



**Diseño e Implementación de un Sistema Inteligente de Gestión de Inventario y Control de
Servicios en una Vulcanizadora de Transporte**

Toapanta Paez, Darwin Andrés

Departamento de Ciencias de la Computación

Carrera en Tecnologías de la Información

Trabajo de integración curricular, previo a la obtención del título de Ingeniero en Tecnologías de
la Información

Mgtr. Ing. Chuquitarco Velasco, Kevin Jair

30 de julio de 2026



Certificado de análisis
Compilatio Magister+ | ESPE-ECU

Diseño e Implementación de un Sistema Inteligente de Gestión de Inventario y Control de Servicios en una Vulcanizadora de Transporte

ID : b29beec9bf846731123c2ba17e8f15d1e662c672



9%
Textos
sospechosos

Nombre del fichero : Diseño e Implementación de un Sistema Inteligente de Gestión de Inventario y Control de Servicios en una Vulcanizadora de Transporte.txt
Tamaño del archivo original : 8,24 MB
Número de palabras : 23.278

Depositante : KEVIN JAIR CHUQUITARCO VELASCO
Fecha de depósito : 24 de agosto de 2026
Tipo de carga : interface
fecha de fin de análisis : 24 de agosto de 2026

Resumen (sección 1/2)

Localización de los textos sospechosos en el documento :



Incluido en el porcentaje de textos sospechosos :

Similitudes 1%

Pasajes con similitudes a fuentes encontradas en diferentes colecciones.



Detección de IA 2%

Textos estilísticamente próximos a un texto generado por una IA.

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



Idiomas no reconocidos 6%

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

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



No incluido en el porcentaje de textos sospechosos :

Textos entre comillas 1%

Pasajes entre comillas, a menudo indicativos de una cita.

Mgtr. Ing. Chuquitarco Velasco, Kevin Jair

Director



Departamento de Ciencias de la Computación

Carrera en Tecnologías de la Información

Certificación

Certifico que el trabajo de integración curricular: **“Diseño e Implementación de un Sistema Inteligente de Gestión de Inventario y Control de Servicios en una Vulcanizadora de Transporte”** fue realizado por el señor **Toapanta Paez, Darwin Andres**, el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE; habiéndose efectuado una revisión integral de forma y de fondo conforme a los lineamientos y formatos institucionales vigentes. Además, fue revisado y analizada en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Santo Domingo, 30 de Julio de 2026

Mgtr. Ing. Chuquitarco Velasco, Kevin Jair

Director

C.C.: 0550111595



Departamento de Ciencias de la Computación

Carrera en Tecnologías de la Información

Responsabilidad de Autoría

Yo, **Toapanta Paez, Darwin Andres**, con cédula de ciudadanía n° 2300412323, declaro que el contenido, ideas y criterios del trabajo de integración curricular: **Diseño e Implementación de un Sistema Inteligente de Gestión de Inventario y Control de Servicios en una Vulcanizadora de Transporte** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Santo Domingo, 30 de Julio de 2026

Toapanta Paez Darwin Andres

C.C.: 2300412323



Departamento de Ciencias de la Computación

Carrera en Tecnologías de la Información

Autorización de Publicación

Yo, Toapanta Paez Darwin Andrés, con cédula de ciudadanía n° 2300412323, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: **Diseño e Implementación de un Sistema Inteligente de Gestión de Inventario y Control de Servicios en una Vulcanizadora de Transporte** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Santo Domingo, 30 de Julio de 2026

Toapanta Paez Darwin Andres

C.C.: 2300412323

Dedicatoria

Dedico este trabajo a toda mi familia, por ser el apoyo principal y constante durante mi formación académica, y estar presentes en cada etapa de este proceso, mostrando paciencia, confianza y motivación. A mis padres, por enseñarme que los objetivos más importantes requieren de disciplina, perseverancia y responsabilidad.

Agradecimiento

Expreso mi agradecimiento a la Universidad de las Fuerzas Armadas ESPE y a los docentes de la carrera por aportar los conocimientos, fundamentos técnicos y criterios profesionales para el desarrollo de este trabajo. A mi familia, por su respaldo constante y la confianza demostrada; A mi pareja y amigos por su motivación y apoyo durante el desarrollo de este trabajo. A todos los que de alguna u otra manera, contribuyeron en la culminación de este trabajo, muchas gracias.

Índice de contenidos

Certificación	3
Dedicatoria.....	6
Agradecimiento	7
Resumen	14
Abstract.....	15
Capítulo I	16
Introducción	16
Planteamiento del problema.....	17
Formulación del problema o pregunta de investigación.....	18
Objetivos.....	19
Objetivo general	19
Objetivos específicos.....	19
Justificación	19
Capítulo II	20
Estado del arte.....	20
Transformación digital de las PyMES.....	21
Sistemas web para el control de inventario en organizaciones	22
Trazabilidad digital mediante códigos QR	23
Sistemas integrados de gestión de servicios, alertas de stock y órdenes de trabajo.....	24
Vacíos reconocidos y posicionamiento del presente estudio.....	25
Marco Teórico	26
Arquitectura web y el modelo cliente/servidor	26
APIs RESTful	27
Sistemas de información para la gestión de inventario.....	27
Trazabilidad digital mediante códigos QR	29

Atributos de calidad	30
Tecnologías de desarrollo web	31
React y el paradigma de componentes.....	31
Node.js y Express como backend.....	32
PostgreSQL.....	32
Prisma ORM.....	33
Autenticación y autorización con JSON Web Tokens (JWT)	34
Metodología Kanban	34
Ciclo de vida del software	36
Function Calling y Modelos de lenguaje	37
Control de acceso basado en roles (RBAC).....	38
Autenticación delegada mediante OAuth 2.0 y OpenID Connect	38
Pasarelas de pago electrónico e integración server-to-server	39
Marco Conceptual.....	39
Conceptos del dominio del negocio.....	39
Vulcanizadora.....	39
Neumático	40
Servicios técnicos.....	40
Orden de trabajo	41
Inventario e insumos	41
Cliente y flota de transporte.....	42
Hoja de vida digital del neumático	42
Capítulo III	43
Metodología.....	43
Marco metodológico investigativo	43
Enfoque de investigación.....	43

Tipo y diseño de investigación.....	44
Población y muestra.....	44
Técnica e instrumento	45
Procedimiento de evaluación.....	45
Consideraciones Éticas	46
Marco metodológico de desarrollo	47
Análisis.....	47
Identificación de requerimientos del sistema	52
Historias de Usuario	53
Requisitos no funcionales.....	72
Diseño	75
Implementación	82
Fases del desarrollo.....	94
Pruebas	95
Estrategia de pruebas	96
Capítulo IV	98
Análisis y discusión de resultados.....	98
Grado de cumplimiento del sistema.....	98
Plan de pruebas	104
Casos de prueba por módulo	104
Análisis estático de código	110
Análisis dinámico.....	112
Capítulo V	115
Conclusiones	115
Recomendaciones	116
Bibliografía.....	117

Índice de tablas

Tabla 1 Ciclo de desarrollo del software	36
Tabla 2 Entrevista semiestructurada.....	49
Tabla 3 Organización del proyecto	53
Tabla 4 Definición de componentes en la plantilla usada para Historias de Usuario	53
Tabla 5 Historias de Usuario.....	55
Tabla 6 Distribución de historias de usuario.	70
Tabla 7 Historias de usuario planificadas por iteración.	71
Tabla 8 Requerimientos no funcionales de Superllanta.	72
Tabla 9 Backlog de ejecución de la Iteración 1	82
Tabla 10 Backlog de ejecución de la Iteración 2	86
Tabla 11 Backlog de ejecución de la Iteración 3	88
Tabla 12 Backlog de ejecución de la Iteración 4	89
Tabla 13 Backlog de ejecución de la Iteración 5	91
Tabla 14 Backlog de ejecución de la Iteración 6	93
Tabla 15 Matriz de cumplimiento de requisitos no funcionales	100
Tabla 16 Evaluación matemática de los requisitos no funcionales.....	103
Tabla 17 Casos de prueba por Módulo.....	104
Tabla 18 Resultados de ejecución por módulo	109
Tabla 19 Métricas de análisis estático de código	110
Tabla 20 Métricas de análisis dinámico.	113

Índice de figuras

Figura 1 Características modelo ISO/IEC 25010:2011 seleccionadas para la evaluación del software Superllanta.	31
Figura 2 Registro manual para registrar el ingreso de neumáticos.	51
Figura 3 Registro manual de ventas de neumáticos.	51
Figura 4 Registro manual de arreglos e insumos usados.	52
Figura 5 Arquitectura lógica del sistema Superllanta.	78
Figura 6 Arquitectura física del despliegue del sistema Superllanta.	80
Figura 7 <i>Modelo base de datos</i>	80
Figura 8 Prototipo pantalla de inicio de sesión del sistema Superllanta.....	82
Figura 9 Resultado de la Iteración 1: núcleo funcional del sistema Superllanta.....	85
Figura 10 Gestión del inventario de insumos implementada en la Iteración 1	86
Figura 11 Resultado de la Iteración 2: gestión de clientes y vehículos	88
Figura 12 Resultado de la Iteración 3: orden de trabajo con múltiples servicios	89
Figura 13 Resultado de la Iteración 4: vista responsive del módulo de órdenes de trabajo.	90
Figura 14 Navegación responsive del sistema en dispositivo móvil.....	90
Figura 15 Vista responsive del portal del cliente en dispositivo móvil	91
Figura 16 Resultado de la Iteración 5: refinamiento de órdenes asignadas al técnico	92
Figura 17 Resultado de la Iteración 6: consultas administrativas mediante inteligencia artificial	94
Figura 18 Resultados de la ejecución de pruebas unitarias (Jest).	97
Figura 19 Ejecución de las pruebas de integración del sistema mediante Jest.....	98
Figura 20 Control de acceso por rol: redirección del técnico ante una ruta restringida	106
Figura 21 Solicitud de vinculación de identidad para un cliente no asociado	106
Figura 22 Registro exitoso de un nuevo insumo en el inventario	107
Figura 23 Generación de alerta por stock bajo	107

Figura 24 Generación de código QR único para un neumático	107
Figura 25 Consulta de la hoja de vida digital mediante código QR	108
Figura 26 Validación de datos obligatorios al registrar una orden de trabajo	108
Figura 27 Confirmación de una orden completada y pagada.....	109
Figura 28 Resultado del análisis estático: confiabilidad y seguridad del código.....	112
Figura 29 Métricas de consumo de recursos y solicitudes del backend en Render	114
Figura 30 Resultados de la prueba de carga del backend mediante k6.....	115

Resumen

El objetivo de este proyecto de integración curricular fue desarrollar e instalar una aplicación web para la gestión de inventarios, trazabilidad de neumáticos y el control de servicios del negocio Superllanta ubicado en Santo Domingo de los Colorados, Ecuador. La investigación se desarrolló bajo un enfoque mixto, de tipo aplicado y descriptivo. Para identificar los procesos operativos, problemas y requerimientos del negocio, se emplearon la observación directa y la entrevista al propietario del negocio. El desarrollo del software se gestionó mediante la metodología Kanban, usando Jira para organizar el Product Backlog y ejecutar seis iteraciones, en las cuales fueron implementadas 47 historias de usuario, distribuidas en 13 módulos funcionales. El sistema se desarrolló con una arquitectura cliente-servidor organizada en cuatro capas, utilizando React, Vite y Tailwind CSS para el frontend; Node.js y Express en el backend; Prisma ORM y PostgreSQL para la gestión de los datos; y Auth0 para autenticación y control de acceso basado en roles. Entre las principales funcionalidades implementadas se encuentran el control de inventario, la gestión de clientes y vehículos, órdenes de trabajo, reparaciones, vulcanizados, reencauches, trazabilidad mediante códigos QR, impresión de etiquetas, gestión de pagos, notificaciones y consultas administrativas asistidas por inteligencia artificial. El Product Backlog alcanzó un cumplimiento del 100% y la evaluación de los requisitos no funcionales evidenció resultados satisfactorios en usabilidad, eficiencia de desempeño, disponibilidad, seguridad, portabilidad y mantenibilidad. En la evaluación de escalabilidad, la variación del percentil 95 entre los escenarios de 1 y 15 usuarios virtuales fue del 94,74%, superando así el límite establecido del 20%; sin embargo, ambos escenarios mantuvieron una tasa de errores del 0%. Los resultados obtenidos evidencian que el sistema permite digitalizar y centralizar procesos que anteriormente se gestionaban de forma manual en Superllanta.

Palabras clave: sistema web, gestión de inventario, trazabilidad, Kanban, vulcanizadora

Abstract

The objective of this capstone project was to develop and install a web application for inventory management, tire traceability, and service control for the Superllanta business located in Santo Domingo de los Colorados, Ecuador. The research was conducted using a mixed-methods approach of an applied and descriptive nature. To identify the business's operational processes, problems, and requirements, direct observation and an interview with the business owner were employed. The software development was managed through the Kanban methodology, utilizing Jira to organize the Product Backlog and execute six iterations, during which 47 user stories were implemented, distributed across 13 functional modules. The system was developed with a client-server architecture organized into four layers, using React, Vite, and Tailwind CSS for the frontend; Node.js and Express for the backend; Prisma ORM and PostgreSQL for data management; and Auth0 for authentication and role-based access control. Among the main functionalities implemented are inventory control, customer and vehicle management, work orders, repairs, vulcanization, retreading, traceability via QR codes, label printing, payment processes, notifications, and AI-assisted administrative queries. The Product Backlog achieved 100% completion, and the evaluation of non-functional requirements yielded satisfactory results in usability, performance efficiency, availability, security, portability, and maintainability. In the scalability evaluation, the 95th percentile variation between the 1 and 15 virtual user scenarios was 94.74%, thus exceeding the established limit of 20%; however, both scenarios maintained an error rate of 0%. The results obtained demonstrate that the system enables the digitization and centralization of processes that were previously managed manually at Superllanta.

Keywords: web system, inventory management, traceability, Kanban, vulcanization

Capítulo I

Introducción

El presente trabajo de integración curricular surge de la necesidad de modernizar ciertos procesos operativos de Superllanta, un negocio ubicado en Santo Domingo de los Colorados, Ecuador, que se especializa en brindar servicios de reparación, vulcanización, reencauche y venta de neumáticos de vehículos de transporte pesado y liviano.

En el presente trabajo utilizaremos el término neumático para referirse al componente de caucho de la rueda. Aunque en el contexto comercial local se empleará frecuentemente el término llanta, en la redacción técnica del documento se mantendrá el término neumático. La denominación llanta se conservará solamente cuando corresponda a expresiones usadas literalmente por participantes o a denominaciones propias del contexto comercial del negocio.

Como solución a la problemática, se diseñó e implementó un sistema web orientado a la automatización de procesos relacionados con el inventario, servicios técnicos y la trazabilidad de neumáticos. Una de las principales funcionalidades consiste en la asignación de un código QR único para cada neumático registrado durante los servicios de reparación y vulcanizado. Este identificador permite consultar su hoja de vida digital, en la que se mantiene el historial de las intervenciones realizadas y cuya información puede ser consultada desde un dispositivo móvil con conexión a internet y capacidad para leer códigos QR [1].

El desarrollo del sistema se gestionó con la metodología Kanban y un tablero digital elaborado en Jira. Este enfoque permitió organizar las historias de usuario, visualizar el estado de las tareas y adaptar progresivamente el Product Backlog a los requerimientos identificados durante el desarrollo y pruebas unitarias del sistema [2].

Para la implementación se utilizaron tecnologías de desarrollo web como React, Vite y Tailwind CSS para el Frontend; Node.js y Express para el Backend; Prisma ORM para el acceso a datos y PostgreSQL como sistema gestor de base de datos. La autenticación se implementó mediante tokens JWT emitidos por Auth0, incluyendo permisos basados en roles

para el sistema. El Capítulo I presenta la introducción, el planteamiento del problema, los objetivos general y específicos y la justificación del proyecto [3].

Planteamiento del problema

En la actualidad, uno de los principales desafíos para las medianas y pequeñas empresas del sector automotriz es la transformación digital. La demanda de servicios de reparación de neumáticos para transportes pesados y livianos se ha incrementado en muchas provincias del país, y especialmente en ciudades estratégicamente ubicadas, como Santo Domingo de los Colorados, debido a su conexión vial con varias ciudades del Ecuador. En este contexto, las vulcanizadoras y talleres especializados en reparación de neumáticos tienen un papel importante dentro del mantenimiento de los vehículos y la seguridad vial.

Gran parte de las vulcanizadoras del país presentan bajos niveles de digitalización de sus procesos operativos o administrativos. Varios negocios siguen realizando su gestión, y sus registros de forma manual en hojas de papel, y en ocasiones genera un desorden en la información, pérdida de datos, además de un descontrol en el inventario tanto en los insumos como en los productos [4].

En este contexto se encuentra la vulcanizadora Superllanta se dedica a ofrecer varios servicios como arreglos, cambios, vulcanizados y reencauches de neumáticos, para los vehículos de transporte pesado y liviano. El negocio en la actualidad registra todos estos servicios y los controles de las operaciones de forma manual, además no lleva un control del inventario de herramientas e insumos, por lo que no cuenta con un sistema informático que le permita gestionar toda esta información del negocio.

Esto ha generado diversos problemas en la administración. Primero se puede evidenciar el descontrol del inventario de insumos como los parches, pegamento, válvulas, materiales para el servicio de vulcanizado, y los neumáticos que ingresan al proceso de reencauche, además de neumáticos que se encuentran en venta o ya vendidos. Esto ocasiona desabastecimiento de productos, además de falta de herramientas por pérdida o falta de

disponibilidad, los cuales son sumamente necesarios para el proceso de los servicios, lo cual puede generar pérdidas económicas e insatisfacción de los clientes.

Otro problema se relaciona con la atención al cliente. La falta de insumos o herramientas necesarias para realizar los servicios puede impedir la ejecución oportuna de los trabajos, generar insatisfacción y ocasionar la pérdida de clientes e ingresos. Esta situación también afecta a los clientes propietarios de flotas de transporte, ya que la gestión manual de varias órdenes de trabajo por vehículo puede generar confusiones durante la facturación. Asimismo, los recibos correspondientes a la venta de neumáticos reencauchados pueden extraviarse, lo que dificulta la verificación de garantías y puede provocar desacuerdos entre el cliente y el negocio.

En consecuencia, Superllanta presenta ciertas limitaciones en el control de inventario, y la gestión administrativa de los servicios técnicos que ofrece, lo cual afecta significativamente la eficiencia operativa, la buena calidad del servicio hacia los clientes y la competitividad del taller dentro del sector automotriz en Santo Domingo de los Colorados.

Formulación del problema o pregunta de investigación

Después de considerar problemas de administración en la vulcanizadora Superllanta, surge la necesidad de analizar la creación de un sistema web de gestión, el cual ayudaría a mejorar el control de inventarios y la trazabilidad de los servicios que realiza el negocio. En este plano, planteamos las siguientes preguntas para una mejor investigación:

¿De qué forma un sistema web de gestión, basado en trazabilidad digital mediante códigos QR, puede mejorar el control de inventario y mejorar el servicio técnico del negocio?, ¿Qué procesos activos en la vulcanizadora Superllanta se pueden automatizar con la implementación de un sistema web, para poder reducir las pérdidas derivadas de una gestión inadecuada del inventario?, ¿De qué manera la implementación de códigos QR por neumático permite generar una hoja de vida digital para registrar y consultar el historial de los servicios

realizados en Superllanta?, ¿De qué manera un sistema con control de stock y alertas automáticas puede optimizar la gestión de insumos y prevenir el desabastecimiento?

Objetivos

Objetivo general

Desarrollar un sistema para la gestión de inventario, trazabilidad de neumáticos y control de servicios para una vulcanizadora de transporte.

Objetivos específicos

- Analizar los procesos de Superllanta e identificar los requisitos funcionales y no funcionales para el sistema, con la observación directa y entrevista al dueño del negocio.
- Diseñar el modelo de base de datos relacional, y las arquitecturas del sistema, generando las entidades, atributos y relaciones necesarias para representar los procesos de gestión del inventario del negocio, clientes y neumáticos.
- Implementar módulos funcionales para el sistema, inventario, servicios, trazabilidad digital por QR y control de acceso por roles, usando lenguajes y frameworks de desarrollo web, para automatizar la gestión de Superllanta.
- Evaluar el funcionamiento y usabilidad del sistema mediante pruebas de campo con usuarios finales y métricas de calidad.

Justificación

La vulcanizadora Superllanta desde hace varios años ofrece varios servicios dentro del negocio, como por ejemplo reparaciones de neumáticos, vulcanizados y reencauches, para vehículos de transporte, ya sean livianos o pesados. Sin embargo, durante este tiempo el registro de todos los servicios que realiza el negocio, se ha hecho en hojas de papel, sin un mecanismo sistemático de organización y gestión. Dicha situación genera varios problemas administrativos y operativos.

La falta de un sistema informático provoca varios problemas en las operaciones y servicios realizados diariamente, entre ellos un evidente descontrol del inventario de los

insumos como los parches, válvulas para tipos de rines, materiales para el servicio de vulcanizado, además de la venta de llantas reencauchadas, lo cual genera desabastecimiento en momentos críticos, interrupciones del servicio y consecuentes pérdidas económicas; dificultades para conocer el historial técnico de cada neumático reparado, lo que genera conflictos o desacuerdos con los clientes para ofrecer garantías del trabajo documentado, así como dificultades para identificar cuando un neumático que ha recibido varios servicios ha alcanzado el límite de reparaciones seguras; dificultades para gestionar los servicios correspondientes a clientes empresariales de transporte liviano o pesado. La implementación de un sistema web de gestión no constituye únicamente una mejora, sino una necesidad para el crecimiento del negocio.

Capítulo II

Estado del arte

La transformación de procesos administrativos manuales hacia medios digitales representa uno de los retos organizacionales y tecnológicos más críticos para las pequeñas y medianas empresas (PyMES) en la actualidad. En el sector de servicios automotrices y mantenimiento de transporte pesado, esta necesidad se vuelve indispensable debido a la complejidad logística que implica el control riguroso de insumos y el seguimiento histórico de los activos físicos. En este contexto, el presente capítulo desarrolla el Estado del Arte que fundamenta, tanto teórica como tecnológicamente, la propuesta de un sistema web de gestión integral y trazabilidad para la vulcanizadora Superllanta.

A continuación, se presenta un análisis que sistematiza la producción académica y los desarrollos tecnológicos publicados en el período comprendido entre 2021 y 2026 con el fin de garantizar la rigurosidad, calidad y pertinencia de la revisión. Además, la investigación documental se sustentó en bases de datos científicas de alto impacto, principalmente en artículos disponibles en IEEE Xplore y repositorios registrados en Scopus, dando prioridad a

publicaciones de revistas especializadas y actas de conferencias internacionales con su respectivo DOI verificable.

Con el objetivo de abordar de manera completa las necesidades del emprendimiento, la revisión se estructuró de forma sistemática en cuatro aspectos principales, que se encuentran vinculados de forma directa con la problemática del proyecto, y son:

Transformación digital de las PyMES

La transformación digital de las pequeñas y medianas empresas ha sido objeto de una creciente atención académica en estos últimos años. Omrani, Rejeb, Maalaoui, Dabić y Kraus han llevado a cabo un estudio de gran escala en el cual analizaron los factores que determinan la aceptación de tecnologías digitales en las pequeñas y medianas empresas, usando el marco de referencia tecnología-organización-entorno (TOE) en una muestra de 15 346 empresas de Europa y fuera de Europa [5]. Publicado en IEEE Transactions on Engineering Management (2022), dicho trabajo concluye que la infraestructura tecnológica existente y el nivel de innovación previsto son los principales factores habilitadores en la adopción de tecnologías digitales, mientras que la regulación corporativa, las habilidades disponibles y los recursos financieros también influyen significativamente en la decisión de adopción [5].

Estos hallazgos son aplicables al contexto de Superllanta, puesto que la vulcanizadora no tiene una infraestructura tecnológica de base y durante años ha operado en un entorno donde la digitalización del sector de servicios de los automotores es naciente, lo cual la ubica en el perfil de las empresas donde la iniciativa digital es mayor, pero su potencial de mejora operativa mediante un software es más significativo.

El estudio de Omrani [5] también evidencia que las pequeñas y medianas empresas que han logrado avanzar en su transformación digital lo hacen de manera incremental, puesto que comienzan por digitalizar los procesos más repetitivos y propensos a errores precisamente los que Superllanta tiene pendientes: registro de servicios, control de inventario y seguimiento de clientes. Esta representación valida el enfoque modular del sistema propuesto, que no pretende

transformar radicalmente la operación de un día para otro, sino introducir mejoras concretas y medibles en los procesos críticos del negocio.

Sistemas web para el control de inventario en organizaciones

En el ámbito de la gestión web de inventarios, Misahuaman, Daza y Zavaleta han llevado a cabo una revisión sistemática de la literatura sobre los sistemas web para el control de inventario, presentada en la 22.^a IEEE/ACIS International Conference on Software Engineering (SNPD 2021) [6]. Dicho trabajo analiza varios sistemas web desplegados en organizaciones de distinto tamaño e industria, y concluye que las aplicaciones web enfocadas en control de inventario generan un impacto cuantificable en la eficiencia de las operaciones, reducen los costos asociados al hardware propietario además de permitir adaptar la lógica de negocio a los requerimientos específicos de cada una de las empresas mediante módulos parametrizables. Además, los autores destacan que la arquitectura web facilita el acceso multiusuario en tiempo real, eliminando así la dependencia de registros en papel y los errores derivados de la duplicación manual de datos [6].

En una línea suplementaria, Tanaman et al. documentaron el diseño e implementación de un sistema web de gestión de inventarios, para una microempresa con cuatro sucursales, en la cual, la totalidad de sus registros era gestionada mediante hojas de papel, una situación similar a la de Superllanta [6]. Su trabajo, publicado en el International Journal of Science and Advanced Information Technology (IJSAIT, vol. 12, núm. 5, 2023), muestra que la migración a una plataforma web permitió al propietario gestionar el inventario con mejoras comprobadas en la exactitud de los datos y en la velocidad de consulta [6]. Los resultados también mostraron que el sistema habilitó el registro en tiempo real de la entrada y salida de productos, permitiendo acceder a una visualización interactiva de datos a través de un panel de control, y generó alertas automáticas cuando los niveles de stock alcanzaban umbrales mínimos [6]. Este antecedente es especialmente significativo para Superllanta, debido a que presenta

condiciones organizacionales similares: se trata de un negocio de servicios con un volumen moderado de productos y la falta total de digitalización previa.

Ambos estudios muestran que los sistemas web son adecuados para pequeñas empresas, ya que no requieren infraestructura de servidores propia, ayudan a reducir costos de licencia y permiten el despliegue de manera incremental. Este hallazgo fundamenta la elección de una arquitectura web para Superllanta frente a soluciones ERP de gran escala que, por su complejidad y costo, excederían las capacidades del negocio.

Trazabilidad digital mediante códigos QR

La implementación de códigos QR como un mecanismo de identificación única y recuperación de información histórica ha sido estudiada en múltiples contextos de gestión de activos. Suakanto et al. plantearon un sistema inteligente de gestión de activos basado en tecnología QR, el cual permite a los usuarios realizar reservas, consultar agendas y registrar la entrada y salida mediante el escaneo de un código único asignado a cada uno de los activos [7]. Esta investigación, presentada en la 2023 International Conference on Advanced Computer and Communication Engineering (ICACEH) de IEEE, manifiesta que la asignación de un código QR a cada entidad en su caso, salas de un edificio, y en el contexto de Superllanta, neumáticos habilita una trazabilidad completa del ciclo de vida sin necesidad de lectores especializados, dado que cualquier teléfono inteligente moderno puede decodificar un código QR [7]. Los autores reportan que la solución facilita significativamente la gestión relacionada con la programación y la utilización de los activos, demostrando la viabilidad técnica del enfoque para contextos operacionales de una moderada complejidad [7].

Desde la perspectiva del sector automotriz, la trazabilidad mediante código QR tiene sus orígenes en la industria japonesa de componentes vehiculares, donde Denso Wave la desarrolló en 1994 para el control de inventario y seguimiento de autopartes en líneas de producción. No obstante existe una limitada investigación sobre la implementación de estas tecnologías en talleres de servicio conocidos como vulcanizadoras dentro de mercados

emergentes, vacío que este trabajo propone abordar. La asignación de un código QR a cada neumático atendido en Superllanta permitirá generar una hoja de vida con el historial completo de las reparaciones realizadas. Esto facilita la gestión de garantías documentadas, el control del ciclo de vida útil de los neumáticos y la resolución de disputas con los clientes. Este enfoque se alinea con la tendencia global hacia la digitalización de la cadena de valor de servicios: en lugar de depender de recibos físicos escritos, los cuales en ocasiones se pierden, el código QR vincula al neumático con su registro en la base de datos del sistema, accesible en tiempo real desde cualquier dispositivo.

Sistemas integrados de gestión de servicios, alertas de stock y órdenes de trabajo

El control automatizado del nivel de inventario mediante niveles mínimos de stock y alertas, es una funcionalidad reconocida como crítica en la literatura sobre sistemas de gestión para negocios que ofrecen servicios. Misahuaman et al. identifican el monitoreo de stock en tiempo real como uno de los beneficios diferenciadores de los sistemas web frente a los registros que se realizan manualmente, donde destaca la capacidad de poder generar notificaciones automáticas cuando un producto alcanza el nivel de reabastecimiento [8]. Esta funcionalidad, aborda directamente el problema de desabastecimiento de insumos como parches, válvulas y otros materiales que son usados dentro de Superllanta, donde la carencia de herramientas en momentos críticos genera pérdidas económicas e insatisfacción de los clientes.

La digitalización de las órdenes de trabajo (OT) constituye otra área de intervención prioritaria. La gestión manual de OT en talleres que ofrecen servicios a múltiples vehículos de una misma flota, como las empresas de transporte que son clientes de Superllanta genera confusiones durante la facturación, y complica la asignación de servicios a cada unidad vehicular. Los sistemas revisados por Misahuaman et al. [8] proponen módulos de registro de servicios vinculados a la identidad del cliente y del activo que se intervino, lo cual elimina la ambigüedad entre registros similares y habilita la generación de reportes por cliente, vehículo o

fecha. Tanaman et al. [6] corroboran esta capacidad al demostrar que la digitalización del registro de pedidos en su sistema ayudó a mantener trazabilidad completa sobre las transacciones sin depender de archivos físicos.

En conjunto, la literatura revisada pone en evidencia una tendencia clara: las soluciones tecnológicas, más efectivas para los negocios de servicios de pequeña escala, combinan en una única plataforma web el control de inventario con alertas de bajo stock del inventario, gestión de órdenes de trabajo e historial de los servicios. Los resultados de Omrani et al. [5] declaran este enfoque integral, al demostrar que las PyMES logran mayor beneficio cuando la transformación digital incorpora varios procesos operativos de manera conjunta, en lugar de implementarlos de forma fragmentada y sin conexión entre módulos.

Vacíos reconocidos y posicionamiento del presente estudio

El análisis bibliográfico permite identificar los siguientes vacíos investigativos que justifican el alcance y la originalidad del sistema propuesto para Superllanta:

- Ausencia de investigaciones que integren la gestión de inventario con la trazabilidad QR por neumático en talleres especializados de transporte pesado en América Latina.
- Escasa documentación sobre sistemas web encaminados específicamente al servicio de vulcanizado y reencauche, en contraste con la abundante bibliografía sobre gestión de inventarios en el área de manufactura o retail.
- Falta de soluciones documentadas que integren, en una misma plataforma, la generación de órdenes de trabajo, el control de stock con alertas y la hoja de vida digital de los neumáticos que hayan recibido servicios de reparación.
- Hay limitaciones en el interés académico sobre las particularidades de las PyMES de los servicios automotrices ubicados en áreas de mucho tráfico vehicular pesado en Ecuador. Aunque la demanda de reencauches y vulcanizado es considerable, la digitalización en los talleres aún se encuentra en etapas iniciales.

Frente a estos vacíos, el presente proyecto se presenta como una propuesta que adapta las mejores prácticas documentadas en arquitectura web, control de inventario con alertas, identificación mediante códigos QR e historial de servicios, adaptadas al contexto específico de Superllanta, ubicado en Santo Domingo de los Colorados. Esto constituye un aporte dentro del negocio y a la comunidad académica interesada en realizar la transformación digital de talleres de pequeña escala en Ecuador.

Marco Teórico

El presente marco teórico fundamenta tecnológicamente el desarrollo del sistema de gestión de Superllanta. En él se abordan las categorías de análisis que estructuran la presente investigación: (1) arquitectura y aplicaciones web, (2) gestión de inventario en los sistemas de información, (3) trazabilidad digital por medio de códigos QR, (4) las tecnologías de desarrollo utilizadas en la implementación, donde se incluyen React, Node.js, PostgreSQL y Prisma ORM, (5) la autenticación de usuarios basada en tokens JWT, y (6) la metodología Kanban aplicada al desarrollo del software. Para cada una de estas categorías se adopta una representación teórica que conecta la literatura académica con cada una de las decisiones de diseño adoptadas en este proyecto.

Arquitectura web y el modelo cliente/servidor

La arquitectura cliente/servidor constituye el paradigma dominante en el desarrollo de las aplicaciones web modernas. Este modelo establece que el cliente, que usualmente es un navegador o una aplicación móvil, envía algunas peticiones al servidor, el cual las procesa y devuelve una respuesta, generalmente en formato JSON. Marascu, Constantinescu y Alexandrescu [3] validaron dicho enfoque en un contexto de plataformas web para la gestión, donde señalaron que la separación entre el frontend y el backend reduce el acoplamiento del sistema, facilita el mantenimiento independiente de cada capa y permite escalar cada componente según las necesidades de los usuarios.

Una de las variantes más extendidas de la arquitectura cliente–servidor en las aplicaciones web modernas, es la denominada arquitectura de separación entre el frontend y backend, en la cual el cliente consume servicios expuestos por una Application Programming Interface (API). Lin et al. [9] han documentado la migración de su sistema NCSLab hacia la arquitectura React/Backend separado, y reportaron mejoras esenciales en la mantenibilidad y el rendimiento frente al modelo monolítico que manejaban con anterioridad. Este antecedente respalda la decisión de optar por la misma separación dentro de Superllanta: el frontend con React junto con Vite se comunica con el backend en Node.js mediante una API RESTful, lo cual permite que ambas capas puedan evolucionar de manera independiente y el sistema pueda ser desplegado en infraestructura de bajo costo.

APIs RESTful

El estilo arquitectónico REST (de sus siglas en inglés: Representational State Transfer) establece un conjunto de restricciones como el modelo cliente-servidor, la arquitectura sin estado, interfaz uniforme y sistemas en capas. Aplicando estos principios en servicios web, se obtienen interfaces predecibles y, sobre todo, escalables [10]. En el contexto de una API RESTful, cada uno de los recursos posee una URI (Identificador Uniforme de Recursos) única, además las operaciones se expresan mediante verbos HTTP (GET, POST, PUT, PATCH y DELETE), y sus respuestas se codifican habitualmente en formato JSON. Esta convención simplifica la integración con diferentes tipos de clientes, como navegadores, PWA y aplicaciones móviles, sin la necesidad de editar el backend. En el sistema de Superllanta, la API RESTful expone varios endpoints para los módulos de clientes, neumáticos, vehículos, servicios técnicos, inventario, reportes y los usuarios, todos estos módulos se encuentran protegidos por medio de la autenticación JWT.

Sistemas de información para la gestión de inventario

La gestión de inventario comprende un conjunto de procesos orientados a tener el control de las existencias de materiales, productos y los insumos utilizados en el negocio, con

el objetivo de garantizar la disponibilidad de recursos en el momento y lugar adecuado, minimizando así los costos de almacenamiento y los riesgos de desabastecimiento [4]. En el contexto de las vulcanizadoras, el inventario comprende tanto los insumos utilizados en los servicios, como parches, válvulas y demás materiales de reparación, como los neumáticos destinados al proceso de reencauches disponibles para la venta.

Misahuaman, Daza y Zavaleta [8] han realizado la revisión sistemática de sistemas web que se encuentran orientados al control de inventario en negocios de distintos tamaños, y concluyeron que las aplicaciones web ofrecen ventajas concretas frente a los registros manuales que se realizan: acceso multiusuario en tiempo real, reducción de errores por duplicación de datos y la generación automática de alertas de reposición cuando el stock disminuye por debajo de un stock mínimo propuesto con anterioridad. De manera adicional, Tanaman et al. [6] documentan la implementación de un sistema web para la gestión de inventarios para una microempresa con las características similares a las de Superllanta: gestión del negocio totalmente manual, con un volumen moderado de productos y sin la infraestructura tecnológica adecuada, y reportaron mejoras conmensurables en la exactitud de los datos y la velocidad de consulta de los datos. Estos descubrimientos justifican el módulo de inventario de Superllanta, el cual implementa alertas automáticas de stock mínimo, registro de movimientos de entrada y salida, y los filtros de información ya sea por categoría o estado.

Muñoz Macas et al. [4] identificaron, a partir de la revisión de la bibliografía sobre la gestión de inventarios en las empresas de servicios, que los sistemas de información que integran el control de stock para el negocio junto con la gestión de órdenes de trabajo generan mayores beneficios operacionales que aquellos negocios que gestionan cada uno de los procesos de manera aislada. Esta perspectiva fundamentó el diseño integrado para el sistema Superllanta, donde el módulo de inventario se encuentra vinculado directamente con los módulos de los servicios técnicos, como los servicios de reparaciones, órdenes de trabajo y

vulcanizados, de tal modo que cada uno de los servicios registrados descuenta automáticamente el insumo que se haya utilizado del stock de inventario.

Trazabilidad digital mediante códigos QR

La trazabilidad es la capacidad de rastrear el ciclo de vida completo de un activo o producto, a través del registro sistemático de las intervenciones que recibe. Dentro del sector automotriz, la trazabilidad de los neumáticos permite documentar el historial de servicios de cada neumático junto con su debida fecha, tipo de intervención, el técnico responsable del arreglo, materiales empleados y con ello ofrecer garantías documentadas, cuando corresponda, detectar los neumáticos que han alcanzado el límite seguro de reparaciones y poder dar de baja los neumáticos que ya no sirven, y resolver disputas con evidencia objetiva.

El código QR fue desarrollado por Denso Wave en el año de 1994, para el seguimiento de componentes en líneas de producción automotriz. La capacidad para almacenar hasta 7 089 caracteres numéricos o 4 296 alfanuméricos, junto con la resistencia a errores gracias al algoritmo de corrección Reed-Solomon, lo convirtieron en un estándar global para la identificación de activos físicos. Muñoz et al. [1] nos mostraron que la transmisión de datos hacia dispositivos móviles mediante códigos QR es sumamente viable y confiable en los entornos de conectividad limitada, lo cual resulta relevante para el contexto de operación de Superllanta, donde el personal de taller accede al sistema desde celulares en entornos con cobertura variable.

Suakanto et al. [8] han propuesto un sistema de gestión de activos, basado en códigos QR para la administración de salas en edificios, donde concluyeron que la asignación de un código único por activo, análoga a la propuesta para neumáticos en el presente proyecto, habilita una trazabilidad completa del ciclo de vida sin necesidad de un hardware especializado, dado que cualquier smartphone con cámara puede decodificar el código. La hoja de vida digital del neumático implementada en Superllanta adopta dicho principio: cada neumático ingresado al taller al realizar un servicio de reparación o vulcanizado, recibe un código QR único con el

formato SL-año-identificador, generado así automáticamente por el backend haciendo uso de la librería qrcode de Node.js. Al escanear el código, el usuario puede acceder, desde cualquier dispositivo con cámara y navegador, al historial completo de servicios del neumático, sin requerir instalación de ninguna aplicación, y sin tener que estar registrado.

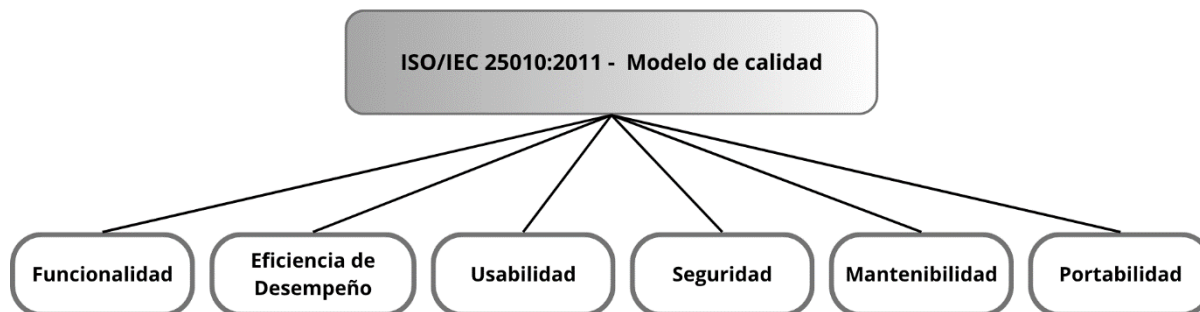
Atributos de calidad

Con el propósito de evaluar formalmente la calidad del sistema Superllanta, más allá del cumplimiento funcional, se adoptó el modelo de calidad del producto definido en la norma ISO/IEC 25010:2011, la cual reemplazó la norma ISO/IEC 9126, que es ampliamente usada para evaluar los sistemas de software de distinta escala, adoptando plataformas web ya en producción [11]. De las ocho características propuestas por este modelo, se seleccionaron seis, en función de la relevancia directa para que sea un sistema de gestión operativo, usado por un número reducido de usuarios en el entorno real de producción: la funcionalidad, eficiencia del desempeño, usabilidad, seguridad, mantenibilidad y portabilidad. Características como fiabilidad y compatibilidad no fueron seleccionadas debido a que exceden el alcance definido para el sistema, orientado a un solo negocio, y sin integración con sistemas externos de terceros.

La selección de estas características fue consistente con estudios recientes que aplican este modelo ISO/IEC 25010 a sus sistemas web de alcance acotado, donde se evalúa la calidad a través de dimensiones pertinentes al contexto de uso real del sistema, en lugar de todas las características en su totalidad como se muestra en Figura 1 [12].

Figura 1

Características modelo ISO/IEC 25010:2011 seleccionadas para la evaluación del software Superllanta.



Tecnologías de desarrollo web

React y el paradigma de componentes

React es una biblioteca de JavaScript de código abierto desarrollada por Meta y orientada a la construcción de interfaces de usuario mediante el paradigma de componentes que pueden ser reutilizables. Cada componente encapsula su propia lógica, estado y presentación, el árbol de componentes refleja la estructura escalonada de la interfaz. Cuando se producen cambios de estado, React actualiza solo las secciones del DOM (Document Object Model) que han cambiado mediante el mecanismo de Virtual DOM, lo cual reduce el costo computacional de los re-renders y mejora el rendimiento percibido por el usuario.

Keshari [13] ha documentado las principales ventajas de React en el desarrollo web tales como: es una arquitectura basada en componentes que favorece la reutilización de código, tiene soporte para server-side rendering, es compatible con React Native para el desarrollo móvil y una amplia comunidad la cual produce librerías especializadas. Dichas características positivas motivaron la elección de React como framework frontend de Superllanta. El sistema emplea React 18 junto con Vite como herramienta de construcción, usando Tailwind CSS para el diseño de la interfaz, Lucide React para el uso de iconografía y

TanStack React Query para la gestión del estado asíncrono y el caché de peticiones directamente a la API.

Node.js y Express como backend

Node.js es un entorno de ejecución de JavaScript, presente del lado del servidor construido sobre el motor V8 de Google Chrome. Esta arquitectura se encuentra orientada a eventos y su modelo de E/S no bloqueante (o también, non-blocking I/O) le permiten manejar una gran cantidad de conexiones concurrentes con un bajo consumo de recursos, característica especialmente valiosa en APIs que atienden múltiples clientes simultáneamente. Express es el framework web minimalista más popular sobre Node.js, el cual proporciona un sistema de enrutamiento, middleware usado para el procesamiento de peticiones y respuestas, y una API sencilla la cual facilita la exposición de endpoints RESTful.

El backend de Superllanta implementa una arquitectura modular de enrutamiento en Express, en donde cada recurso del dominio (clientes, neumáticos, inventario, usuarios, reparaciones, órdenes de trabajo, vulcanizados o reencauches) muestra sus endpoints en un router independiente. El middleware de autenticación JWT valida el token en cada petición antes de que llegue al controlador, y el middleware de autorización por roles restringe el acceso a operaciones sensibles, como la eliminación de registros o la consulta de reportes financieros, a los usuarios con rol de administrador.

PostgreSQL

PostgreSQL es un sistema gestor de base de datos relacional, de código abierto con más de 30 años de desarrollo activo. Este sistema soporta el estándar SQL completo, transacciones ACID (incluyendo propiedades como la atomicidad, asegurando que se completen las transacciones; la consistencia, donde garantiza la validez de los datos; el aislamiento, permitiendo la independencia de las transacciones; y la durabilidad, la cual garantiza que se mantengan los cambios en caso de fallos), integridad referencial mediante las claves foráneas, índices de varios tipos y extensiones avanzadas como búsqueda de texto

completo y tipos de datos JSON. Su robustez, rendimiento y soporte para concurrencia por el mecanismo MVCC (Multiversion Concurrency Control) la han posicionado como la base de datos relacional de código abierto más avanzada.

La base de datos de Superllanta está formada por 28 tablas organizadas alrededor de las entidades principales, como: clientes, vehículos, neumáticos e historial_neumatico (registra cada intervención sobre un neumático). Además, incluye elementos clave como reparaciones, vulcanizados, reencauches, órdenes de trabajo, productos (insumos) y seguimiento del inventario. Además, las claves foráneas garantizan la integridad referencial: por ejemplo, un servicio no puede existir sin el cliente al que está asociado, ni un movimiento de inventario sin el producto que modifica. PostgreSQL fue seleccionado por su confiabilidad y su capacidad para manejar consultas con múltiples JOINS, comunes en el módulo de reportes y en la hoja de vida del neumático, y su compatibilidad nativa con Prisma ORM.

Prisma ORM

Un ORM (Object-Relational Mapping) es una capa de abstracción, la cual establece una correspondencia entre las entidades de un modelo orientado a objetos, y las tablas de una base de datos relacional, lo cual permite al desarrollador manipular la base de datos mediante objetos del lenguaje de programación en lugar de sentencias directas SQL [14]. Un ORM reduce el código repetitivo asociado a las operaciones CRUD, agrupa la lógica de acceso a datos y mejora la seguridad mediante la parametrización automática de las consultas, lo cual previene inyecciones SQL.

Prisma es un ORM moderno para Node.js y TypeScript que introduce el concepto de Prisma Schema, un archivo declarativo donde se definen los modelos de datos, sus campos, tipos y relaciones. Basándose en este esquema, Prisma genera automáticamente un cliente tipado el cual ofrece métodos intuitivos para llevar a cabo las operaciones CRUD de todos los registros. Para Superllanta, este cliente en Prisma actúa como único punto de conexión entre PostgreSQL y el backend. Las migraciones del esquema son gestionadas a través del

comando “prisma migrate”, lo que asegura un seguimiento exitoso de los cambios en la estructura de la base de datos durante el ciclo de desarrollo.

Autenticación y autorización con JSON Web Tokens (JWT)

La autenticación es el proceso mediante el cual se verifica la identidad de un usuario, mientras que la autorización determina a qué recursos y operaciones puede acceder una vez autenticado. En aplicaciones web con arquitectura cliente-servidor sin estado (stateless), el mecanismo más extendido para poder implementar ambos procesos es el JSON Web Token (JWT), definido en la RFC 7519 del IETF.

Un JWT es una cadena codificada en Base64URL, compuesta por tres partes separadas por puntos, las cuales son: el encabezado, la carga y la firma (HMAC-SHA256 o RSA). Al autenticarse, el servidor genera un token y lo devuelve al cliente, donde en cada petición subsiguiente, el cliente incluye el token en el apartado Authorization: Bearer {token-generado}. El servidor se encarga de verificar la firma criptográfica sin necesidad de consultar a la base de datos, lo que es beneficioso, ya que elimina la sobrecarga de sesiones del lado del servidor y escala horizontalmente sin fricción.

Varalakshmi et al. [15] han demostrado la viabilidad y robustez de JWT como un mecanismo de autenticación en los entornos de aplicaciones web, señalando que la combinación de firma criptográfica junto con tiempo de expiración del token ofrece un equilibrio adecuado entre seguridad y rendimiento para aplicaciones de gestión empresarial. En la versión final de Superllanta, la autenticación se encuentra delegada a Auth0, que emite los tokens JWT utilizados por la aplicación. El backend actúa como servidor de recursos y valida la firma y los claims contenidos en el token antes de autorizar el acceso a cada endpoint. Los permisos se determinan según los roles administrador, técnico y cliente.

Metodología Kanban

Kanban es una metodología de gestión del flujo de trabajo, la cual tiene sus orígenes en los sistemas de producción Just-In-Time de la compañía Toyota, en la década de 1950 y fue

adaptada al desarrollo de software por David J. Anderson para el año 2007. El nombre de Kanban proviene de una palabra japonesa la cual significa “tarjeta” o “señal visual”. Dentro del contexto del software, Kanban se basa en cuatro principios, los cuales son: representar el trabajo a través del tablero que muestra los distintos estados del flujo, los cuales pueden ser “Por hacer”, “En progreso”, “En revisión” y “Realizado”; establecer los límites en el trabajo que se encuentra en progreso para evitar la sobrecarga y la multitarea excesiva; gestionar el flujo por la medición del tiempo de ciclo para cada tarea; y finalmente promover la mejora continua del proyecto.

Ahmad, Markkula y Oivo [16] realizaron una revisión sistemática de la literatura sobre Kanban en desarrollo de software y reportaron que sus principales beneficios son la mejora del tiempo de entrega, la reducción de defectos reportados por los clientes y el aumento de la consistencia en las entregas.

Furtado y Vinholi de Jesus [17] analizaron la aplicabilidad conjunta de Scrum y Kanban en equipos pequeños de desarrollo y concluyeron que la combinación de ambos enfoques, adoptando el tablero visual de Kanban con la priorización de backlog de Scrum, produce resultados superiores en contextos de mantenimiento concurrente de software en producción, situación análoga al trabajo de integración curricular donde el desarrollo de nuevas funcionalidades coexiste con la corrección de errores detectados en pruebas de campo.

El tablero Kanban de Superllanta se gestionó en Jira mediante el proyecto KAN. Primero se definieron las historias de usuario y posteriormente se generaron los tickets, organizados en las columnas ‘Por hacer’, ‘En progreso’ y ‘Hecho’. Esta organización permitió mantener visibilidad sobre el estado del proyecto en todo momento.

Durante el desarrollo del sistema se empleó la metodología Kanban mediante Jira para organizar el Product Backlog, registrar las historias de usuario y gestionar el flujo de trabajo. Las actividades se administraron en un tablero con estados ‘Por hacer’, ‘En progreso’, ‘Hecho’ y

‘Finalizado’, permitiendo controlar el avance del proyecto, priorizar tareas y dar seguimiento continuo a cada requerimiento hasta su implementación y validación.

Ciclo de vida del software

Con el fin de diferenciar la metodología de desarrollo de software de la metodología de investigación descrita al principio de este capítulo, en la Tabla 1 se detallan las etapas técnicas que fueron ejecutadas en el ciclo de desarrollo del sistema de Superllanta, cada una remite a la sección correspondiente donde se documenta cada detalle.

Tabla 1

Ciclo de desarrollo del software

Etapas	Descripción
Análisis de requerimientos	Levantamiento de requerimientos funcionales y no funcionales, con la observación directa y entrevista al propietario del negocio (en Identificación de requerimientos del sistema)
Diseño	Definición del modelo de base de datos relacional (un total de 28 tablas en PostgreSQL) y la arquitectura lógica y física del sistema (en Arquitectura del sistema)
Implementación	Desarrollo de cada historia de usuario, gestionado mediante el tablero Kanban en Jira (ver Planificación y Fases del desarrollo)
Pruebas	Verificación funcional y técnica del sistema mediante un plan de pruebas con casos de pruebas por cada módulo.

Despliegue	Publicación del sistema en el entorno de producción, accesible en el dominio superllanta.lat (ver Arquitectura física)
-------------------	--

Function Calling y Modelos de lenguaje

Los modelos de lenguaje de gran escala (LLM) son redes neuronales entrenadas con grandes volúmenes de texto que permiten interpretar y generar lenguaje natural con alta coherencia contextual. Aunque inicialmente su uso se centró en la generación de texto libre, en los últimos años se ha reconocido la técnica function calling o tool use, esta permite que el modelo identifique cuando la consulta requiere de datos externos y devuelva la instrucción estructurada para que sea ejecutada por el sistema que invoca la función sobre una fuente de datos reales y no solamente responda con texto generado con el conocimiento interno. Se fundamenta la respuesta con información verificable y no con contenido generado libremente por el modelo utilizado [18].

Maddigan y Susnjak al proponer Chat2VIS demuestran esta capacidad como un sistema que traduce consultas de lenguaje natural directo en código ejecutable para generar visualización de datos, evidenciando que los LLM pueden actuar como la capa intermedia confiable entre una fuente de datos estructurada y el usuario incluso si son consultas ambiguas o mal formuladas [18]. Este principio fundamenta el agente conversacional implementado en Superllanta y tiene un conjunto de funciones ya predefinidas de solo lectura (como consultar el stock, servicios pendientes, resumen de ventas, saldos pendientes de clientes, entre otras funciones) y no solo genera respuestas que se basan en el conocimiento general del modelo (esto es ineficiente, ya que el modelo puede generar cifras inventadas). Con esto el propio modelo es el encargado de decidir qué función debe invocar según la pregunta del usuario administrador, a continuación, el sistema ejecuta la función predefinida contra la base de datos

de PostgreSQL del negocio y devuelve información real al modelo, para que este redacte la respuesta final en español.

Control de acceso basado en roles (RBAC)

Este modelo organiza permisos de un sistema, en función de los roles asignados, en lugar de ser asignados directamente a usuarios individuales. Cada rol tiene un conjunto de permisos específicos, y los usuarios pueden tener uno o más roles, heredando automáticamente estos permisos. Este enfoque ayuda a la administración de un sistema dentro de organizaciones que contemplan varios usuarios con distintas funciones.

Superllanta implementa tres roles: administrador, técnico y cliente. El administrador tiene acceso completo al sistema y puede gestionar usuarios, consultar reportes financieros, eliminar registros y configurar el sistema. El técnico, en cambio, puede gestionar los trabajos asignados por el administrador, pero no tiene acceso a información financiera, no puede modificar los roles de los usuarios ni eliminar registros. Por su parte, el cliente puede consultar la información correspondiente a los trabajos realizados en Superllanta. Este diseño garantiza que el personal operativo de Superllanta tenga a disposición las herramientas necesarias para el trabajo, sin la necesidad de acceder a la información sensible del negocio como tal. Estos roles se codifican en el payload de JWT, además son validados en cada petición por el middleware de autorización en el backend, siguiendo el principio de defensa en profundidad: la restricción de acceso se aplica tanto en el frontend, ocultando opciones de interfaz, como en el backend, rechazando peticiones no autorizadas con código HTTP 403 Forbidden.

Autenticación delegada mediante OAuth 2.0 y OpenID Connect

Superllanta migró de un esquema propio de autenticación, contraseñas cifradas con bcrypt y tokens JWT, hacia el proveedor de identidad como Servicio, basado en estándares abiertos OAuth 2.0 el cual es un framework de autorización delegada que permite a una aplicación tener acceso limitado a recursos protegidos sin la necesidad de compartir credenciales directas con la aplicación, y OpenID Connect que añade sobre OAuth 2.0 una

capa de autenticación que permite verificar la identidad del usuario mediante un token firmado (ID Token) [19]. Dodanduwa y Kaluthanthri destacan que la robustez de este modelo depende de las relaciones de confianza preestablecidas entre los distintos roles del protocolo (propietario del recurso, cliente, servidor de autorización y servidor de recursos), relaciones que en Superllanta se materializan al delegar por completo la verificación de identidad a Auth0 como servidor de autorización, mientras el backend propio actúa únicamente como servidor de recursos [19].

Pasarelas de pago electrónico e integración server-to-server

Para el registro de pagos en línea de las órdenes de trabajo y del catálogo de neumáticos, se integró la pasarela de pagos PayPhone mediante su producto de Cajita de Pagos, un widget embebido en la propia interfaz del sistema que evita que los datos sensibles de la tarjeta transiten por el servidor propio, reduciendo el alcance de cumplimiento normativo de seguridad de datos de tarjetas (PCI DSS). La confirmación final del estado de cada transacción se realiza mediante una llamada directa servidor-a-servidor hacia la API de PayPhone, verificando tanto el estado de aprobación como el monto exacto de la transacción, en lugar de confiar únicamente en la redirección del navegador del cliente, que podría ser manipulada.

Marco Conceptual

El marco conceptual tiene como propósito delimitar y definir con precisión los conceptos centrales que estructuran la presente investigación, establecer sus características esenciales y exponer las relaciones que los vinculan entre sí [20]. Los conceptos aquí definidos corresponden a los elementos del dominio del negocio, como vulcanizadora, neumático, servicios técnicos e inventario, además de los componentes tecnológicos del sistema desarrollado.

Conceptos del dominio del negocio

Vulcanizadora

Una vulcanizadora es un negocio que es especializado en la reparación, vulcanizado y comercialización de neumáticos para vehículos de transporte ya sea liviano o pesado. El término vulcanizar proviene del proceso térmico del caucho con azufre, descubierto por Charles Goodyear en 1839. Este proceso transforma el caucho natural en un material más duradero y resistente al calor. En el contexto operativo de Superllanta, la vulcanizadora gestiona simultáneamente algunos flujos de trabajo, como la recepción de neumáticos, proceso para el diagnóstico del daño de los neumáticos, ejecución del servicio técnico, control de los insumos consumidos, facturación a los clientes y entrega. La digitalización de estos flujos mediante el sistema web constituye uno de los objetivos del presente proyecto.

Neumático

Un neumático, coloquialmente conocido como llanta en Latinoamérica, es el componente de caucho que recubre la llanta metálica de una rueda y proporciona tracción, absorción de impactos y estabilidad al vehículo. Está compuesto por algunas capas de alambre, la banda de rodamiento, los flancos laterales y talones de anclaje para el rin. Dentro del sistema, el neumático es la entidad central del modelo de datos, pues cada neumático que se ingresa al taller para realizar una reparación, recibe el identificador único (el código QR) y a medida que ese neumático recibe varias reparaciones acumula un historial de las intervenciones, lo cual constituye su hoja de vida. Estos neumáticos pueden ser clasificados en el taller según su tipo de registro, cuando ingresan por un servicio, para la venta o para ser procesados para reencauche.

Servicios técnicos

En una vulcanizadora los servicios técnicos comprenden un conjunto de operaciones que se realizan sobre un neumático para restaurar o mejorar la funcionalidad del neumático. El sistema Superllanta gestiona cuatro tipos de servicios, los cuales son:

- **Reparación:** aquí se realiza una intervención puntual que corrige un daño localizado, pinchazo, corte o válvula defectuosa, utilizando insumos como parches,

moñones, adhesivos o válvulas nuevas. La reparación descuenta los insumos utilizados del inventario y registra el costo del servicio.

- Cambio de neumático: un servicio de mano de obra, el cual consiste en desmontar un neumático del rin del vehículo, y montar otro. Este servicio no implica el uso de insumos, pero se registra la cantidad de cambios realizados y el precio por cada cambio de neumático.
- Vulcanizado: proceso de reparación aplicado a neumáticos que presentan daños de mayor magnitud, como cortes o perforaciones causadas por objetos de gran tamaño, cuyo arreglo se realiza aplicando un parche de caucho crudo y mediante la aplicación de calor y presión sellar el daño realizado.
- Reencauche: Este es un proceso de renovación de labrado o banda de rodamiento de un neumático ya desgastado pero que no haya sufrido daños con anterioridad. Aplica especialmente para neumáticos de camiones de carga pesada. El sistema permite gestionar varios neumáticos en una sola orden de reencauche y seguir el avance del proceso a través de las etapas.

Orden de trabajo

Una orden de trabajo (OT) es un documento el cual agrupa y formaliza una o más reparaciones o cambios de neumáticos que hayan sido solicitados por un cliente en una misma visita o para un solo vehículo. Constituye la unidad de facturación del taller: este contiene la identificación del cliente y el vehículo, la relación de servicios realizados con los respectivos costos y su total a cobrar. Dentro de Superllanta, la gestión manual de órdenes de trabajo para los clientes de flota anteriormente generaba algunas confusiones en la facturación o pérdida de los recibos, además el módulo de órdenes de trabajo del sistema resuelve este problema, ya que vincula cada servicio con el cliente, el vehículo, los neumáticos y los insumos que se hayan usado, manteniendo así un registro permanente y consultable.

Inventario e insumos

El inventario es el conjunto de los materiales, productos o insumos que dispone el taller para realizar los servicios y la comercialización. Los insumos son los materiales que se pueden usar directamente en cada uno de los servicios técnicos ya mencionados. Por ejemplo, parches de diferentes medidas, moñones, válvulas, adhesivos, los materiales para realizar el vulcanizado e incluso rines. El inventario también incluye los neumáticos que el negocio dispone para la venta. El sistema gestiona el stock de cada uno de los productos, permite registrar la entrada y salida, además de generar alertas automáticas cuando el nivel de existencias de un insumo desciende por debajo del stock mínimo que se haya configurado. El stock mínimo es el umbral de existencias a partir del cual se considera que el producto debe reponerse para evitar el desabastecimiento durante la prestación de servicios.

Cliente y flota de transporte

El cliente es la persona natural que solicita los servicios de la vulcanizadora. El sistema distingue dos tipos de clientes, donde tenemos al cliente individual, el cual es propietario de uno a cuatro vehículos, que son para uso personal o comercial, y el cliente flota que opera varios vehículos de transporte pesado y liviano. El cliente con flota es relevante en Superllanta, ya que genera órdenes de trabajo complejas, registrando varios neumáticos con diferentes unidades vehiculares, y su respectivo conductor. El sistema asocia cada vehículo con los neumáticos de cada uno, y asimismo con el propietario, así facilitando el seguimiento técnico y la facturación por vehículo y por flota.

Hoja de vida digital del neumático

La hoja de vida digital del neumático es el registro histórico completo de todas las intervenciones que haya recibido un neumático desde su ingreso al sistema. Aquí se incluye la fecha, el tipo de servicio, la descripción del daño o proceso aplicado y el técnico responsable de cada reparación realizada. Este concepto es clave para la debida innovación del proyecto y el negocio de vulcanizadora: al asignar un código QR único a cada neumático y vincularlo con su historial de reparaciones, cualquier usuario con acceso a un smartphone con cámara capaz

de leer códigos QR puede consultar en tiempo real la trayectoria técnica del neumático, sin necesidad de buscar recibos físicos o registros manuales. Dicha funcionalidad aborda directamente disputas relacionadas con garantías entre los clientes y el taller, además de permitir identificar neumáticos que ya hayan alcanzado un número crítico de reparaciones, lo que indica que se deben reemplazar por seguridad vial [8].

Capítulo III

Metodología

Marco metodológico investigativo

Enfoque de investigación

La presente investigación se desarrolló bajo un enfoque mixto, el cual combina los elementos del paradigma cualitativo y cuantitativo [21]. El componente cualitativo se refleja en el levantamiento de los siguientes requerimientos: la observación directa de los servicios y procesos manuales del taller, además la entrevista respectiva con el propietario del negocio y el personal técnico que labora en el taller, permitieron comprender la lógica del negocio e identificar los flujos de trabajo, las fuentes de error y las necesidades reales de digitalización. Por otro lado, el componente cuantitativo fue aplicado en la evaluación del sistema a través de métricas objetivas, como las historias de usuario completadas, número de módulos implementados y pruebas exitosas.

Los dos componentes se integraron de manera complementaria y secuencial, respondiendo así a los objetivos específicos de la investigación: el componente cualitativo estableció los requerimientos para lo cual fue necesario utilizar observaciones directas y entrevistas que permitieron identificar la realidad operativa del negocio. Así se definieron los requerimientos funcionales y no funcionales que fueron necesarios para el sistema; el componente cuantitativo, fue el encargado de verificar que los requerimientos se cumplan con todas las especificaciones definidas, esto mediante pruebas de software y cálculo de métricas de calidad. Esta integración secuencial respalda los objetivos descritos en el Capítulo I. Los dos

primeros objetivos fundamentaron el componente cualitativo: analizar los procesos del negocio para definir el modelo de datos y la arquitectura, apoyándose en el componente cualitativo; el tercero implementar los módulos funcionales, lo cual articula ambos componentes; y el cuarto, evaluar el funcionamiento del sistema mediante las pruebas de campo para determinar la usabilidad, se sustenta con el componente cuantitativo.

Tipo y diseño de investigación

De acuerdo con el propósito del estudio, la investigación que se aplicó es de tipo descriptivo y aplicado. Es descriptiva porque caracteriza los procesos operativos de Superllanta y las funcionalidades del sistema desarrollado, sin establecer relaciones causales ni manipular variables. Además, es aplicada o tecnológica en el ámbito de la ingeniería, porque su objetivo central es la resolución del problema concreto por medio del diseño e implementación de un sistema de software funcional, el cual opera en el entorno real del negocio [22].

El diseño es no experimental, debido a que los procesos aplicados en el taller fueron observados y analizados en el contexto natural sin la manipulación artificial de variables.

Población y muestra

El presente estudio se desarrolló en la vulcanizadora Superllanta, ubicada en Santo Domingo de los Colorados, Ecuador, y dedicada a la prestación de servicios automotrices. La unidad de análisis estuvo conformada por los usuarios directos del sistema de gestión implementado, es decir, las personas que interactúan con este en el entorno real del taller durante la fase de desarrollo.

La población estuvo conformada por los integrantes de Superllanta que utilizan directamente el sistema: el propietario del negocio, el cual desempeña el rol de administrador, y los trabajadores que desempeñan el rol de técnicos, y registran los servicios realizados mediante la aplicación.

Debido al tamaño de la población, se aplicó un muestreo censal, el cual es un tipo de muestreo que ocurre cuando la población es pequeña y accesible en cuanto a la relación

directa con el objeto de estudio, lo cual garantiza la visión integral y representativa de percepciones sobre el sistema implementado [21]. Dado que los usuarios de la organización a la que representan tienen contacto directo y regular con el sistema de gestión, no se han aplicado criterios de exclusión.

La elección de muestreo censal permite responder los objetivos específicos propuestos, como evaluar el funcionamiento del sistema por medio de pruebas de campo. Dado el enfoque aplicado de la investigación, estos resultados no tienen como finalidad la generalización estadística en otras vulcanizadoras, sino una validación del sistema en contexto específico para el cual fue diseñado, lo que constituye un alcance propio de una investigación de tipo aplicada y descriptiva en ingeniería [22].

Técnica e instrumento

La técnica de recolección de datos empleada fue la observación directa, aplicada durante la fase de levantamiento de requerimientos con el propósito de definir los requerimientos funcionales y no funcionales del sistema a partir de la operación real del taller. Mediante visitas al taller se registraron los procesos operativos de forma manual: registro de clientes, servicios ofrecidos, ausencia de control de inventario, neumáticos vendidos registrados en papel, falta de trazabilidad en los neumáticos reparados y desorganización de los insumos utilizados en los servicios de reparación. Esta técnica, explicada por Hernández Sampieri et al. [21], permitió identificar el flujo de trabajo real del taller y orientó el diseño de la solución propuesta. La validación del sistema una vez implementado se realizó mediante los mecanismos técnicos objetivos descritos en la sección “Pruebas” de este capítulo (análisis estático, análisis dinámico, evaluación de atributos de calidad ISO/IEC 25010, pruebas funcionales, de API y unitarias), en lugar de un instrumento de percepción aplicado a los usuarios.

Procedimiento de evaluación

La evaluación del sistema se desarrolló en cuatro etapas secuenciales, una vez concluida la implementación. En la primera etapa se ejecutó el análisis estático del código fuente mediante las herramientas Qodana Cloud y SonarCloud, obteniendo métricas objetivas de calidad interna del software. En la segunda etapa se realizó el análisis dinámico del sistema en producción, midiendo el tiempo de respuesta, rendimiento de la API, consumo de recursos del servidor y estabilidad durante su ejecución. En la tercera etapa se ejecutó el plan de pruebas técnico, compuesto por pruebas funcionales, de integración, de autenticación y autorización, de la API REST, de trazabilidad QR y pruebas unitarias sobre funciones críticas del backend. Finalmente, en la cuarta etapa se evaluaron los siete requisitos no funcionales definidos, relacionados con los atributos de calidad seleccionados de ISO/IEC 25010:2011 y con las restricciones operativas del sistema.

Consideraciones Éticas

El desarrollo de este proyecto fue llevado a cabo respetando los principios éticos que rigen la investigación aplicada en ingeniería, garantizando el respeto a los derechos, la dignidad y la participación voluntaria de las personas involucradas.

Tipo de participantes: los participantes de la investigación fueron los usuarios que operan el sistema de gestión Superllanta (propietario y técnicos), quienes participaron en la etapa de observación directa y en las pruebas funcionales del sistema durante su fase de uso real, sin presentar riesgos físicos o laborales. Consentimiento informado: previo a la observación directa y a las pruebas del sistema, cada participante fue informado sobre el objetivo del estudio, las actividades que se iban a realizar, el carácter voluntario de su participación y el tratamiento que se daría a la información recopilada.

Confidencialidad: toda la información recopilada durante la observación directa y las pruebas del sistema fue tratada de forma confidencial, sin vincular los datos registrados a la identidad individual de cada uno de los participantes. Estos datos fueron utilizados exclusivamente con fines académicos. Integridad de datos del negocio: puesto que el sistema

opera con información real del negocio (usuarios, clientes, inventario y servicios), las pruebas técnicas previas a esta evaluación de usuarios se realizaron con datos de prueba (con el sistema funcionando en local), minimizando todo riesgo de perder la información o cualquier acción que afecte a las operaciones reales de Superllanta.

Marco metodológico de desarrollo

El desarrollo del sistema de gestión Superllanta se llevó a cabo mediante la metodología ágil Kanban, la cual fue gestionada mediante el tablero digital en la plataforma Jira. Kanban es un método de gestión de flujo originado en sistemas de producción automotriz, y fue adaptado para el desarrollo de software en el año 2007; el nombre de la metodología deriva del japonés cuyo significado es “señal visual” o también “tarjeta”. La metodología se fundamenta en cuatro principios: (1) visualizar el trabajo en un tablero organizado en columnas que indican el estado del flujo de trabajo; (2) limitar el trabajo en progreso, para evitar la sobrecarga de trabajo; (3) gestionar el flujo con la medición del ciclo; y finalmente (4) mejorar continuamente de forma colaborativa [16].

A diferencia de la metodología Scrum, la cual organiza el trabajo mediante sprints de duración fija, la metodología Kanban permite la entrega continua e incremental de las funcionalidades a medida que se completan y validan, lo que permitió adaptarse dinámicamente a los requerimientos que surgieron durante las pruebas del sistema en el taller. Esta principal característica fue adecuada para el proyecto, puesto que los requerimientos de Superllanta se refinaron progresivamente a través de la interacción con el propietario y personal que labora en el negocio, lo cual generaba nuevas historias de usuario en función de las necesidades identificadas durante el uso real del sistema.

Análisis

Proceso de elicitación de requerimientos. Para la elicitación de requerimientos se desarrolló el proceso definido por Sommerville en Software Engineering [23], el cual establece que el levantamiento de requerimientos en sistemas de información para las organizaciones

debe combinar técnicas de descubrimiento, clasificación, priorización, validación y especificación de los requerimientos.

Este proceso, descrito por Sommerville, está compuesto de cuatro actividades: descubrimiento de requerimientos, la cooperación directa con la parte interesada e interacción con los usuarios; clasificación y organización, agrupados en funcionales y no funcionales; priorización y negociación con la parte interesada; y finalmente especificación y documentación. Para Superllanta, estas actividades fueron ejecutadas de manera secuencial en la fase de análisis, donde se aplicaron dos técnicas específicas.

Observación directa. Se realizó mediante visitas al taller durante los horarios de operación para poder registrar los procesos actuales sin intervenir en la rutina de trabajo. Esta técnica corresponde al método de observación etnográfica descrito por Sommerville (2016), y se aplica cuando los usuarios no pueden explicar verbalmente las tareas automatizadas o de forma automática debido a que son procesos complejos. Se documentaron: registro de servicios en notas de ventas físicas, control visual del inventario, ausencia de un registro sistemático de neumáticos disponibles o vendidos, y una gestión informal de los clientes.

Entrevista semiestructurada. Distribuida en cuatro bloques según Sommerville (2016), donde se identificaron: procesos actuales, problemas del proceso actual, funcionalidades deseadas y roles de cada usuario. Esta estructura permite analizar a fondo cada detalle del negocio, y con la entrevista se completó metódicamente cada aspecto necesario para la especificación completa de cada requerimiento. La cantidad de requerimientos se verificó junto con el propietario al término del análisis, confirmando que los requerimientos identificados representaran las necesidades reales del negocio.

Entrevista de elicitación al propietario. Como parte del proceso de descubrimiento de requerimientos explicado por Sommerville [23], se realizó una entrevista semiestructurada al propietario del negocio Superllanta, complementada con observación directa de los procesos operativos que se realizan en el negocio. Esta entrevista se organizó en dos bloques, detalles

del funcionamiento actual del negocio y expectativas para el sistema a desarrollar. A continuación, se muestran fragmentos más característicos de la entrevista, que ayudaron a sustentar la definición de los requerimientos funcionales del sistema como se muestra en la Tabla 2.

Tabla 2

Entrevista semiestructurada

Pregunta	Respuesta del propietario
¿Cómo se cuántos parches o materiales tengo en stock ahora mismo?	Simplemente viendo y contando la cantidad de producto que se dispone, y el problema es que a veces el producto no hay, y existe un gran problema.
¿Qué pasa cuando se acaba un insumo sin darme cuenta?	Esta es la parte más difícil del negocio, al no tener insumos, no se realiza el trabajo, para lo cual sería excelente contar con un sistema que me notifique cuando me quedo con poco producto, y este me recuerde que debo adquirir más.
¿Qué información necesito ver yo como dueño al final del día o mes?	Como dueño del negocio lo que yo necesito saber es la cantidad y precio de llantas que he vendido, la cantidad y precios de arreglos que se han realizado, y la cantidad de material que he usado.
¿Quiénes utilizarán el sistema dentro del negocio?	El propietario utilizaría el sistema para revisar reportes, controlar el inventario y consultar el historial de reparaciones, mientras que el personal operativo registraría el ingreso de neumáticos, las reparaciones realizadas y el consumo de insumos.

Pregunta	Respuesta del propietario
¿Cómo le entrego información al cliente sobre lo que le hice a su neumático?	Pues simplemente le digo el precio del parche, más bien no le doy una información muy detallada, sería bueno que dentro del sistema se pueda registrar la información de lo que hemos hecho en el neumático, y además la información del cliente, para que se le envíe un reporte, ya sea por correo o mejor aún por WhatsApp.

Estos fragmentos evidencian directamente el origen de las historias de usuario relacionadas con el control de inventario de insumos (H1), los reportes de gestión por período (H19 y H20) y las notificaciones al cliente (H28), detalladas en la sección Historias de usuario de este capítulo, así como el origen del requisito no funcional (RNF-4), el cual trata de seguridad y el control de acceso por roles, detallado en la Tabla 8.

Evidencia del proceso manual. Como evidencia complementaria de los problemas identificados durante la entrevista, se dispone de un registro fotográfico de los formatos físicos utilizados en el taller: hojas de ingreso de neumáticos, el registro de ventas de neumáticos y el registro de servicios en el negocio, este cuenta con insumos usados, o descripción de ciertos procesos. A continuación, se presentan fotografías de estos registros:

Figura 2

Registro manual para registrar el ingreso de neumáticos.

16 Ene 2028

NOMBRE: D.T.O. TELÉFONO: DIRECCIÓN:

N°	MARCA	MEDIDA	SERIE	DISEÑO SOLICITADO	VALOR
1	Conti-H	295-22.5	1777		
2	AEOLUS	295-22.5	4673		
3	Pichelin	295-22.5	2473		
4	Conti	295-22.5	7724		
5	Barot	295-22.5	7719		
6	Dunlop	295-22.5	5073		
7	AEOLUS	17R-22.1	7177		
8	Yokoh	215-22.5	7174		
9	Fullon	425-65-22.5	1418	40/Kon26	70
10					
11				SIN ORDEN	

Figura 3

Registro manual de ventas de neumáticos.

FACTURA

RUC.: 001-001
AUT. SRI.:

FECHA: 13-22-2025 RUC / C.I.:
CLIENTE: Fabian Delgado
DIRECCIÓN: TELÉFONO:

CANTIDAD	DESCRIPCIÓN	P.V.P	TOTAL
6	llantas Peencow		
	Techshield 315/60/225 (140,=		
	Double coin 12-22.5 (2619) 140,=		
	Longmarch 12-22.5 (4221) 140,=		
	Ovation 315/80/225 (6023) 140,=		
	Triangle 315/80/225 (1323) 120,=		
	Triangle 315/80/225 (3424) 120,=		
	Total 800,=		
	Abono 500,=		
	Saldo 300,=		

WST

FIRMA AUTORIZADA: [Firma] FIRMA CLIENTE: [Firma]

SUBTOTAL %: 800,=

SUBTOTAL 0%:

IVA:

VALOR TOTAL:

Figura 4

Registro manual de arreglos e insumos usados.

FACTURA		RUC.:	
		001 - 001	
		AUT. SRI.:	11
FECHA:	06 Feb. 2026	RUC / C.I.:	
CLIENTE:	HERRERA		
DIRECCIÓN:	JEN	TELÉFONO:	

CANTIDAD	DESCRIPCIÓN	P.V.P	TOTAL
1	LLANTA 700 R16		50 ⁰⁰
1	REBO Nuevo 700x16		15 ⁰⁰
1	DERESA NENA R16		12 ⁰⁰
2	ARREGLOS DE LLANTAS 700x16	4 ⁰⁰	8 ⁰⁰
	PCB-8021		
	ST. LENA MORENO		
		TOTAL	85 ⁰⁰

Estas evidencias confirman de manera objetiva las limitaciones descritas por el propietario, con respecto al control manual del inventario, trazabilidad de reparaciones en los neumáticos y el registro de ventas, lo que respalda la pertinencia de los RF definidos para el sistema Superllanta.

Identificación de requerimientos del sistema

Para la identificación de los requerimientos del sistema Superllanta se ha ejecutado en la fase de análisis del proyecto, efectuando dos técnicas: observación directa de los procesos operativos del negocio y una entrevista al propietario del negocio. Mediante la observación se pudo registrar ciertas limitaciones del proceso manual que realiza el negocio: no existe control de inventario, aún se registran servicios o llantas vendidas manualmente y falta de información para la toma de decisiones. La entrevista realizada completó el análisis, ayudando a que el dueño pueda comentar las necesidades específicas en cuanto a términos operativos y funcionalidades que se espera del sistema.

Con el proceso de elicitación de requerimientos se pudieron definir un total de 47 HU y siete requerimientos no funcionales (RNF). Cada requerimiento fue trazado desde la

problemática operativa, identificando hasta las funcionalidades implementadas, garantizando que el sistema responda a necesidades reales del negocio.

Historias de Usuario

Se presenta en detalle cada una de las 47 HU que se identificaron, las cuales se encuentran organizadas en 13 módulos funcionales, siguiendo una plantilla que complementa el registro simplificado del tablero Kanban y campos necesarios para justificar la priorización y esfuerzo de desarrollo.

La Tabla 3 presenta de manera detallada del proyecto la nomenclatura académica usada en este documento.

Tabla 3

Organización del proyecto

Elemento	Código/cantidad
Historia de Usuario	HU-01 a HU-47
Ticket Jira	KAN-12 en adelante
Módulos funcionales	13
Iteraciones	6
Estado final	47 finalizadas

A continuación, en la Tabla 4 se detalla cada uno de los campos de la plantilla que se usó para completar las historias de usuario.

Tabla 4

Definición de componentes en la plantilla usada para Historias de Usuario

Campo	Definición
Rol	Quien se beneficia de la acción.
Descripción	Que se necesita y para qué.

Campo	Definición
Criterio de aceptación	Cumplimento.
Prioridad	Representa la importancia de la HU, la cual no sustituye el módulo y la iteración.
Módulo funcional	Modulo al que pertenece la HU y organizar el Product Backlog

En la Tabla 5 tenemos las historias de usuario donde obtenemos la descripción, criterios de aceptación, y el módulo al que pertenece cada HU usadas en el proyecto

Superllanta

Tabla 5*Historias de Usuario*

ID / Jira	Rol	Descripción	Criterios de aceptación	Prioridad	Módulo funcional
HU-01 KAN-12	Administrador	Gestión de inventario de insumos. Registrar, actualizar y consultar los insumos del inventario para mantener un control actualizado de los materiales disponibles en el negocio.	Permite registrar insumos con nombre, categoría, stock y precio; editar insumos existentes; y visualizar el listado con el stock actual.	Alta	Inventario de insumos
HU-02 KAN-13	Administrador	Notificaciones de bajo stock. Recibir notificaciones cuando el stock de productos sea bajo para adquirir materiales antes de que se agoten.	Notificar al administrador cuando existe bajo stock en un insumo, mostrando el nombre y la cantidad actual del insumo.	Alta	Inventario de insumos
HU-03 KAN-14	Administrador	Consulta de stock para saber la cantidad de insumos que existe en el inventario en tiempo real.	Mostrar el nombre y cantidad disponible del insumo, además poder filtrar por categoría.	Alta	Inventario de insumos

ID / Jira	Rol	Descripción	Criterios de aceptación	Prioridad	Módulo funcional
HU-04 KAN-15	Administrador	Movimientos de inventario y consultar el historial de cada insumo, y detectar inconsistencia en el inventario.	Filtra movimientos, en un rango de fecha, mostrando la fecha, tipo y cantidad del insumo.	Media	Inventario de insumos
HU-05 KAN-16	Administrador	Registrar neumáticos para la venta y mantener el inventario actualizado con el stock real.	Poder registrar el DOT, marca, medida, precio de cada neumático, y marcarlo como disponible para la venta.	Alta	Neumáticos y ventas
HU-06 KAN-17	Administrador	Venta de neumáticos, y registrar ventas de uno o más neumáticos, y mantener el inventario de neumáticos actualizado en tiempo real.	Permitir seleccionar los neumáticos disponibles, y cambiar el estado si está vendido, además de actualizar el inventario.	Alta	Neumáticos y ventas
HU-07 KAN-18	Técnico	Generar y asignar código QR. Genera un código QR para identificar cada neumático, y realizar la trazabilidad.	Generar el QR por neumático, e imprimirlos, al escanearlo mostrar la hoja de vida del neumático.	Alta	Trazabilidad QR

ID / Jira	Rol	Descripción	Criterios de aceptación	Prioridad	Módulo funcional
HU-08	Usuario del sistema	Consulta la hoja de vida por código QR. Escanear el código para ver las reparaciones realizadas en el neumático.	Muestra datos del neumático; historial cronológico de intervenciones; insumos utilizados y técnico responsable.	Alta	Trazabilidad QR
KAN-19					
HU-09	Técnico	Registro de clientes. Registrar información de clientes para asociar correctamente servicios y ventas a cada persona o empresa.	Permite registrar datos del cliente; buscar por nombre o cédula; y utilizar el cliente registrado en servicios y ventas.	Media	Clientes y flota
KAN-20					
HU-10	Técnico	Registro de neumático para servicio de vulcanizado. Registrar el ingreso de un neumático para realizar servicio de vulcanizado, y asociar el servicio y neumático al cliente, para realizar la trazabilidad.	Registrar la información del cliente, abono y saldo pendiente por el servicio y generar el QR del neumático.	Alta	Servicio de taller
KAN-21					

ID / Jira	Rol	Descripción	Criterios de aceptación	Prioridad	Módulo funcional
HU-11 KAN-22	Técnico	Registro de reparación. Registrar las reparaciones realizadas al neumático y mantener la hoja de vida actualizada.	Poder seleccionar el neumático y registrar la reparación, seleccionando los insumos usados. Guardar la fecha y el técnico que realizó el servicio.	Alta	Servicio de taller
HU-12 KAN-23	Técnico	Orden de vulcanizado. Registrar la orden con la información del cliente y neumático, y saldos pendientes.	Registro de cliente y neumático, de existir saldos pendientes informar, y cambiar el estado de la orden.	Alta	Servicio de taller
HU-13 KAN-24	Usuario del sistema	Consultar estados de vulcanizados. Poder ver el estado de cada vulcanizado, para tener organizado los servicios del negocio.	Permite filtrar por estado, entre pendiente, listo y entregado, actualizando la lista respecto al estado.	Alta	Servicio de taller
HU-14 KAN-25	Técnico	Actualizar el estado, cambiando el estado conforme se realice el	Permite cambiar el estado, pendiente, listo y entregado,	Alta	Servicio de taller

ID / Jira	Rol	Descripción	Criterios de aceptación	Prioridad	Módulo funcional
		trabajo, y notificar al cliente en tiempo real.	registrando la fecha de entrega, ya actualizar el listado.		
HU-15 KAN-26	Técnico	Orden de reencauche. Registrar la orden con la información del cliente, y los neumáticos que se realizaran el servicio.	Permite registrar varios neumáticos por orden, junto con los datos del cliente, además de los abonos y saldos pendientes del cliente.	Alta	Servicio de taller
HU-16 KAN-27	Usuario del sistema	Consulta los reencauches por cliente. Buscar el cliente y consultar las órdenes de reencauches, y saber los neumáticos, valores y el estado de la orden.	Permite listar las órdenes por clientes, y consultar los detalles como precios, abonos, saldos y estados de las órdenes.	Media	Servicio de taller
HU-17 KAN-28	Administrador	Gestión de usuarios. Crear, editar y desactivar las cuentas de técnicos, y tener el control del sistema.	Permite editar el rol de los usuarios, y desactivar al usuario sin eliminarlo	Alta	Usuario y seguridad

ID / Jira	Rol	Descripción	Criterios de aceptación	Prioridad	Módulo funcional
HU-18	Usuario	Control de acceso por rol. Acceso	El administrador tiene acceso	Alta	Usuarios y seguridad
KAN-29	autenticado	único al módulo autorizado, según el rol. Para protección de la información del sistema.	total del sistema, y el técnico tiene módulos restringidos.		
HU-19	Administrador	Reporte de servicios por	Permite filtrar tipo y fechas,	Media	Reportes y analítica
KAN-30		periodos. Generar reportes de los servicios realizados por fecha, y conocer valores facturados.	mostrando la cantidad total y saldos pendientes por fechas.		
HU-20	Administrador	Reporte de insumos consumidos.	Agrupar consumos por producto y	Media	Reportes y analítica
KAN-31		Consultar el consumo de insumos por servicio y período para identificar materiales de mayor rotación.	servicio; permite filtrar por fechas; y muestra insumo, cantidad, unidad y servicio relacionado.		
HU-21	Técnico /	Listado de reparaciones y	Muestra la fecha, cliente,	Media	Servicio de taller
KAN-32	Administrador	cambios. Consultar reparaciones y cambios de neumáticos para	neumático, insumo y costo, y filtrar por tipo, fecha y técnico,		

ID / Jira	Rol	Descripción	Criterios de aceptación	Prioridad	Módulo funcional
		mantener control operativo y revisar el detalle de los servicios.	además de mostrar estadísticas diarias de servicios realizados.		
HU-22 KAN-33	Técnico / Administrador	Gestión de vehículos con sus choferes. Para clientes de flotas, permitir registrar el chofer de cada vehículo que pertenece a la flota. Y vincularlo al servicio de cada neumático con el vehículo.	Una flota tiene varios vehículos, cada vehículo se puede registrar la placa, el tipo y nombre del chofer. El vehículo se selecciona al registrar un servicio.	Media	Cliente y Flota
HU-23 KAN-34	Técnico / Administrador	Búsqueda inteligente de cliente y medida. Buscar la información deseada mediante autocompletado para localizar registros de manera rápida.	Filtrar resultados mostrando datos del cliente cuando se realiza la búsqueda y se selecciona el cliente.	Media	Experiencia de usuario
HU-24 KAN-35	Administrador	Corrección de inventario. Editar, eliminar y actualizar insumos y neumáticos registrados incorrectamente.	Permite eliminar y editar insumos en los inventarios, además de validar para que no existan movimientos relacionados.	Media	Inventario de insumos

ID / Jira	Rol	Descripción	Criterios de aceptación	Prioridad	Módulo funcional
HU-25 KAN-36	Administrador	Mejoras para ventas de neumáticos. Con métodos de pago, corrigiendo ventas y consultar el historial por cliente para tener control completo de las transacciones.	Registra método de pago; permite editar o eliminar ventas con confirmación; y muestra historial de ventas por cliente con detalle de cada neumático.	Media	Neumáticos y ventas
HU-26 KAN-37	Administrador	Gráficas de tendencias. Visualizar gráficas de servicios y ventas para identificar tendencias del negocio por período.	Muestra gráficas de servicios y ventas; permite filtrar por día, semana o mes; y diferencia visualmente tipos de servicio.	Media	Reportes y analítica
HU-27 KAN-38	Técnico / Administrador	Baja de neumático y alerta de cambio. Dar de baja neumáticos que ya no son aptos y generar una alerta de cambio para informar oportunamente su condición.	Permite registrar motivo de baja; muestra estado de baja en la hoja de vida; y genera alerta cuando supera el umbral de reparaciones.	Media	Trazabilidad QR

ID / Jira	Rol	Descripción	Criterios de aceptación	Prioridad	Módulo funcional
HU-28 KAN-39	Administrador	Notificación de servicio por WhatsApp. Notificar al cliente por WhatsApp cuando un servicio finaliza para informarle el trabajo realizado y los datos relevantes.	Incluye acción de notificación desde el servicio; genera mensaje con datos del trabajo; y para empresas puede incluir placa y chofer.	Media	Comunicación con clientes
HU-29 KAN-40	Administrador	Reporte de neumáticos vendidos. Consultar neumáticos vendidos por cliente y período para analizar ventas y el estado actual de cada neumático.	Incluir reporte de ventas, filtrar por fecha y cliente. Mostrando los detalles de cada neumático y los totales por periodo.	Media	Reportes y analítica
HU-30 KAN-41	Usuario del sistema	Aplicación móvil. Usar la aplicación desde cualquier dispositivo con acceso a la web.	La aplicación se adapta a las pantallas móviles.	Media	Experiencia de usuario
HU-31 KAN-42	Técnico	Orden de trabajos con varios servicios. Registrar varios arreglos y cambios en un mismo	Una orden tiene varios arreglos y cambios, a finalizar calcula el total, y muestra el detalle de los servicios realizados	Media	Orden de trabajo

ID / Jira	Rol	Descripción	Criterios de aceptación	Prioridad	Módulo funcional
		servicio, y unificar la cuenta de un cliente.			
HU-32 KAN-43	Técnico	Autocompletado en insumos y ciudades. Buscar insumos y ciudades en la ficha de órdenes, con autocompletados y agilizar los registros en las órdenes.	Filtrar insumos, y buscar ciudades, lo cual mantiene un buen comportamiento visual con los otros navegadores.	Media	Experiencia de usuario
HU-33 KAN-44	Técnico	Validación de formularios de servicios. Validar la información en formularios antes de agregar ítems, y evitar el registro con campos incompletos.	Impide agregar insumos al inventario si tiene ítems incompletos, aplica a los servicios.	Media	Órdenes de trabajo
HU-34 KAN-45	Técnico	Filtrar neumáticos por vehículo. Mostrar solo los neumáticos del vehículo seleccionado.	Seleccionar un cliente y filtrar los vehículos, si no existe el neumático, permite registrar un nuevo neumático.	Media	Cliente y flota

ID / Jira	Rol	Descripción	Criterios de aceptación	Prioridad	Módulo funcional
HU-35 KAN-46	Administrador	Notificar al cliente vía WhatsApp cuando su neumático ha sido dado de baja.	Esta acción solo tiene acceso el administrador, el cual tiene permitido notificar.	Media	Comunicación con cliente
HU-36 KAN-47	Administrador	Dashboard por rol. Mostrar al técnico un diferente panel al de administrador.	El técnico puede visualizar los servicios que esta realizando o ha realizados. Administrador tiene acceso a toda la información.	Media	Usuario y seguridad
HU-37 KAN-48	Administrador	Editar y eliminar servicios. Corregir o eliminar los registros de reparación, vulcanizados o reencauches ingresados de manera incorrecta.	Permitir editar los datos, y la eliminación con un mensaje de confirmación, esta acción solamente debe realizarla el administrador.	Media	Servicio del taller
HU-38 KAN-49	Técnico / Administrador	Mejorar las órdenes de trabajo. Una mejor gestión en órdenes, vinculación de los neumáticos existentes y notificaciones al cliente.	Tener una buena navegación, permitiendo seleccionar neumáticos del cliente o vehículo, además de notificar vía	Media	Orden de trabajo

ID / Jira	Rol	Descripción	Criterios de aceptación	Prioridad	Módulo funcional
			WhatsApp desde la visualización de la orden.		
HU-39 KAN-50	Técnico	Insumos y el QR en las órdenes de trabajo. Registrar los insumos usados y poder generar el código QR desde las órdenes de trabajo.	Permite seleccionar los insumos para los arreglos y automáticamente descontarlos del stock del inventario. Si el neumático no tiene código QR lo genera automáticamente.	Media	Orden de trabajo
HU-40 KAN-51	Administrador	Consulta administrativa con IA. Realizar preguntas sobre datos del negocio en lenguaje natural, y obtener una respuesta sin la necesidad de buscar o navegar en la aplicación web.	Con acceso al administrador, el agente de solo lectura puede responder con datos reales, manteniendo el historial de la conversación en la sesión del administrador.	Media	Inteligencia Artificial
HU-41 KAN-52	Técnico / Administrador	Registrar el número del negocio en WhatsApp y notificar desde el número oficial del negocio	Las notificaciones salen desde la sesión oficial del negocio; el envío se realiza desde el backend	Media	Comunicación con cliente

ID / Jira	Rol	Descripción	Criterios de aceptación	Prioridad	Módulo funcional
			autenticado; muestra estado de carga y confirmación; y está disponible para administradores y técnicos.		
HU-42 KAN-53	Administrador	Nota de venta digital por WhatsApp. Enviar al cliente una nota de venta informativa en PDF al finalizar una orden.	Genera PDF con detalle de insumos, mano de obra; permite obtener el documento desde la orden; lo envía por WhatsApp.	Media	Documento y comunicación
HU-43 KAN-54	Administrador	Consolidación para reparaciones en las órdenes de trabajo. Registrar arreglos y cambios solamente en las órdenes de trabajo, y evitar módulos redundantes y la duplicidad de datos.	Mantener datos para consultas, registrar los insumos asociados a ítems de las órdenes, además actualizar el dashboard, reportes y consultas y mantener la información histórica y nueva.	Media	Orden de trabajo

ID / Jira	Rol	Descripción	Criterios de aceptación	Prioridad	Módulo funcional
HU-44 KAN-56	Usuario del sistema	Navegación accesible para buscadores. Poder navegar los resultados en los buscadores con el teclado sin depender del mouse.	Usar el teclado para seleccionar la información deseada en los buscadores.	Media	Experiencia de usuario
HU-45 KAN-57	Administrador	Orden y paginación de listas. Ordenar los listados de clientes y reportes.	Permite ordenar de menor a mayor o en orden alfabético la información por columna.	Media	Reporte y analítica
HU-46 KAN-58	Técnico / Administrador	Sugerir marcas de neumáticos. Seleccionar marcas frecuentes y sugerir marcas ya precargadas, o registrar una nueva marca.	Sugerir marcas reconocidas, además de seleccionar una no registrada e incorporarla al catálogo.	Media	Neumáticos y ventas
HU-47 KAN-59	Visitante	Preguntas frecuentes en la página pública. Consultar preguntas frecuentes e información visual organizada en la página pública para resolver	La landing muestra preguntas frecuentes sobre temas del negocio; presenta servicios mediante componentes visuales	Media	Página pública

ID / Jira	Rol	Descripción	Criterios de aceptación	Prioridad	Módulo funcional
		dudas antes de contactar al negocio.	consistentes; y facilita la consulta de información previa al contacto.		

La Tabla 6 resume la distribución de las 47 HU del proyecto según su prioridad de negocio, determinada durante la elicitación y refinada durante la evolución del Product Backlog.

Tabla 6

Distribución de historias de usuario.

Prioridad	Cantidad de HU	Módulos representativos
Alta	15	Inventario de insumos, neumáticos y ventas, trazabilidad QR, servicios de taller, usuarios y seguridad.
Media	32	Clientes y flota, reportes y analítica, órdenes de trabajo, experiencia de usuario, comunicación con clientes, inteligencia artificial, documentos y página pública.
Baja	0	No se registraron historias con prioridad baja.
Total	47	13 módulos funcionales.

Planificación de Iteraciones. El tablero Kanban del proyecto se configuró en Jira con cuatro columnas que representan estos estados de flujo de desarrollo: (1) Por hacer: contiene todas las historias de usuario identificadas y priorizadas, pendientes para iniciar el desarrollo; (2) En progreso: se encuentran las historias en desarrollo o implementación y se encuentran activas en el momento de análisis; (3) En revisión: ingresa a un estado de prueba para validar que se cumpla lo que se solicitó en la historia; (4) Finalizado: las historias completadas, ya probadas en el entorno del taller y han sido validadas por el propietario del taller. El tablero del proyecto KAN comprende 6 iteraciones y 47 HU, y todas en estado finalizado en el cierre del proyecto. Con el propósito de mostrar qué historias se planificaron ejecutar en cada iteración y el criterio detrás de ese orden, se presenta la Tabla 7.

Tabla 7*Historias de usuario planificadas por iteración.*

Iteración	Periodo	Historias de usuario	Justificación
Iteración 1	09/04/2026- 22/04/2026	HU-01 a HU-21	Implementar el núcleo funcional del sistema: inventario, venta de neumáticos, trazabilidad QR, clientes, vulcanizados, reparaciones, reencauches, usuarios y reportes iniciales. Estas funcionalidades proporcionaron la base de datos y los procesos necesarios para la operación del taller.
Iteración 2	23/04/2026- 08/05/2026	HU-22 a HU-29, HU-33 y HU-35	A partir del uso del sistema incorporar funciones para gestión de vehículos, mantenimiento del inventario, mejoras de ventas, dashboard, baja de neumáticos, notificaciones y validaciones de formularios.
Iteración 3	11/05/2026- 16/05/2026	HU-31, HU-34 y HU-37	Fortalecer la operación de los servicios mediante órdenes de trabajo con múltiples servicios, filtrado de neumáticos por vehículo y mecanismos para corregir registros ingresados incorrectamente.
Iteración 4	17/05/2026- 22/06/2026	HU-30	Adaptar el sistema para el funcionamiento desde dispositivos móviles, teniendo un diseño responsive y con características de una aplicación web progresiva.

Iteración 5	23/06/2026	HU-32, HU-36,	Mejoras en cuanto a la experiencia del
	–	HU-38 y HU-39	usuario y el módulo de órdenes de trabajo
	27/07/2026		con características como búsquedas con
			autocompletado, diferenciación basada en
			roles en cuanto a la visualización, vinculación
			de llantas, notificaciones y el uso de los
			datos.
Iteración 6	25/07/2026-	HU-40 a HU-47	Incorporar mejoras finales surgidas durante
	31/07/2026		la evolución del sistema: consultas
			administrativas mediante IA, comunicación
			institucional mediante WhatsApp, notas de
			venta digitales, consolidación de órdenes,
			accesibilidad en buscadores, paginación de
			información y mejoras de la página pública.

Requisitos no funcionales

Los requerimientos no funcionales se definen atributos de calidad, los cuales el sistema debe cumplir independientemente de funcionalidades específicas. Se identificaron a partir de restricciones operativas dentro del negocio, perfil de usuarios y condiciones del entorno de uso, siguiendo las categorías del estándar ISO/IEC 25010:2011 los cuales se detallan en la Tabla 8.

Tabla 8

Requerimientos no funcionales de Superllanta.

ID	Categoría	Descripción	Problema identificado
RNF-01	Usabilidad	La interfaz debe ser fácil de usar, para usuarios	El personal del taller no cuenta con experiencia

ID	Categoría	Descripción	Problema identificado
		que no tengan formación técnica. Registra un servicio o producto no debe tomar más de cinco minutos al usuario.	para manejar sistemas informáticos.
RNF-02	Eficiencia de desempeño.	El sistema en cada consulta debe tener un tiempo de respuesta no mayor a cinco segundos	El sistema es usado en atención al cliente en tiempo real, por lo cual no debe haber tiempos de respuesta altos e interrumpir la operación.
RNF-03	Disponibilidad	El sistema debe estar disponible durante horario laboral (lunes a sábado de 07:00 a 19:00). Por lo cual la información debe estar almacenada de forma persistente.	El taller opera toda la semana en ocasiones con horario extendido, por lo que no debe depender de un servidor local que requiera mantenimiento
RNF-04	Seguridad	El sistema debe autenticar a los usuarios y aplicar control de acceso basado en roles, garantizando que	Datos financieros y operacionales del negocio, es información sensible. Los técnicos no pueden modificar precios

ID	Categoría	Descripción	Problema identificado
		administrador, técnico y cliente accedan únicamente a las funciones e información que les corresponden.	y consultar reportes de ingresos.
RNF-05	Portabilidad	El sistema debe funcionar en dispositivos con navegador web, sin una instalación adicional. Puede funcionar en dispositivos móviles.	Administradores y técnicos necesitan utilizar el sistema desde distintos dispositivos dentro y fuera del área de trabajo.
RNF-06	Escalabilidad	El sistema debe poder soportar el crecimiento del inventario, además de soportar nuevas funcionalidades sin degradar el rendimiento.	Superllanta crece, por lo que la solución debe ser válida a medida que se agreguen más clientes, neumáticos o se agreguen más tipo de servicios.
RNF-07	Mantenibilidad	El sistema debe mantener una estructura modular y separación entre presentación, lógica de negocio y acceso a datos,	El sistema requiere mantenimiento continuo y evolución funcional conforme aparezcan nuevas necesidades del negocio.

ID	Categoría	Descripción	Problema identificado
		permitiendo modificar o	
		incorporar	
		funcionalidades sin	
		afectar innecesariamente	
		otros componentes.	

Diseño

Arquitectura del sistema. Para el proyecto de gestión Superllanta se usó una arquitectura de software cliente/servidor, caracterizada por la separación entre el backend y frontend, manejando así también una estructura interna en capas. Dicha decisión arquitectónica permitió que ambas capas tuvieran una evolución independiente, lo cual ayuda al mantenimiento y permite el despliegue en una infraestructura de bajo costo, como se pudo implementar en este proyecto [3]. El sistema se encuentra organizado en dos dimensiones complementarias, las cuales son: arquitectura lógica, la cual describe la organización interna del software en sus capas funcionales, y arquitectura física, que detalla la infraestructura de despliegue a producción.

Arquitectura lógica. En Superllanta la arquitectura lógica se encuentra organizada en cuatro capas funcionales apiladas verticalmente, como se muestra en la Figura 5, para lo cual sigue el patrón de arquitectura en capas, documentado en la literatura de la ingeniería de software:

Capa de presentación. Esta capa constituye la interfaz con la cual el usuario final del sistema interactúa. Para el proyecto se implementó con React 18 como una biblioteca de interfaz de usuario, Vite como la herramienta de construcción y servidor de desarrollo, para el diseño visual se utilizó Tailwind CSS. Para la iconografía se utilizó la librería de Lucide React y TanStack React Query para la gestión de estado asíncrono y caché de las peticiones a la API,

y Recharts para la visualización de gráficas en el dashboard. La capa de presentación consume los datos exclusivamente mediante la API RESTful del backend, sin que esta capa tenga acceso directo a la base de datos.

Capa de comunicación. Actúa como un intermediario entre la capa de presentación y la capa lógica del negocio. Esta implementa un estilo arquitectónico REST (Representational State Transfer) por medio del protocolo HTTP con el intercambio de los datos con el formato JSON. En cuanto a seguridad, esta capa lo garantiza mediante JSON Web Tokens (JWT): al autenticarse un usuario, Auth0 emite los tokens JWT utilizados para la autenticación y el backend valida su firma y los claims necesarios antes de autorizar el acceso a los recursos. El control de acceso basado en roles distingue los permisos entre los usuarios administradores, técnicos y clientes, en el frontend (ocultando ciertas opciones de interfaz) y en el backend (rechazando peticiones no autorizadas para el código HTTP 403).

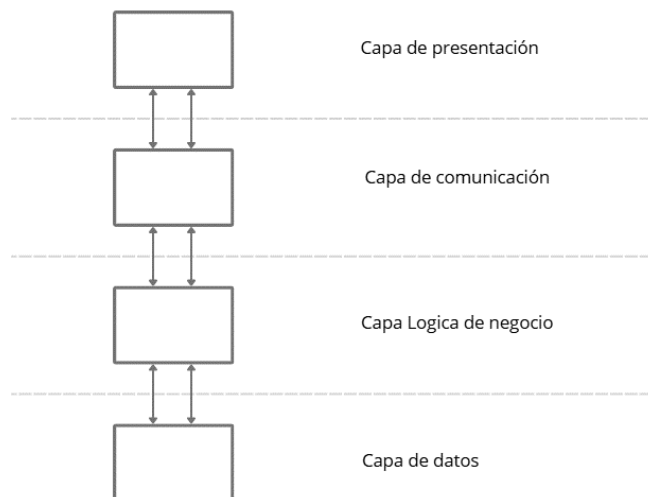
Capa lógica del negocio. Esta capa concentra todas las reglas que operan el sistema y fue implementada con Node.js como entorno de ejecución y como framework de enrutamiento se usó Express. Se describieron cinco módulos funcionales para el sistema, que son: Inventario de insumos, el cual gestiona los movimientos, el stock de productos y alertas para reponer los insumos; Servicios Técnicos, donde se registran los tipos de reparaciones que ofrece Superllanta, arreglos, vulcanizados, reencauches y la opción de órdenes de trabajo; Trazabilidad QR, donde se generan los códigos QR únicos y muestra una hoja de vida digital de ciertos neumáticos; Clientes y vehículos, usado para gestionar información del cliente, ya sea individual (dueño de un solo vehículo) o flotas de transporte (varios vehículos de un solo dueño); Reportes y usuarios, donde se genera el análisis por periodo y se administra el acceso para el sistema. La comunicación con la base de datos fue realizada por medio de Prisma ORM: el cual garantiza la parametrización automática de las consultas y contribuir a prevenir la inyección SQL.

Capa de datos. La capa utiliza PostgreSQL como sistema gestor de bases de datos y contiene 28 tablas que representan las entidades principales del negocio, como: clientes, vehículos, neumáticos, historial de los servicios, inventario, reportes y usuarios. Estas relaciones entre tablas se aseguran por medio de las claves foráneas, que garantizan la integridad referencial: un servicio no puede ser registrado si no existe un cliente válido, tampoco puede existir un movimiento en el inventario si no se agrega un producto en servicios de reparación u órdenes de trabajo. Las restricciones CHECK validan los valores de campos enumerados, que correspondan a valores permitidos por lógica del negocio.

Adicionalmente, el sistema integra servicios externos: la librería qrcode de Node.js, fue usada para generar automáticamente los códigos QR; para el tema de las impresiones en etiquetas con la impresora NIIMBOT B1 vía Web Bluetooth API la librería @mmote/niimbluelib, la cual permite imprimir directamente la etiqueta QR desde el navegador, sin la necesidad de agregar drivers a los dispositivos; por otro lado tenemos el servicio de inteligencia artificial, donde el backend se comunica con las peticiones HTTPS con la API de Groq, es un proveedor de inferencia, y expone el modelo de lenguaje openai/gpt-oss-120b y tiene soporte de function calling, el cual está empleado para responder preguntas sobre administración, con lenguaje natural contra la base de datos del negocio, este servicio es sumamente exclusivo para el rol de administrador; por otro lado, en el backend se integra WasenderApi, un proveedor externo y usa la API de WhatsApp, para el envío automatizado de mensajes, ya sean de texto o documentos PDF para el caso del negocio, y esto desde el número oficial del negocio, suplantando así el mecanismo basado en enlaces que redirigían hacia la aplicación de WhatsApp desde el dispositivo que se encontraba conectado el usuario.

Figura 5

Arquitectura lógica del sistema Superllanta.



Arquitectura física. Dentro de la arquitectura física se describe la infraestructura de despliegue que se usó en el sistema para enviar a producción, el sistema quedó accesible mediante el dominio superllanta.lat. El diseño actual prioriza un bajo costo operativo y disponibilidad durante el horario laboral en el negocio, para lo cual se aprovechan los servicios en la nube en niveles de bajo costo o planes gratuitos, como se muestra en la Figura 6 donde se muestra el diagrama de la arquitectura física de Superllanta.

Los dispositivos del cliente acceden al sistema mediante una computadora de escritorio o dispositivos móviles por medio de cualquier navegador moderno (Chrome, Brave o Microsoft Edge). El acceso desde el dispositivo móvil puede ser especialmente relevante, ya que el personal técnico del taller puede operar desde el celular para registrar los servicios que realicen si no hay comunicación con un administrador, y además pueden escanear el código QR de los neumáticos. Además, los dispositivos también usan la conexión Bluetooth para establecer conexión con la impresora NIIMBOT B1.

La capa de Cloudflare cumple con tres funciones en la arquitectura: DNS autoritativo para el dominio adquirido Superllanta.lat, resolviendo las peticiones según el recurso solicitado;

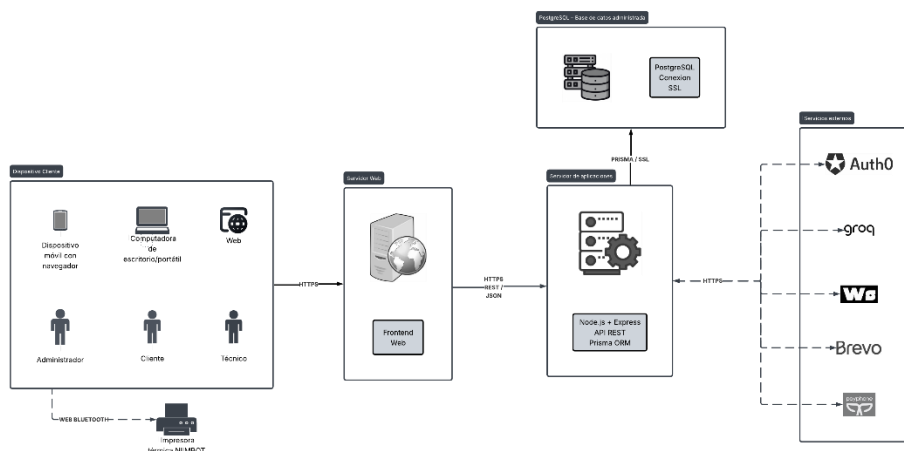
su función como CDN y caché, lo cual acelera la carga del frontend ya que almacena archivos estáticos en servidores distribuidos; uso de Cloudflare Workers, plataforma serverless, la cual permite el despliegue del frontend compilado (build de React y Vite), esto sin tener un costo adicional.

Render.com para el despliegue del backend, bajo un plan Starter (7 dólares estadounidenses por mes), el cual se ejecuta en el servidor Node.js + Express en el puerto 3000, con sus variables de entorno para credenciales sensibles. PostgreSQL como base de datos, también tiene su alojamiento en Render bajo el mismo costo (7 dólares estadounidenses por mes) y un pago adicional por aumentar el almacenamiento de 1 GB (0,30 dólares estadounidenses por mes), en el puerto 5432 con la conexión SSL activa. La comunicación entre el backend y la base de datos es interna, ya que las dos se encuentran dentro de la plataforma Render, lo cual reduce la latencia y elimina un costo adicional por transferencia de datos.

Las peticiones realizadas desde los dispositivos de los clientes siguen el siguiente flujo: el navegador realiza la solicitud HTTPS hacia el dominio Superllanta.lat; Cloudflare resuelve el DNS y entrega los archivos estáticos del frontend desde el Workers; React en el frontend, se ejecuta en el navegador del usuario, realiza la petición a la API RESTful del backend que se encuentra en Render mediante HTTPS; el backend se encarga de procesar la lógica del negocio, y consulta PostgreSQL por medio de Prisma ORM, y devuelve la respuesta en su formato JSON. Todo este tráfico se encuentra cifrado con TLS 1.3.

Figura 6

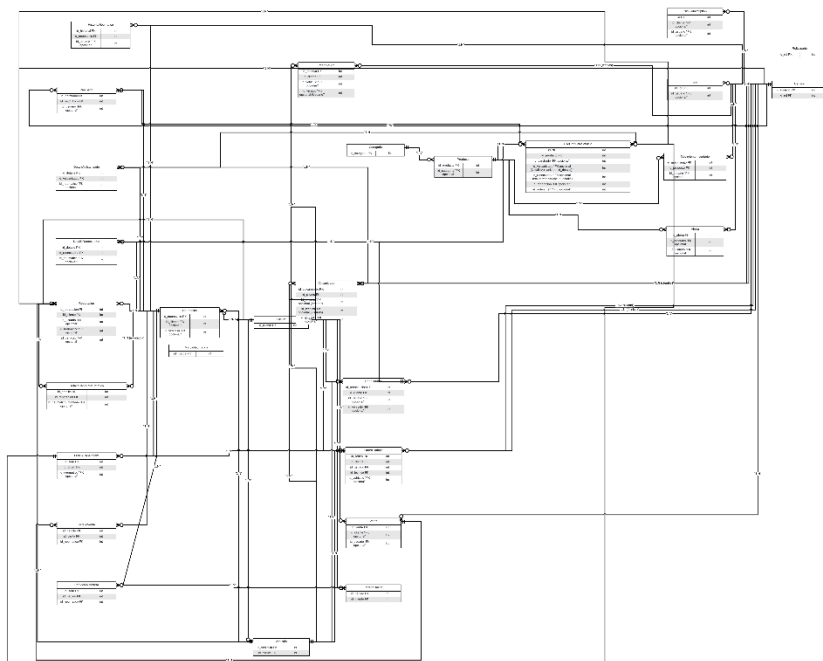
Arquitectura física del despliegue del sistema Superllanta.



Modelo de base de datos. El modelo de datos del sistema Superllanta fue diseñado de forma relacional sobre PostgreSQL, compuesto por 28 tablas que representan las entidades del dominio del negocio (insumos, neumáticos, clientes, vehículos, servicios técnicos, órdenes de trabajo, usuarios y reportes) y sus relaciones en la Figura 7. El diseño del esquema se gestionó mediante Prisma ORM, el cual permite versionar la evolución del modelo a través de migraciones controladas.

Figura 7

Modelo base de datos

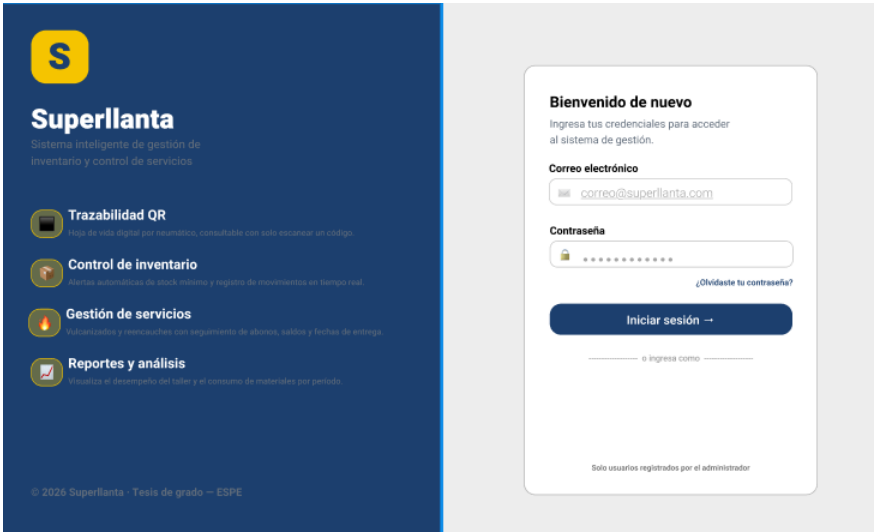


Prototipos. Para el diseño de las interfaces del sistema se emplearon prototipos de fidelidad media, elaborados con la herramienta Figma. A diferencia de un boceto de baja fidelidad, este tipo de prototipo incorpora mayor detalle visual e interactivo (paleta de colores, tipografía, iconografía y disposición definitiva de los elementos) y permite una navegación básica entre pantallas. Este nivel de fidelidad se consideró adecuado para el proyecto, ya que permitió validar con el propietario del negocio la propuesta visual y el flujo de uso de cada pantalla antes de iniciar la implementación, sin invertir el tiempo adicional que requeriría un prototipo de alta fidelidad, dado que el diseño definitivo se materializó directamente durante la implementación con React y Tailwind CSS.

La Figura 8 muestra a modo de ejemplo, el prototipo de fidelidad media correspondiente a la pantalla de inicio de sesión del sistema Superllanta.

Figura 8

Prototipo pantalla de inicio de sesión del sistema Superllanta



Implementación

Resultados por iteración. A continuación, se presenta, para cada una de las seis iteraciones de desarrollo, el conjunto de historias de usuario planificadas (Product Backlog de la iteración) y los resultados obtenidos al finalizar cada una.

Iteración 1- Implementación del módulo funcional. Al finalizar la primera iteración se encontraba disponible el núcleo funcional del sistema, incluyendo inventario, ventas, clientes, trazabilidad QR, servicios técnicos, usuarios y reportes iniciales. Las 21 historias fueron implementadas y trasladadas al estado Finalizado después de su verificación .

Tabla 9

Backlog de ejecución de la Iteración 1

Iteración	Inicio	Fin	Días hábiles	HU completadas	Puntos
1	09/04/2026	22/04/2026	10	21	63

Backlog	Jira	Historia de usuario	Prioridad	Estimación	Estado
HU-01	KAN-12	Gestión de inventario de insumos	Alta	3	Finalizado
HU-02	KAN-13	Notificaciones de bajo stock	Alta	3	Finalizado
HU-03	KAN-14	Consulta de stock	Alta	2	Finalizado
HU-04	KAN-15	Movimientos de inventario	Media	3	Finalizado
HU-05	KAN-16	Ingreso de neumáticos para venta	Alta	3	Finalizado
HU-06	KAN-17	Venta de neumáticos	Alta	3	Finalizado
HU-07	KAN-18	Generación y asignación de código QR	Alta	4	Finalizado
HU-08	KAN-19	Consulta de hoja de vida mediante QR	Alta	3	Finalizado
HU-09	KAN-20	Registro de clientes	Media	2	Finalizado

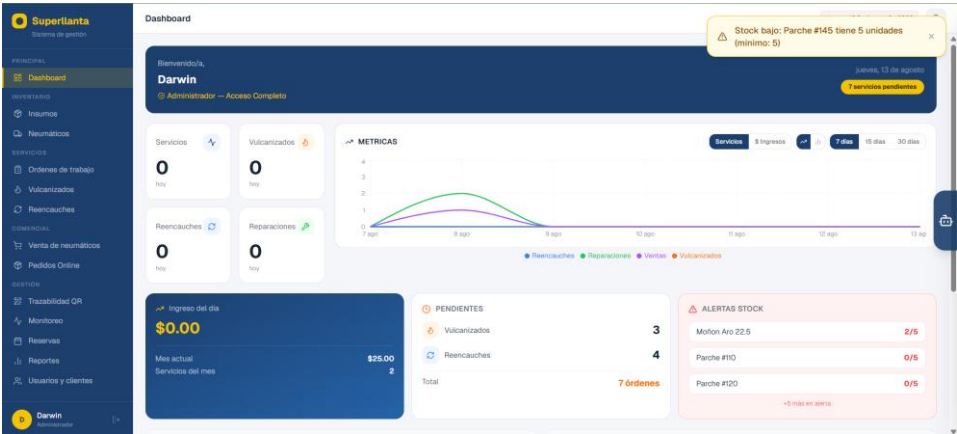
Backlog	Jira	Historia de usuario	Prioridad	Estimación	Estado
HU-10	KAN-21	Ingreso de neumático para vulcanizado	Alta	4	Finalizado
HU-11	KAN-22	Registro de reparación de neumático	Alta	3	Finalizado
HU-12	KAN-23	Orden de vulcanizado	Alta	4	Finalizado
HU-13	KAN-24	Consulta de vulcanizados	Alta	3	Finalizado
HU-14	KAN-25	Actualización de estado de vulcanizado	Alta	2	Finalizado
HU-15	KAN-26	Orden de reencauche	Alta	4	Finalizado
HU-16	KAN-27	Consulta de reencauches por cliente	Media	3	Finalizado
HU-17	KAN-28	Gestión de usuarios	Alta	3	Finalizado
HU-18	KAN-29	Control de acceso por rol	Alta	3	Finalizado

Backlog	Jira	Historia de usuario	Prioridad	Estimación	Estado
HU-19	KAN-30	Reporte de servicios por período	Media	3	Finalizado
HU-20	KAN-31	Reporte de insumos consumidos	Media	3	Finalizado
HU-21	KAN-32	Listado de reparaciones y cambios	Media	2	Finalizado

La Figura 9 muestra el dashboard administrativo implementado durante la primera iteración, en el cual se integran indicadores de servicios, métricas operativas y alertas de inventario. Esta evidencia permite observar el funcionamiento del núcleo inicial del sistema.

Figura 9

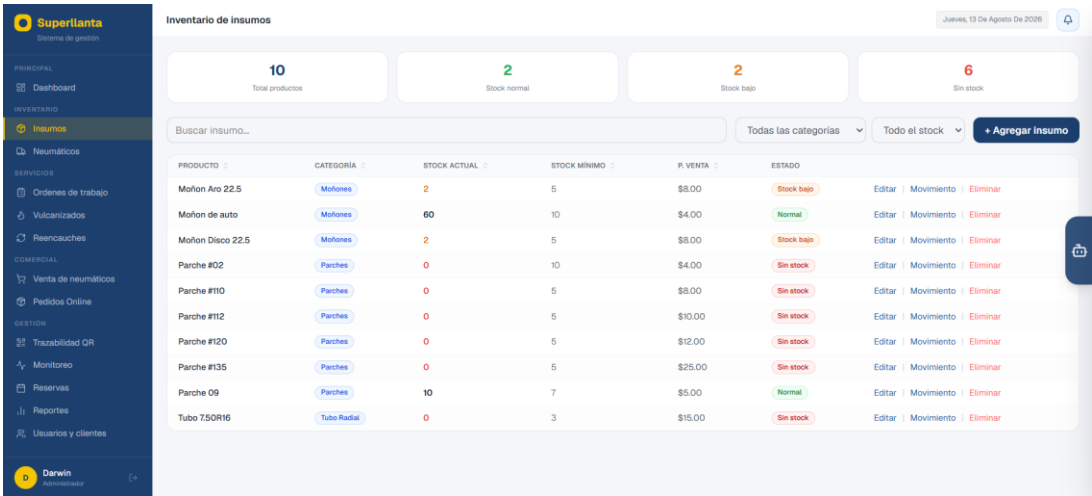
Resultado de la Iteración 1: núcleo funcional del sistema Superllanta



La Figura 10 presenta el listado de inventario de insumos con cantidades disponibles, stock mínimo, precios y estados de disponibilidad, como parte de las funcionalidades base implementadas durante la primera iteración.

Figura 10

Gestión del inventario de insumos implementada en la Iteración 1



Iteración 2 - Refinamiento funcional y operación real. Durante la segunda iteración se incorporaron funcionalidades derivadas del uso inicial del sistema: gestión de vehículos, mantenimiento de inventario, mejoras en ventas, tendencias, baja de neumáticos, notificaciones y validaciones. Las 10 historias quedaron finalizadas.

Tabla 10

Backlog de ejecución de la Iteración 2

Iteración	Inicio	Fin	Días hábiles	HU completadas	Puntos
2	23/04/2026	08/05/2026	11	10	28

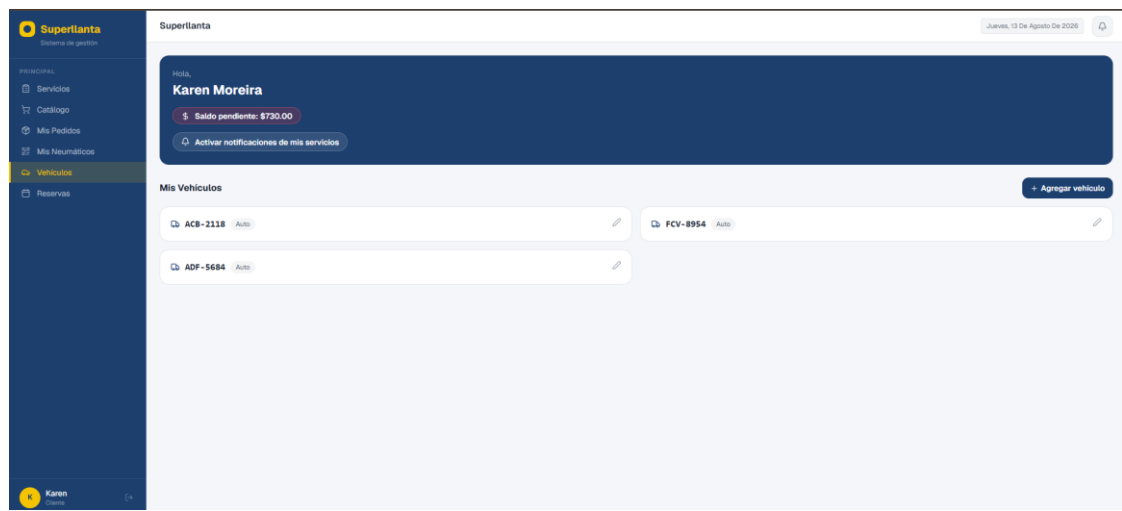
Backlog	Jira	Historia de usuario	Prioridad	Estimación	Estado
HU-22	KAN-33	Gestión de vehículos y choferes	Media	3	Finalizado

Backlog	Jira	Historia de usuario	Prioridad	Estimación	Estado
HU-23	KAN-34	Búsqueda inteligente de clientes y medidas	Media	2	Finalizado
HU-24	KAN-35	Corrección y mantenimiento de inventario	Media	5	Finalizado
HU-25	KAN-36	Mejoras en ventas de neumáticos	Media	3	Finalizado
HU-26	KAN-37	Gráficas de tendencias	Media	3	Finalizado
HU-27	KAN-38	Baja de neumático y alerta de cambio	Media	3	Finalizado
HU-28	KAN-39	Notificación de servicio por WhatsApp	Media	3	Finalizado
HU-29	KAN-40	Reporte de neumáticos vendidos	Media	3	Finalizado
HU-33	KAN-44	Validación de formularios de servicios	Media	2	Finalizado
HU-35	KAN-46	Notificación al dar de baja un neumático	Media	1	Finalizado

La Figura 11 evidencia la gestión de vehículos asociados a un cliente, permitiendo mantener la relación entre el propietario y las unidades vehiculares que posteriormente intervienen en los servicios del taller.

Figura 11

Resultado de la Iteración 2: gestión de clientes y vehículos



Iteración 3 - Consolidación de órdenes y gestión de servicios. La tercera iteración fortaleció el flujo operativo de los servicios mediante órdenes con múltiples trabajos, filtrado de neumáticos por vehículo y mecanismos de corrección de registros. Las 3 historias fueron finalizadas.

Tabla 11

Backlog de ejecución de la Iteración 3

Iteración	Inicio	Fin	Días hábiles	HU completadas	Puntos
3	11/05/2026	16/05/2026	6	3	14

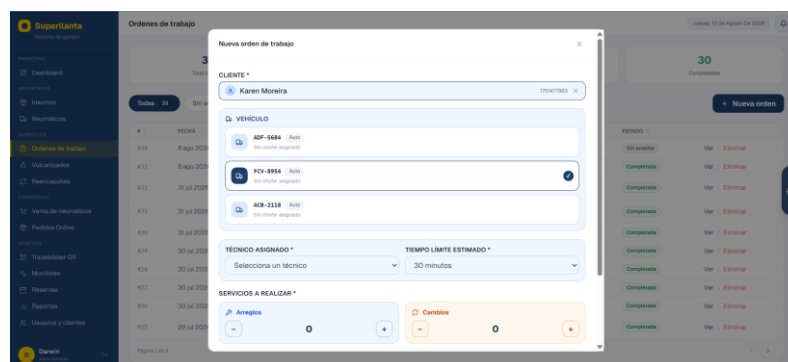
Backlog	Jira	Historia de usuario	Prioridad	Estimación	Estado
HU-31	KAN-42	Orden de trabajo con múltiples servicios	Media	8	Finalizado
HU-34	KAN-45	Filtrado de neumáticos por vehículo	Media	3	Finalizado

HU-37	KAN-48	Edición y eliminación de servicios	Media	3	Finalizado
-------	--------	------------------------------------	-------	---	------------

La Figura 12 muestra la creación de una orden de trabajo vinculada a un cliente y vehículo, con selección de neumáticos y configuración de los servicios incluidos. La evidencia representa la consolidación del flujo operativo de órdenes.

Figura 12

Resultado de la Iteración 3: orden de trabajo con múltiples servicios



Iteración 4 - Adaptación para dispositivos móviles. La cuarta iteración se concentró en la adaptación de la aplicación para teléfonos y tabletas mediante diseño responsive y capacidades PWA. La historia planificada fue completada y finalizada.

Tabla 12

Backlog de ejecución de la Iteración 4

Iteración	Inicio	Fin	Días hábiles	HU completadas	Puntos
4	17/05/2026	22/06/2026	25	1	8

Backlog	Jira	Historia de usuario	Prioridad	Estimación	Estado
HU-30	KAN-41	Uso móvil de la aplicación	Media	8	Finalizado

La Figura 13 muestra la adaptación del módulo de órdenes de trabajo a una resolución móvil, manteniendo indicadores, filtros, acciones y registros visibles en una pantalla de smartphone.

Figura 13

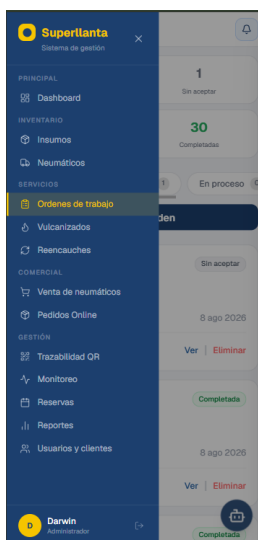
Resultado de la Iteración 4: vista responsive del módulo de órdenes de trabajo



La Figura 14 evidencia la adaptación del menú lateral del sistema a una pantalla móvil, permitiendo acceder a los distintos módulos desde un dispositivo de dimensiones reducidas.

Figura 14

Navegación responsive del sistema en dispositivo móvil



La Figura 15 presenta el portal del cliente adaptado a una resolución móvil, mostrando información del usuario y los vehículos asociados sin perder la funcionalidad principal de la interfaz.

Figura 15

Vista responsive del portal del cliente en dispositivo móvil



Iteración 5 - Refinamiento de experiencia y órdenes de trabajo. La quinta iteración refinó la experiencia de uso y el módulo de órdenes de trabajo mediante autocompletado, dashboard por rol, vinculación de neumáticos, notificaciones, consumo de insumos y generación de QR. Las 4 historias fueron finalizadas.

Tabla 13

Backlog de ejecución de la Iteración 5

Iteración	Inicio	Fin	Días hábiles	HU completadas	Puntos
5	23/06/2026	27/07/2026	25	4	12

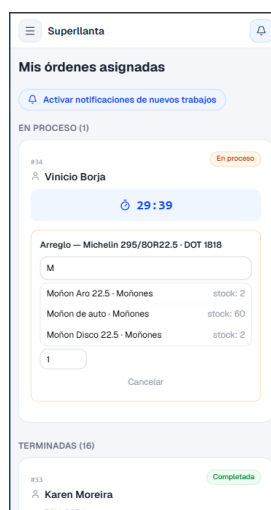
Backlog	Jira	Historia de usuario	Prioridad	Estimación	Estado
HU-32	KAN-43	Autocompletado de insumos y ciudades	Media	2	Finalizado

Backlog	Jira	Historia de usuario	Prioridad	Estimación	Estado
HU-36	KAN-47	Dashboard diferenciado por rol	Media	2	Finalizado
HU-38	KAN-49	Mejoras en órdenes de trabajo	Media	3	Finalizado
HU-39	KAN-50	Insumos y QR en órdenes de trabajo	Media	5	Finalizado

La Figura 16 muestra la vista móvil de órdenes asignadas al técnico, incluyendo el estado de la orden, control temporal y detalle de los neumáticos o actividades asociadas, evidenciando el refinamiento del flujo operativo.

Figura 16

Resultado de la Iteración 5: refinamiento de órdenes asignadas al técnico



Iteración 6 - Mejoras finales y evolución del sistema. La sexta iteración incorporó mejoras finales de evolución del sistema: consultas con inteligencia artificial, mensajería institucional, documentos digitales, consolidación de órdenes, accesibilidad, paginación, sugerencias de marcas y preguntas frecuentes. Las 8 historias fueron finalizadas.

Tabla 14*Backlog de ejecución de la Iteración 6*

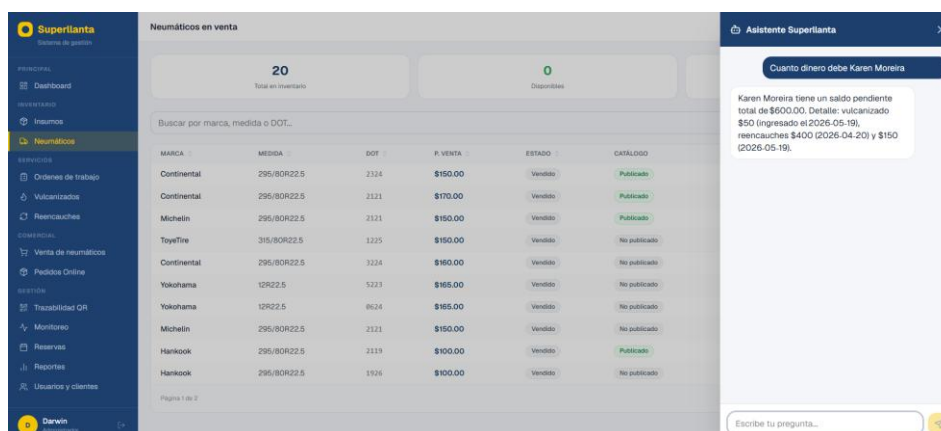
Iteración	Inicio	Fin	Días hábiles	HU completadas	Puntos
6	25/07/2026	31/07/2026	5	8	40

Backlog	Jira	Historia de usuario	Prioridad	Estimación	Estado
HU-40	KAN-51	Consultas administrativas con inteligencia artificial	Media	8	Finalizado
HU-41	KAN-52	WhatsApp desde el número oficial	Media	5	Finalizado
HU-42	KAN-53	Nota de venta digital por WhatsApp	Media	5	Finalizado
HU-43	KAN-54	Consolidación de reparaciones en órdenes de trabajo	Media	8	Finalizado
HU-44	KAN-56	Navegación accesible en buscadores	Media	3	Finalizado
HU-45	KAN-57	Ordenamiento y paginación de listados	Media	5	Finalizado
HU-46	KAN-58	Sugerencias de marcas de neumáticos	Media	3	Finalizado
HU-47	KAN-59	Preguntas frecuentes en la página pública	Media	3	Finalizado

La Figura 17 muestra el asistente administrativo integrado en el sistema respondiendo una consulta en lenguaje natural a partir de la información disponible en la aplicación, como parte de las mejoras finales incorporadas.

Figura 17

Resultado de la Iteración 6: consultas administrativas mediante inteligencia artificial



Fases del desarrollo

El ciclo de desarrollo para el sistema Superlanta fue estructurado en tres distintas fases, coherentes con el enfoque de entrega continua de Kanban, y son:

Fase 1 - Análisis, diseño e implementación inicial. A diferencia de un enfoque en cascada, en Kanban el análisis y el diseño no se desarrollan necesariamente como etapas completamente aisladas antes de iniciar la implementación. En esta fase se recolectaron los requerimientos a través de la observación y entrevistas con el propietario del negocio y luego se crearon el modelo y la arquitectura de la base de datos. El modelo de la base de datos consta de 28 tablas en PostgreSQL y la arquitectura del sistema. Además, se crearon e implementaron las primeras 20 historias de usuario (HU-01 a HU-20 para KAN-12 a KAN-31) principalmente relacionadas con el inventario, las ventas de llantas, la trazabilidad a través de códigos QR, la vulcanización, el reacondicionamiento, los usuarios, los clientes y los informes.

Fase 2 - Desarrollo Incremental y Refinamiento. En esta fase se crearon e implementaron las historias de usuario HU-21 a HU-39 (para KAN-32 a KAN-50)

progresivamente incorporándolas al sistema y probándolas antes de implementar nuevas funcionalidades. En esta fase se corrigieron algunos errores encontrados en el uso del sistema y se implementaron algunas mejoras en los pedidos de trabajo, la gestión de vehículos, el diseño responsivo, UX/UI y las operaciones. Estas funcionalidades correspondieron a las iteraciones intermedias del proyecto y permitieron adaptar el sistema a necesidades identificadas durante su utilización en el negocio.

Fase 3 - Mejoras finales y evolución del sistema. La fase final comprendió las historias HU-40 a HU-47, correspondientes a la sexta iteración del proyecto. Estas historias surgieron como parte del proceso de mejora continua y permitieron incorporar funcionalidades complementarias como consultas administrativas mediante inteligencia artificial, comunicación mediante WhatsApp desde el número oficial del negocio, generación y envío de notas de venta digitales, consolidación de reparaciones dentro de las órdenes de trabajo, mejoras de accesibilidad, paginación y ordenamiento de información, sugerencias de marcas de neumáticos y mejoras en la página pública.

Al finalizar las tres fases, las 47 historias de usuario que conformaron el Product Backlog final fueron implementadas, integradas y trasladadas al estado Finalizado en Jira. De esta manera, el desarrollo evolucionó desde la construcción del núcleo funcional del sistema hasta la incorporación de mejoras derivadas del uso progresivo de la aplicación, manteniendo el enfoque de entrega continua establecido por Kanban.

Pruebas

El proceso de pruebas de Superllanta se desarrolló con el propósito de verificar el correcto funcionamiento de las funcionalidades implementadas y comprobar la interacción entre los componentes que forman la aplicación. Para tener un mejor orden en la documentación de pruebas, se han tomado como referencia los lineamientos de la ISO/IEC/IEEE 29119-3:2021, donde se considera la definición de escenarios, condiciones de ejecución, resultados esperados y las evidencias obtenidas durante el proceso de validación [24].

Estas pruebas fueron realizadas progresivamente durante el desarrollo del sistema. Las pruebas unitarias y de integración, fueron ejecutadas por medio del entorno controlado usando el backend local y la base de datos PostgreSQL destinada a pruebas, evitando así, modificar la información real del negocio. Posteriormente, las pruebas funcionales, de autenticación, trazabilidad y de rendimiento se ejecutaron sobre funcionalidades integradas, y cuando correspondía, sobre el sistema ya desplegado, con el objetivo de comprobar su comportamiento en condiciones cercanas al entorno de operación real.

El alcance de estas pruebas comprendió principalmente los módulos de autenticación y autorización, inventario, clientes, órdenes de trabajo, trazabilidad mediante códigos QR y la persistencia de información. Adicionalmente, se evaluaron funciones críticas del backend, como la comunicación entre los componentes de la aplicación, y el comportamiento de la API frente a carga concurrente.

Estrategia de pruebas

La estrategia de validación fue estructurada en diferentes niveles de prueba, seleccionados de acuerdo con las características del sistema y los riesgos asociados a las principales funcionalidades.

Esta estrategia contempló pruebas unitarias, pruebas de integración, funcionales, de autenticación, trazabilidad QR, autorización y de rendimiento. Las pruebas unitarias y de integración fueron automatizadas y ejecutadas sobre un entorno controlado, mientras que las pruebas funcionales, de acceso y trazabilidad, fueron verificadas mediante escenarios de uso representativo. Finalmente, las pruebas de rendimiento permitieron evaluar el comportamiento de la API frente a carga concurrente mediante k6 y complementar los resultados con métricas de operación obtenidas mediante Render.

Pruebas unitarias. Se definieron pruebas unitarias sobre funciones críticas del backend, entre ellas el cálculo de disponibilidad de stock al registrar un servicio con insumos, y la generación del código QR con el formato SL-{año}-{identificador}.

Las pruebas se implementaron utilizando el framework Jest, ejecutándose sobre las funciones `calcularStockDisponible` y `generarCodigoQR` del módulo `utils/reglasNegocio.js` del backend. Se definieron 21 pruebas unitarias en total como se muestra en la Figura 18: 11 sobre el cálculo de disponibilidad de stock y 10 sobre la generación de código QR (validando el formato esperado mediante expresión regular, el uso del año actual, la aceptación de identificadores tipo string, y el rechazo de identificadores nulos, indefinidos o vacíos). Las 21 pruebas resultaron exitosas, obteniendo el 100% de éxito, confirmando el correcto funcionamiento de ambas funciones ante los escenarios válidos e inválidos contemplados.

Figura 18

Resultados de la ejecución de pruebas unitarias (Jest).

```
PS C:\Users\super\Desktop\Superlanta\superlanta-backend> npm test
> superlanta-backend@1.0.0 test
> jest
PASS src/utils/ tests /reglasNegocio.test.js
  calcularStockDisponible
    ✓ descuenta la cantidad solicitada del stock actual (2 ms)
    ✓ permite dejar el stock exactamente en cero
    ✓ permite cantidades decimales válidas
    ✓ lanza error cuando la cantidad solicitada supera el stock disponible (9 ms)
    ✓ lanza RangeError cuando el stock actual es negativo (1 ms)
    ✓ lanza RangeError cuando la cantidad solicitada es cero (1 ms)
    ✓ lanza TypeError cuando la cantidad solicitada es negativa
    ✓ lanza TypeError cuando el stock no es un número
    ✓ lanza TypeError cuando la cantidad solicitada no es un número
    ✓ lanza TypeError cuando algún valor es NaN o Infinity (1 ms)
    ✓ permite que el stock inicial sea cero si no se solicita nada mayor a él
  generarCodigoQR
    ✓ genera el código con el formato SI-{año}-{identificador}
    ✓ acepta identificadores de tipo string
    ✓ usa el año actual cuando no se especifica (1 ms)
    ✓ respeta el formato esperado mediante una expresión regular (1 ms)
    ✓ lanza error si el identificador es undefined (1 ms)
    ✓ lanza error si el identificador es null (1 ms)
    ✓ lanza error si el identificador es una cadena vacía o solo espacios (1 ms)
    ✓ lanza error si el año no es un entero
    ✓ lanza error si el año es anterior a 2000
    ✓ acepta el identificador 0 como válido (1 ms)
Test Suites: 1 passed, 1 total
Tests: 21 passed, 21 total
Snapshots: 0 total
Time: 0.36 s, estimated 2 s
Ran all test suites.
```

Pruebas de integración. Las pruebas de integración verificaron la interacción entre la API REST desarrollada con Node.js y Express, la lógica de negocio, Prisma ORM y una base de datos PostgreSQL destinada a pruebas. Los escenarios comprobaron operaciones críticas relacionadas con inventario y trazabilidad QR, comprobando que los datos sean procesados y almacenados correctamente en la base de datos Figura 19. El detalle de los casos de prueba se presenta en la Tabla 17 del Capítulo IV.

Figura 19

Ejecución de las pruebas de integración del sistema mediante Jest

```
PS C:\Users\super\Desktop\Superlanta\superlanta-backend> npm run test:integration -- --verbose
> superlanta-backend@1.0.0 test:integration
> jest --config jest.integration.config.js --runInBand --verbose

PASS tests/integration/qr.integration.test.js (21.248 s)
  ✓ CP-INT-09: crear vulcanizado con neumático nuevo genera y persiste un código QR válido (3959 ms)
  ✓ CP-INT-10: cada neumático nuevo en el mismo servicio recibe un código QR único y persistente (5038 ms)
  ✓ CP-INT-11: reutilizar un neumático existente no genera un nuevo código QR (2401 ms)

PASS tests/integration/stock.integration.test.js (15.445 s)
  ✓ CP-INT-05: registrar salida de insumo descuento y persiste el stock disponible (2015 ms)
  ✓ CP-INT-06: salida mayor al stock disponible es rechazada y no altera el stock persistente (563 ms)
  ✓ CP-INT-07: registrar entrada de insumo incrementa el stock persistente (1300 ms)
  ✓ CP-INT-08: stock por debajo del mínimo genera y persiste una alerta (1804 ms)

PASS tests/integration/auth.integration.test.js (7.666 s)
  ✓ CP-INT-01: login con credenciales válidas devuelve token y persiste el log de acceso (1563 ms)
  ✓ CP-INT-02: login con contraseña incorrecta responde 401 sin generar token (425 ms)
  ✓ CP-INT-03: endpoint protegido sin token responde 401 (3 ms)
  ✓ CP-INT-04: /api/auth/me devuelve el perfil persistente correspondiente al token (1168 ms)

Test Suites: 3 passed, 3 total
Tests: 11 passed, 11 total
Snapshots: 0 total
Time: 44.498 s
Run all test suites.
```

Capítulo IV

Análisis y discusión de resultados

El proceso de validación técnica del sistema Superlanta se estructuró siguiendo los lineamientos del estándar ISO/IEC/IEEE 29119-3:2021, el cual establece la documentación y organización de las pruebas de software mediante planes de prueba, especificaciones de casos de prueba y reportes de resultados [24].

En este caso el proceso de validación del sistema se realizó con base en las evidencias recopiladas de diferentes tipos de pruebas, incluyendo pruebas funcionales, pruebas unitarias, pruebas de integración, la validación de la API REST, análisis estático y dinámico, así como la evaluación de los atributos de calidad de acuerdo con los estándares ISO/IEC 25010. Esta estrategia ayudó a verificar el correcto funcionamiento del software, su calidad y el cumplimiento de los requisitos.

Grado de cumplimiento del sistema

El Product Backlog final del proyecto estuvo conformado por 47 historias de usuario, de las cuales 39 correspondieron al alcance funcional desarrollado durante las primeras cinco iteraciones y 8 fueron incorporadas durante la sexta iteración como resultado del proceso de mejora continua del sistema.

Las 47 historias fueron implementadas, integradas y trasladadas al estado Finalizado en el tablero Kanban de Jira, alcanzando un 100% de cumplimiento del Product Backlog final. El porcentaje de cumplimiento de las historias de usuario se determinó mediante la siguiente expresión:

$$C_{HU} = \left(\frac{HU_{finalizadas}}{HU_{planificadas}} \right) \times 100$$

Donde:

$HU_{finalizadas}$: número de historias implementadas y trasladadas a Finalizado

$HU_{planificadas}$: total de historias del Product Backlog.

$$C_{HU} = \left(\frac{47}{47} \right) \times 100 = 100\%$$

El resultado evidencia que las 47 historias de usuario que conformaron el Product Backlog final fueron implementadas y finalizadas, obteniéndose un cumplimiento funcional del 100%.

Respecto a los requisitos no funcionales definidos en el Capítulo III, su cumplimiento fue evaluado mediante métricas cuantitativas y procedimientos de verificación acordes con cada atributo de calidad. La Tabla 15 presenta la métrica utilizada, el criterio de aceptación establecido, el método de validación, el resultado obtenido y el estado de cumplimiento de cada requisito no funcional.

Tabla 15*Matriz de cumplimiento de requisitos no funcionales*

RNF	Nombre	Métrica	Criterio de aceptación	Validado mediante	Resultado obtenido	Estado
RNF-01	Usabilidad	Tiempo promedio de ejecución de tareas operativas y porcentaje de tareas realizadas dentro del umbral	Promedio ≤ 5 min y al menos 90% de tareas ≤ 5 min	Observación directa y medición cronometrada durante tareas representativas	2 min 05 s de promedio; 9/9 tareas completadas en ≤ 5 min = 100%.	Cumple
RNF-02	Eficiencia de desempeño	Tiempo de respuesta de peticiones API y carga inicial de la aplicación	≤ 5 s	Chrome DevTools Network y PageSpeed Insights	344,5 ms de tiempo promedio en peticiones API. PageSpeed: 97/100; FCP 0,9 s; LCP 0,9 s; TBT 100 ms; Speed Index 1,1 s	Cumple

RNF	Nombre	Métrica	Criterio de aceptación	Validado mediante	Resultado obtenido	Estado
RNF-03	Disponibilidad	Porcentaje de comprobaciones exitosas del servicio durante una ventana de observación controlada	≥ 99% durante horario operativo	Shell de Render y endpoint /health, mediante 20 comprobaciones HTTP a intervalos de 5 segundos	20/20 comprobaciones respondieron HTTP 200 = 100% de disponibilidad durante la ventana evaluada.	Cumple
RNF-04	Seguridad	Porcentaje de casos de autenticación y autorización aprobados	100%	CP-AUTH-01 a CP-AUTH-03	3/3 casos aprobados = 100%	Cumple
RNF-05	Portabilidad	Porcentaje de configuraciones de pantalla evaluadas correctamente	100%	Chrome DevTools Device Toolbar y verificación responsive	4/4 configuraciones verificadas correctamente = 100%	Cumple

RNF	Nombre	Métrica	Criterio de aceptación	Validado mediante	Resultado obtenido	Estado
RNF-06	Escalabilidad	Variación del tiempo de respuesta ante incremento de carga	Incremento $\leq$ 20% respecto al escenario base	Prueba comparativa de carga mediante k6	P95 con 1 VU: 2,47 ms; P95 con 15 VUs: 4,81 ms. La variación del tiempo de respuesta fue de 94,74%.	No Cumple
RNF-07	Mantenibilidad	Hallazgos críticos y métricas de calidad/mantenibilidad del código	0 problemas críticos y Maintainability Rating A	SonarCloud, Qodana y revisión de arquitectura modular	Maintainability Rating A; duplicación 3,4%; Qodana ejecutó 168 inspecciones y detectó 28 hallazgos en 10 archivos, sin problemas clasificados como críticos	Cumple

Con el propósito de evidenciar la aplicación de las métricas definidas para los requisitos no funcionales, la Tabla 16 presenta las fórmulas utilizadas para cada RNF y la sustitución de los valores obtenidos durante la evaluación. Esta verificación permite contrastar matemáticamente los resultados medidos con los criterios de aceptación establecidos previamente en la Tabla 15.

Tabla 16

Evaluación matemática de los requisitos no funcionales

RNF	Fórmula aplicada	Cálculo
RNF-01	$\bar{t} = \frac{\sum t_i}{n} P_{tareas} = \frac{N_{\leq 5 \text{ min}}}{N_{total}} \times 100$	$125 \text{ s} \leq 300 \text{ s} P_{tareas} = \frac{9}{9} \times 100$ $= 100\%$
RNF-02	$P_{desempeño} = \frac{N_{cumplen}}{N_{evaluados}} \times 100$	$344,5 \text{ ms} = 0,3445 \text{ s}$ $\leq 5 \text{ s} P_{desempeño}$ $= \frac{5}{5} \times 100 = 100\%$
RNF-03	$A = \frac{N_{exitosas}}{N_{total}} \times 100$	$A = \frac{20}{20} \times 100 = 100\%$
RNF-04	$P_{seguridad} = \frac{CP_{aprobados}}{CP_{ejecutados}} \times 100$	$P_{seguridad} = \frac{3}{3} \times 100 = 100\%$
RNF-05	$P_{portabilidad} = \frac{C_{correctas}}{C_{evaluadas}} \times 100$	$P_{portabilidad} = \frac{4}{4} \times 100 = 100\%$
RNF-06	$\Delta t = \frac{t_{alta} - t_{base}}{t_{base}} \times 100$	$\Delta t = \frac{4,81 - 2,47}{2,47} \times 100$ $\approx 94,74\%$
RNF-07	$P_{críticos} = \frac{N_{críticos}}{N_{hallazgos}} \times 100 P_{dup}$ $= \frac{L_{duplicadas}}{L_{analizadas}} \times 100$	$\frac{0}{28} \times 100 = 0\% \frac{192}{5692} \times 100$ $\approx 3,37\% \approx 3,4\%$

Para la evaluación del RNF-06 se ejecutó una prueba comparativa específica usando k6, independiente de la prueba de carga progresiva presentada posteriormente en el análisis dinámico. En el escenario comparativo se mantuvo una carga de un usuario virtual durante 60 segundos y posteriormente una carga de 15 usuarios virtuales en el mismo intervalo, donde se obtuvieron valores P95 de 2,47 ms y 4,81 ms, respectivamente. Estos resultados fueron utilizados exclusivamente para calcular la variación porcentual del tiempo de respuesta.

Plan de pruebas

El plan de pruebas definido en el Capítulo III fue aplicado a los módulos críticos del sistema. Los resultados fueron documentados mediante escenarios de prueba, resultados esperados, evidencias visuales y el estado final de aprobación de cada caso.

Casos de prueba por módulo

La Tabla 17 presenta los casos de prueba seleccionados para verificar los principales módulos del sistema. Para cada escenario se estableció previamente un resultado esperado. Posteriormente, las evidencias visuales permiten comprobar el comportamiento obtenido durante su ejecución. Una vez presentadas las evidencias, se incorpora la evaluación final de cada caso mediante el estado.

Tabla 17

Casos de prueba por Módulo

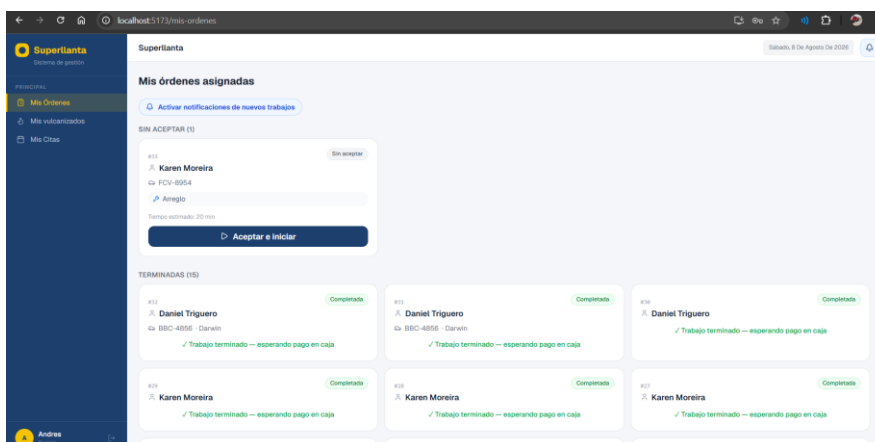
Código	Módulo	Escenario	Resultado esperado
CP-AUTH-01	Autenticación	Administrador inicia sesión con Google (Auth0)	Accede al dashboard completo del sistema
CP-AUTH-02	Autenticación	Técnico intenta acceder a /insumos escribiendo la URL manualmente	El sistema lo redirige a /mis-órdenes; no puede ver el inventario

Código	Módulo	Escenario	Resultado esperado
CP-AUTH-03	Autenticación	Cliente nuevo inicia sesión y su correo no coincide con ningún registro	El sistema solicita su cédula para intentar vincularlo a un registro existente
CP-INV-01	Inventario	Administrador registra un insumo con stock inicial 10 y stock mínimo 5	El insumo aparece en el listado con estado "Stock normal"
CP-INV-02	Inventario	El stock de un insumo baja de 5 a 3 tras un consumo	El sistema muestra alerta de "Stock bajo" en el dashboard
CP-QR-01	Trazabilidad QR	Técnico registra un arreglo para un neumático nuevo	El sistema genera un código QR único con formato SL-{año}-{id}
CP-QR-02	Trazabilidad QR	Se escanea el código QR generado en CP-QR-01 desde un celular	Se muestra la hoja de vida pública del neumático, sin requerir sesión
CP-ORD-01	Órdenes de trabajo	Administrador crea una orden sin asignar técnico	El sistema rechaza el formulario indicando que el técnico es obligatorio
CP-PAGO-01	Pagos electrónicos	Cliente paga con tarjeta vía PayPhone	El sistema confirma el pago automáticamente y envía la nota de venta

Evidencia de casos de prueba por módulo. La Figura 20 muestra el panel correspondiente al rol técnico después del intento de acceso a una ruta administrativa restringida. La evidencia permite demostrar que el usuario técnico permanece en su entorno autorizado y no accede al módulo de administración.

Figura 20

Control de acceso por rol: redirección del técnico ante una ruta restringida



La Figura 21 muestra la pantalla de confirmación de identidad en la que el sistema solicita datos adicionales para vincular la cuenta autenticada con un registro de cliente existente. Corresponde al comportamiento esperado cuando el correo autenticado no se encuentra asociado automáticamente.

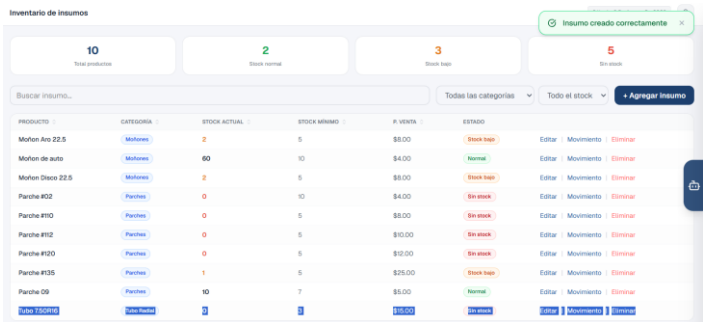
Figura 21

Solicitud de vinculación de identidad para un cliente no asociado

La Figura 22 evidencia que, después de guardar el nuevo insumo, el registro aparece en el listado de inventario y el sistema presenta un mensaje de confirmación. Esta evidencia permite verificar la persistencia visual del registro y la actualización del módulo de inventario.

Figura 22

Registro exitoso de un nuevo insumo en el inventario



La Figura 23 presenta el componente de alertas de stock con insumos cuya cantidad disponible se encuentra por debajo del nivel mínimo configurado. Evidencia la detección automática de existencias insuficientes.

Figura 23

Generación de alerta por stock bajo



La Figura 24 muestra el código QR generado por el sistema junto con su identificador único y los datos principales del neumático.

Figura 24

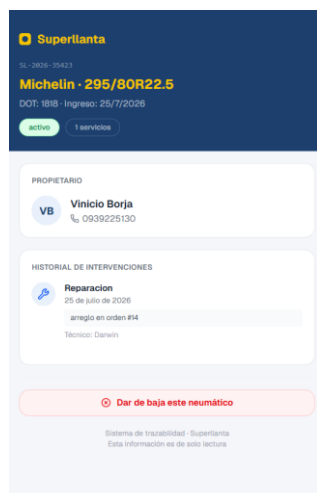
Generación de QR para un neumático



La Figura 25 es un ejemplo de la hoja de vida del neumático disponible después de leer el código QR. La hoja de vida incluye detalles sobre el propietario, el estado del neumático y el historial de intervenciones sobre el mismo.

Figura 25

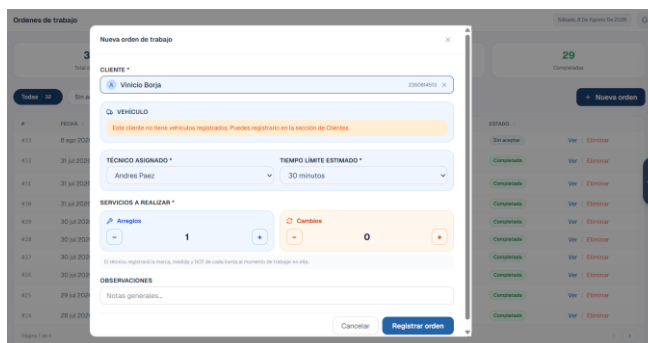
Pantalla con la hoja de vida digital escaneando el código QR



La Figura 26 muestra el mensaje que se presentará cuando se intente registrar una orden de trabajo, pero no se cuente con toda la información obligatoria.

Figura 26

Validación de datos obligatorios al registrar una orden de trabajo



La Figura 27 muestra la orden registrada con el estado “Completado y pagado”, lo que evidencia la actualización del estado de la transacción después del flujo de pago.

Figura 27

Confirmación de una orden completada y pagada

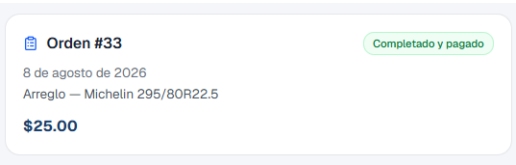


Tabla 18

Resultados de ejecución por módulo

Código	Resultado obtenido	Estado	Evidencia
CP-AUTH-01	Administrador accedió correctamente al dashboard completo	PASA	Figura 9 Resultado de la Iteración 1: núcleo funcional del sistema Superllanta
CP-AUTH-02	El técnico no accedió al módulo restringido	PASA	Figura 20
CP-AUTH-03	El sistema solicitó vinculación de identidad	PASA	Figura 21
CP-INV-01	El insumo fue registrado y mostrado en inventario	PASA	Figura 22
CP-INV-02	Se generó alerta al quedar bajo el stock mínimo	PASA	Figura 23
CP-QR-01	El sistema generó un QR único para el neumático	PASA	Figura 24

CP-QR-02	El QR permitió consultar la hoja de vida	PASA	Figura 25
CP-ORD-01	El sistema impidió guardar una orden con datos inválidos	PASA	Figura 26
CP-PAGO-01	La orden quedó registrada como completada y pagada	PASA	Figura 27

Análisis estático de código

Los resultados obtenidos mediante el análisis estático evidencian un nivel adecuado de calidad del código fuente del sistema Superllanta. La clasificación A en confiabilidad y mantenibilidad, junto con una tasa de duplicación del 3,4%, indica que el software presenta una adecuada organización y un bajo nivel de redundancia. Aunque se detectaron 13 vulnerabilidades y 339 "olores de código", estos hallazgos indican que existen oportunidades de mejora orientadas a elevar la calidad interna del código. Esto está en línea con la investigación de Yeboah y Popoola, quienes afirman que las herramientas de análisis de código estático, especialmente SonarQube, contribuyen en gran medida a la detección de defectos, problemas de mantenibilidad y áreas de mejora, lo cual contribuye a aumentar la calidad y la confiabilidad del software [25].

Tabla 19

Métricas de análisis estático de código

Métrica	Resultado	Descripción
Bugs detectados	0	Errores potenciales de confiabilidad identificados por la herramienta. SonarCloud clasificó el proyecto con Rating A en Reliability.
Vulnerabilidades	13	Patrones de código con riesgo de seguridad detectados por SonarCloud.

Métrica	Resultado	Descripción
Code Smells	339	Problemas relacionados con mantenibilidad y calidad del código.
Duplicación de código	3,4%	Porcentaje de líneas duplicadas detectadas sobre 5 692 líneas analizadas (192 líneas duplicadas).
Complejidad ciclomática	3,13	Promedio de rutas de decisión independientes por función (877 de complejidad total / 280 funciones).
Índice de mantenibilidad	A	Clasificación de mantenibilidad obtenida por SonarCloud, indicando un nivel alto de mantenibilidad.

La Figura 28 de SonarCloud muestra un Reliability Rating A y un Security Review Rating A, además de cero Security Hotspots visibles en el análisis. Sirve como evidencia visual complementaria de la evaluación estática del código.

Figura 28

Resultado del análisis estático: confiabilidad y seguridad del código



Análisis dinámico

Con el propósito de ampliar la evaluación del RNF-06, se ejecutó una segunda prueba de carga mediante k6 usando un escenario progresivo de hasta 15 usuarios virtuales durante 60 segundos. Esta prueba tuvo como objetivo observar el comportamiento general de la API durante el incremento gradual de carga. Se realizaron un total de 443 solicitudes HTTP durante la fase de ejecución, con un promedio de 7,34 solicitudes por segundo. La latencia promedio fue de 1,74 ms, la latencia al 95% fue de 2,89 ms y la latencia máxima fue de 103,29 ms. No hubo fallos en las solicitudes HTTP, por lo tanto, la tasa de éxito fue del 100% mientras que la tasa de error fue del 0%.

Sin embargo, el análisis de las métricas obtenidas en Render ha permitido determinar el uso de la memoria, el uso de CPU y el número de peticiones procesadas por el sistema en su ejecución real. Estos hallazgos son consistentes con las afirmaciones realizadas por Ilays et al., quienes argumentan que la implementación de pruebas sistemáticas en entornos de desarrollo

ágil ayuda a detectar fallos, optimizar el proceso de validación y mejorar la calidad del software antes de su lanzamiento al mercado [26].

Tabla 20

Métricas de análisis dinámico.

Métrica	Resultado	Procedimiento
Tiempo de respuesta promedio bajo carga	1,74 ms	Prueba k6 con escenario progresivo de hasta 15 VUs durante 60 s
Percentil 95 del tiempo de respuesta	2,89 ms	Métrica http_req_duration, con umbral p(95) < 1000 ms
Tiempo máximo de respuesta	103,29 ms	Resultado de http_req_duration durante la prueba k6
Rendimiento de la API	7,34 solicitudes/s	443 solicitudes procesadas durante 60,3 s
Tasa de solicitudes fallidas	0%	0 de 443 solicitudes HTTP fallidas
Verificaciones exitosas	886/886 = 100%	Checks de respuesta HTTP 200 y servidor saludable
Consumo de memoria del servidor	≈102 MB (≈20% de 512 MB)	Métricas de Render durante la operación del backend
Consumo de CPU del servidor	< 1%	Métricas de Render
Solicitudes observadas en producción	3 277 solicitudes	Panel de métricas de Render durante 7 días

La Figura 29 presenta las métricas de operación del backend desplegado en Render durante el periodo evaluado. Se observa un consumo de memoria cercano a 102 MB respecto al límite disponible de 512 MB, un uso de CPU inferior al 1% y un total de 3 277 solicitudes procesadas. Estos resultados permiten complementar la prueba de carga ejecutada con k6 mediante evidencia del consumo real de recursos del servicio durante su operación.

Figura 29

Métricas de consumo de recursos y solicitudes del backend en Render



La Figura 30 muestra los resultados obtenidos durante la prueba de carga ejecutada con k6. Se procesaron 443 solicitudes con una tasa de error del 0%, un tiempo promedio de respuesta de 1,74 ms y un percentil 95 de 2,89 ms. Los umbrales definidos para el tiempo de respuesta y la tasa de errores fueron satisfechos durante la ejecución.

Figura 30

Resultados de la prueba de carga del backend mediante k6



```

PS C:\Users\super\Desktop\Superllanta\Superllanta-backend> k6 run tests/performance/carga-api.js

      Grafana
      17.0
      17.0

execution: local
script: tests/performance/carga-api.js
web dashboard: http://127.0.0.1:5665
output:

scenarios: (100.00%) 1 scenario, 15 max VUs, 1m30s max duration (incl. graceful stop):
 * default: Up to 15 looping VUs for 1m30s over 3 stages (gracefulRampdown: 30s, gracefulStop: 30s)

THRESHOLDS
http_req_duration
✓ 'p(95)<1000' p(95)=2.89ms
http_req_failed
✓ 'rate<0.01' rate=0.00%

TOTAL RESULTS
checks_total.....: 896 14.688374/s
checks_succeeded...: 100.00% 896 out of 896
checks_failed.....: 0.00% 0 out of 896
✓ respuesta HTTP 200
✓ servidor saludable

HTTP
http_req_duration.....: avg=1.74ms min=0s med=1.44ms max=101.29ms p(90)=1.31ms p(95)=2.89ms
{ expected_response true }... avg=1.74ms min=0s med=1.44ms max=101.29ms p(90)=1.31ms p(95)=2.89ms
http_req_failed.....: 0.00% 0 out of 443
http_reqs.....: 443 7.344187/s

EXECUTION
iteration_duration.....: avg=1s min=1s med=1s max=1.1s p(90)=1s p(95)=1s
iterations.....: 443 7.344187/s
vus.....: 1 min=1 max=15
vus_max.....: 15 min=15 max=15

NETWORK
data_received.....: 173 kB 2.9 kB/s
data_sent.....: 34 kB 558 B/s

running (1m30.3s), 00/15 VUs, 443 complete and 0 interrupted iterations
default [=====] 00/15 VUs 1m30s
  
```

Capítulo V

Conclusiones

Se analizaron los procesos operativos de Superllanta mediante observación directa y entrevistas, identificando los requerimientos funcionales y no funcionales necesarios para el desarrollo del sistema, lo que permitió establecer una base adecuada para la automatización de los procesos del negocio.

Se diseñó e implementó la solución de software para la aplicación web en un modelo cliente-servidor usando un enfoque que consta de cuatro capas. React se utilizó para la capa de interfaz de usuario, mientras que Node.js y Express se utilizaron para la lógica del sistema. La capa de acceso a datos incluyó Prisma ORM y PostgreSQL como sistema gestor de bases de datos con el modelo relacional que consta de 28 tablas.

Como resultado de la implementación del enfoque Kanban, el proceso de desarrollo se dio de forma gradual a través de seis iteraciones. El Product Backlog final estuvo conformado

por 47 historias de usuario distribuidas en 13 módulos funcionales, todas implementadas y trasladadas al estado 'Finalizado'. La introducción gradual de las historias de usuario durante el curso del desarrollo ayudó a adaptar el sistema a los requerimientos descubiertos durante su uso real.

La evaluación de calidad permitió obtener resultados favorables en eficiencia de desempeño, seguridad, portabilidad y mantenibilidad. El análisis estático mediante SonarCloud presentó una clasificación A en mantenibilidad y confiabilidad, una duplicación de código del 3,4% y ausencia de bugs detectados, mientras que Qodana permitió identificar oportunidades de mejora relacionadas con la calidad interna del código. En la evaluación de escalabilidad, la variación del P95 entre 1 y 15 usuarios virtuales alcanzó el 94,74%, superando el límite del 20% definido para el RNF-06. No obstante, ambos escenarios mantuvieron una tasa de errores del 0% y tiempos de respuesta reducidos.

Recomendaciones

Se recomienda implementar un módulo de facturación electrónica integrado con el Servicio de Rentas Internas (SRI) del Ecuador, ya que actualmente el sistema genera únicamente una nota de venta no tributaria en formato PDF. La integración deberá evaluarse de acuerdo con la normativa tributaria vigente.

Aumentar las capacidades del módulo de informes agregando algunas nuevas métricas financieras, como la ganancia por categoría de servicio o por técnico. Esto permitiría al propietario a tomar decisiones de negocio más acertadas.

Continuar mejorando el sistema con base en los hallazgos identificados durante su implementación, como se hizo en la Iteración 6 del proyecto, siempre asegurando el enfoque en la mejora continua, priorizando las funcionalidades mediante el tablero Kanban.

Bibliografía

- [1] M. Rizky y R. Munir, «Data Transmission to Mobile Devices via Multiple QR Codes», en *2023 10th International Conference on Advanced Informatics: Concept, Theory and Application (ICAICTA)*, oct. 2023, pp. 1-6. doi: 10.1109/ICAICTA59291.2023.10390307.
- [2] Patrick. Li, «Jira Software Essentials: Plan, track, and release great applications with Jira Software». [En línea]. Disponible en: <https://ieeexplore.ieee.org/document/11130808>
- [3] M.-D. Marascu, R.-C. Constantinescu, y B. Alexandrescu, «An Integrative Approach to a Web Platform Development for the Management of a Sports Association», en *2025 IEEE 31st International Symposium for Design and Technology in Electronic Packaging (SIITME)*, oct. 2025, pp. 302-305. doi: 10.1109/SIITME67657.2025.11293604.
- [4] M. Vanessa, E. Andrés, R. Carrión, y M. Peña, «Inventory management for retail companies: A literature review and current trends», en *2021 Second International Conference on Information Systems and Software Technologies (ICI2ST)*, mar. 2021, pp. 71-78. doi: 10.1109/ICI2ST51859.2021.00018.
- [5] N. Omrani, N. Rejeb, A. Maalaoui, M. Dabić, y S. Kraus, «Drivers of Digital Transformation in SMEs», *IEEE Trans. Eng. Manag.*, vol. 71, pp. 5030-5043, 2024, doi: 10.1109/TEM.2022.3215727.
- [6] M. T. Tanaman, J. L. A. Baylosis, B. J. A. Abiles, M. L. P. Catungal, y P. C. Encarnacion, «Web-based Inventory Management System», *Int. J. Sci. Appl. Inf. Technol.*, vol. 12, n.º 5, pp. 44-48, oct. 2023, doi: 10.30534/ijisait/2023/021252023.
- [7] S. Suakanto, R. Y. Fa'rifah, W. A. Nurtrisha, y T. A. Nugroho, «Development of a Smart Asset Management System for Room Management in Buildings based on QR Code Technology», en *2023 10th International Conference on ICT for Smart Society (ICISS)*, sep. 2023, pp. 1-7. doi: 10.1109/ICISS59129.2023.10291247.
- [8] G. Misahuaman, A. Daza, y E. Zavaleta, «Web-based systems for inventory control in organizations: A Systematic Review», en *2021 IEEE/ACIS 22nd International Conference*

- on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, nov. 2021, pp. 15-20. doi: 10.1109/SNPD51163.2021.9704993.
- [9] S. Guan, W. Hu, y H. Zhou, «Front-end and Back-end Separation - React Based Framework for Networked Remote Control Laboratory», en *2018 37th Chinese Control Conference (CCC)*, jul. 2018, pp. 6314-6319. doi: 10.23919/ChiCC.2018.8483988.
- [10] R. T. Fielding y R. N. Taylor, «Principled design of the modern Web architecture», en *Proceedings of the 22nd international conference on Software engineering - ICSE '00*, Limerick, Ireland: ACM Press, 2000, pp. 407-416. doi: 10.1145/337180.337228.
- [11] International Organization for Standardization, «ISO/IEC 25010:2011 — Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models». ISO, Ginebra, Suiza, 2011. [En línea]. Disponible en: <https://www.iso.org/standard/35733.html>
- [12] J. P. Marroquin y C. R. Rodriguez, «Quality Assessment Model Based on ISO/IEC 25010 for E-Learning Platforms in the Cloud and Based on Laravel», en *2023 IEEE 15th International Conference on Computational Intelligence and Communication Networks (CICN)*, Bangkok, Thailand: IEEE, dic. 2023, pp. 751-757. doi: 10.1109/CICN59264.2023.10402281.
- [13] P. Keshari, P. Maurya, P. Kumar, y A. Katiyar, «Web Development Using ReactJS», en *2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, dic. 2023, pp. 1571-1575. doi: 10.1109/ICAC3N60023.2023.10541743.
- [14] R. Kouatly, O. Demirci, y D. Das, «Comparison of Object-Relational Mapping Frameworks on C#, Python, Go, and Node.js using MSSQL With A Case Study», en *2025 International Conference Automatics and Informatics (ICAI)*, Varna, Bulgaria: IEEE, oct. 2025, pp. 430-437. doi: 10.1109/ICAI67591.2025.11324528.

- [15] P. Varalakshmi, G. B, V. S. P, D. T, y S. K, «Improvising JSON Web Token Authentication in SDN», en *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, mar. 2022, pp. 1-8. doi: 10.1109/IC3IoT53935.2022.9767873.
- [16] M. O. Ahmad, J. Markkula, y M. Oivo, «Kanban in software development: A systematic literature review», en *2013 39th Euromicro Conference on Software Engineering and Advanced Applications*, sep. 2013, pp. 9-16. doi: 10.1109/SEAA.2013.28.
- [17] S. Ferrao y E. D. Canedo, «A study of the applicability of an agile methodology scrum allied to the Kanban method», jun. 2015, pp. 1-6. doi: 10.1109/CISTI.2015.7170382.
- [18] P. Maddigan y T. Susnjak, «Chat2VIS: Generating Data Visualizations via Natural Language Using ChatGPT, Codex and GPT-3 Large Language Models», *IEEE Access*, vol. 11, pp. 45181-45193, 2023, doi: 10.1109/ACCESS.2023.3274199.
- [19] K. Dodanduwa y I. Kaluthanthri, «Role of Trust in OAuth 2.0 and OpenID Connect», 22 de noviembre de 2018, *arXiv*: arXiv:1808.10624. doi: 10.48550/arXiv.1808.10624.
- [20] L. M. Reidl-Martínez, «Marco conceptual en el proceso de investigación», *Investig. En Educ. Médica*, vol. 1, n.º 3, pp. 146-151, sep. 2012.
- [21] R. Hernández Sampieri, C. Fernández Collado, y P. Baptista Lucio, *Metodología de la investigación*. McGraw Hill España, 2014. [En línea]. Disponible en: <https://dialnet.unirioja.es/servlet/libro?codigo=775008>
- [22] J. Lozada, «Investigación Aplicada: Definición, Propiedad Intelectual e Industria», *CienciAmérica Rev. Divulg. Científica Univ. Tecnológica Indoamérica*, vol. 3, n.º 1, pp. 47-50, 2014.
- [23] I. Sommerville, *Software engineering*, 10. ed. Boston, Munich: Pearson, 2016.
- [24] «IEEE/ISO/IEC International Standard for Software and systems engineering–Software testing–Part 3:Test documentation - Redline», *ISO/IEC/IEEE 29119-3:2021E - Redline*, pp. 1-274, oct. 2021.

- [25] J. Yeboah y S. Popoola, «Efficacy of static analysis tools for software defect detection on open-source projects», 2024, *arXiv*. doi: 10.48550/ARXIV.2405.12333.
- [26] I. Ilays *et al.*, «Towards Improving the Quality of Requirement and Testing Process in Agile Software Development: An Empirical Study», *Comput. Mater. Contin.*, vol. 80, n.º 3, pp. 3761-3784, 2024, doi: 10.32604/cmc.2024.053830.