresponde con observaciones o no corresponde. El tercero recoge las funciones que el participante considera necesarias y que no figuran en la lista. El cuarto registra la conformidad con nombre, cédula, firma y fecha.

Encabezado del acta. Universidad Católica de Cuenca, Unidad Académica de Informática, Ciencias de la Computación e Innovación Tecnológica. Trabajo de titulación: Prototipo ERP móvil para la gestión de facturación e inventario en una hacienda bananera de La Troncal, Cañar. Acta de aceptación de requerimientos funcionales.

Bloque 1, identificación. Nombre del participante. Número de cédula. Rol asignado en el sistema (administrador, dueño, mayordomo, facturador, bodeguero o capitán). Función que desempeña en la hacienda. Tiempo de experiencia en esa función. Fecha de la sesión de validación.

Bloque 2, validación de requerimientos. Para cada requerimiento funcional asociado al rol se registra el código del requerimiento, su descripción y la valoración del participante. Las tres opciones de valoración son: corresponde a la tarea que ejecuto; corresponde con observaciones, en cuyo caso se anota la observación; y no corresponde a mi función, con indicación del motivo. Se incluye una columna de comentarios abierta.

Bloque 3, requerimientos no contemplados. Espacio abierto en el que el participante describe funciones que necesita para su trabajo y que no aparecen en la lista revisada, con una indicación de prioridad entre alta, media y baja.

Bloque 4, conformidad. Declaración de que el participante revisó los requerimientos asociados a su rol y que la valoración consignada refleja su criterio, seguida de nombre, cédula, firma y fecha. Se incluye además la firma del investigador que aplicó el instrumento.

Las actas suscritas se conservan en formato físico y digitalizado, y se ponen a disposición del tribunal para su verificación durante la defensa.

Victor Alfonso Castro Tomala portador(a) de la cédula de ciudadanía N° **0928595479**. En calidad de autor/a y titular de los derechos patrimoniales del trabajo de titulación **Prototipo ERP móvil para la gestión de facturación e inventario en una hacienda bananera de La Troncal, Cañar** de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos y no comerciales. Autorizo además a la Universidad Católica de Cuenca, para que realice la publicación de éste trabajo de titulación en el Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Cuenca, **03 de septiembre de 2026**



F:

Victor Alfonso Castro Tomala

C.I. 0928595479