

Implementación de un prototipo orientado a la web para la optimización de la gestión de inventarios y ventas en la empresa “ADIIC Dotaciones Institucionales” mediante la pila MERN (MERN Stack)

Juan Carlos Arrieta Bustos

Richard Pardo Cardona

Trabajo de grado

Directora

Mg. Arellys de Jesús Correa Rodríguez

Universidad Nacional Abierta y a Distancia - UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería – ECBTI

Programa Ingeniería de Sistemas

2025

Dedicatoria

Este trabajo de grado está dedicado, en primer lugar, a nuestras familias, quienes con su amor, apoyo incondicional y comprensión han sido nuestro pilar fundamental durante este proceso académico. Gracias por motivarnos a seguir adelante incluso en los momentos más exigentes. Dedicamos también este logro a nuestros docentes, especialmente a nuestra directora de trabajo de grado, Mg. Arellys de Jesús Correa Rodríguez, por su guía, compromiso y valiosos aportes a lo largo de este proyecto.

A nuestras amistades y compañeros de estudio, gracias por ser parte de este camino, por el intercambio de conocimientos y el acompañamiento constante.

Finalmente, nos dedicamos este logro a nosotros mismos, Juan Carlos Arrieta Bustos y Richard Pardo Cardona, por la perseverancia, el esfuerzo y la convicción con la que asumimos este desafío, demostrando que con disciplina y pasión es posible alcanzar nuestras metas.

Agradecimientos

Expresamos nuestra más sincera gratitud a Dios, fuente de fortaleza e inspiración, por guiarnos en cada paso de este proceso académico.

Agradecemos a la Universidad Nacional Abierta y a Distancia – UNAD, por brindarnos las herramientas, conocimientos y oportunidades para nuestro crecimiento profesional y personal.

A nuestra directora de trabajo de grado, Mg. Arellys de Jesús Correa Rodríguez, por su orientación, compromiso, paciencia y valiosos aportes, que fueron esenciales para el desarrollo de este proyecto.

Nuestro reconocimiento especial a la empresa ADIIC Dotaciones Institucionales, por abrirnos sus puertas y permitirnos aplicar nuestros conocimientos en un entorno real, aportando a la mejora de sus procesos.

A nuestras familias, por su amor incondicional, apoyo constante y por ser nuestro principal motor para continuar. A nuestros amigos y compañeros de carrera, por acompañarnos en este camino con palabras de aliento, trabajo colaborativo y espíritu solidario.

Finalmente, agradecemos a todas las personas que, de una u otra forma, hicieron parte de este logro. Cada palabra de aliento, cada consejo y cada gesto de apoyo fueron fundamentales para culminar con éxito este proceso.

Resumen

En muchas pequeñas y medianas empresas, la gestión manual de inventarios y ventas produce errores operativos, limita la trazabilidad de los procesos y retrasa la toma de decisiones. Este fue el caso de la empresa ADIIC Dotaciones Institucionales, la cual presentaba dificultades en el control de su inventario y en la organización de sus ventas. Con el fin de mejorar esta situación, se desarrolló un prototipo de aplicación web basado en una arquitectura cliente-servidor y construido mediante la pila tecnológica MERN (MongoDB, Express.js, React.js y Node.js). El trabajo se estructuró en seis fases que comprenden desde el análisis del problema hasta la entrega del sistema funcional. En primer lugar, se contextualizó la problemática y se planteó su justificación. En segundo lugar, se realizó una revisión bibliográfica sobre sistemas de gestión de inventarios y metodologías ágiles. Posteriormente, se definió SCRUM como marco metodológico para guiar el desarrollo iterativo e incremental del prototipo. A continuación, se abordaron las fases de diseño e implementación. Finalmente, se presentó el sistema a la empresa y se analizaron las observaciones recibidas. Como resultado, se obtuvo una herramienta digital que permite gestionar en tiempo real productos, categorías, clientes y facturación, facilitando así decisiones operativas relacionadas con existencias y ventas.

Palabras claves: Aplicación web, pila MERN, SCRUM, API, framework.

Abstract

In many small and medium-sized enterprises, manual inventory and sales management often results in errors, limited traceability, and delays in decision-making. Such was the situation at ADIIC Dotaciones Institucionales, a company struggling to maintain control over its inventory and sales processes. To address these challenges, a web-based prototype was designed and implemented using a client-server architecture and the MERN stack (MongoDB, Express.js, React.js, and Node.js). The project followed six phases, ranging from problem analysis to the delivery of a functional system. Initially, the problem was contextualized and its relevance justified. A literature review was then conducted on inventory management systems and agile methodologies, leading to the adoption of the Scrum framework to guide the iterative and incremental development process. The subsequent phases involved the design, implementation, and evaluation of the prototype. The final product is a digital tool that enables real-time management of products, categories, clients, and billing, providing greater accuracy and supporting informed decision-making within the company's operations.

Keywords: Web application, MERN Stack, SCRUM, API, framework.

Tabla de Contenido

Portada	1
Dedicatoria	2
Agradecimientos	3
Resumen.....	4
Abstract	5
Tabla de Contenido	6
Lista de Tablas	11
Lista de Figuras.....	12
Lista de Anexos.....	15
Introducción	16
Planteamiento del Problema	18
Justificación	21
Objetivos	22
Objetivo General	22
Objetivos Específicos.....	22
Marco Referencial.....	23
Estado del Arte	23
Marco Contextual.....	27
Marco Conceptual	28

La Gestión de Inventarios.....	28
Pila MERN (MERN Stack).....	29
Business Process Management (BPM).....	30
Patrón Modelo-Vista-Controlador (MVC).....	31
Marco Teórico.....	31
Inventarios.....	32
Ingeniería de Software.....	33
Paradigmas de Programación.....	36
Desarrollo Orientado a la Web (DOW).....	38
Marco Legal.....	40
Sistema de Gestión de Seguridad y Salud en el Trabajo.....	40
Dotaciones Institucionales.....	43
Control de Inventarios.....	45
Facturación Electrónica.....	46
Protección de datos personales.....	47
Otras Normas.....	47
Artefactos de cumplimiento exigidos por el proyecto:.....	48
Metodología.....	49
Tipo de Estudio.....	49

Enfoque Metodológico	49
Técnicas e Instrumentos de Recolección de Información	49
Metodologías Implementadas	50
Fases del Proyecto	52
Análisis de Procesos de Inventario y Ventas con Enfoque BPM	52
Diseño de la Arquitectura del Sistema de Gestión de Inventario y Ventas	52
Desarrollo Iterativo del Sistema de Inventario y Ventas	53
Evaluación de Usabilidad del Prototipo del Sistema de Inventario y Ventas	53
Análisis de los Procesos de Adquisición y Venta	54
Plan de Obtención de Requisitos	54
Entrevistas Individuales	54
Identificación de Roles en ADIIC Dotaciones Institucionales	55
Descripción del Proceso de Adquisición y Ventas	56
Rediseño del Proceso de Adquisición y Venta	60
Rediseño Proceso de Adquisición.	61
Rediseño Proceso de Venta.	61
Definición de las Necesidades	62
Diagrama Estado Futuro del Proceso de Adquisición TO-BE	64
Diagrama Estado Futuro del Proceso de Venta TO-BE	65

Definición del Producto Mínimo Viable (MVP) del Sistema.....	67
Definición de Épicas	69
Historias de Usuario	71
Mapa de Trazabilidad del Producto Mínimo Viable (MVP).....	88
Diseño de la Arquitectura del Sistema.....	91
Tipo de Arquitectura Aplicada.....	91
Selección del Modelo de Vista de Arquitectura.....	93
Selección de Puntos de Vista para la Arquitectura	96
Vista de Contexto	97
Vista de Información	99
Vista de Despliegue	102
Vista Funcional.....	104
Vista de Desarrollo	107
Vista de Concurrencia	111
Vista de Seguridad y Autenticación	114
Desarrollo del Prototipo Mediante el Marco Ágil Scrum.....	120
Tipo de Prototipo.....	120
Configuración del Entorno (Sprint 0).....	120
Product Backlog	122

Desarrollo Iterativo del Proyecto (Sprints)	148
Resultados	162
Resumen de Cumplimiento por Módulo (MVP).....	163
Consolidado de pruebas unitarias por módulo	166
Conclusiones y Recomendaciones	168
Conclusiones	168
Recomendaciones.....	169
Referencias bibliográficas.....	171
Anexos	181

Lista de Tablas

Tabla 1 <i>Comparación de soluciones disponibles</i>	24
Tabla 2 <i>Matriz de tecnologías y casos en PyMEs colombianas</i>	26
Tabla 3 <i>Asignación de roles</i>	55
Tabla 4 <i>Necesidades del estado futuro para los procesos de adquisición y venta</i>	63
Tabla 5 <i>Épicas funcionales del sistema de gestión de inventario y venta</i>	70
Tabla 6 <i>Mapa de trazabilidad del MVP: relación entre épicas e historias de usuario</i>	88
Tabla 7 <i>Descripción de vistas arquitectónicas adoptadas</i>	94
Tabla 8 <i>Product backlog del proyecto</i>	123

Lista de Figuras

Figura 1 <i>¿Como funciona la pila MERN?</i>	30
Figura 2 <i>Flujo de Scrum para un sprint</i>	51
Figura 3 <i>Diagrama de flujo proceso de adquisiciones</i>	58
Figura 4 <i>Diagrama de flujo proceso de ventas</i>	59
Figura 5 <i>Diagrama de flujo proceso de adquisición TO-BE</i>	65
Figura 6 <i>Diagrama de flujo proceso de venta TO-BE</i>	67
Figura 7 <i>Formato tarjeta de historia de usuario</i>	72
Figura 8 <i>Historia de usuario HU001 - Página de inicio con login integrado</i>	72
Figura 9 <i>Historia de usuario HU002 – Formulario de login con redirección por rol</i>	73
Figura 10 <i>Historia de usuario HU003 – Sidebar para acceso a funcionalidades de gestión</i>	73
Figura 11 <i>Historia de usuario HU004 – Encabezado de la aplicación Web</i>	74
Figura 12 <i>Historia de usuario HU005 – Página de administración de categorías y productos</i>	74
Figura 13 <i>Historia de usuario HU006 – Registro de nuevos usuarios con asignación de rol</i>	75
Figura 14 <i>Historia de usuario HU007 – Creación de nuevas categorías</i>	75
Figura 15 <i>Historia de usuario HU008 – Actualización de categorías existentes</i>	76
Figura 16 <i>Historia de usuario HU009 – Actualización de información de un producto</i>	76
Figura 17 <i>Historia de usuario HU010 – Registro de nuevos productos asociados a categorías</i>	77
Figura 18 <i>Historia de usuario HU011 – Listado de productos por categoría</i>	77
Figura 19 <i>Historia de usuario HU012 – Visualización y gestión de productos de una categoría específica</i>	78
Figura 20 <i>Historia de usuario HU013 – Visualización de categorías por parte del usuario regular (Vendedor)</i>	78

Figura 21 <i>Historia de usuario HU014 – Visualización de productos por categoría para vendedores</i>	79
Figura 22 <i>Historia de usuario HU015 – Creación de cabeceras para facturas por sucursal</i>	79
Figura 23 <i>Historia de usuario HU016 – Edición de cabeceras</i>	80
Figura 24 <i>Historia de usuario HU017 – Eliminación de cabeceras innecesarias</i>	80
Figura 25 <i>Historia de usuario HU018 – Búsqueda de cabeceras por NIT en gestión administrativa</i>	81
Figura 26 <i>Historia de usuario HU019 – Selección de cabecera al crear una factura</i>	81
Figura 27 <i>Historia de usuario HU020 – Creación de nueva factura</i>	82
Figura 28 <i>Historia de usuario HU021 – Agregar productos con cantidad, descuento e IVA a la factura</i>	82
Figura 29 <i>Historia de usuario HU022 – Cálculo automático de totales en la factura</i>	83
Figura 30 <i>Historia de usuario HU023 – Verificación automática del stock al facturar</i>	83
Figura 31 <i>Historia de usuario HU024 – Edición de factura antes de finalizar</i>	84
Figura 32 <i>Historia de usuario HU025 – Descarga de factura finalizada en PDF</i>	84
Figura 33 <i>Historia de usuario HU026 – Búsqueda de facturas por número o NIT (Administrador)</i>	85
Figura 34 <i>Historia de usuario HU027 – Descuento automático del stock al generar factura</i>	85
Figura 35 <i>Historia de usuario HU028 – Consulta del historial de facturación</i>	86
Figura 36 <i>Historia de usuario HU029 – Visualización de lista de clientes con compras (Administrador)</i>	86
Figura 37 <i>Historia de usuario HU030 – Búsqueda de clientes por NIT o identificación (Administrador)</i>	87

Figura 38 <i>Historia de usuario HU031 – Visualización del número de compras por cliente</i>	87
Figura 39 <i>Historia de usuario HU032 – Registro automático de cliente al realizar la primera compra</i>	88
Figura 40 <i>Modelo cliente-servidor de tres capas</i>	91
Figura 41 <i>Diagrama de arquitectura del sistema</i>	93
Figura 42 <i>Árbol de carpetas en VS code del frontend y backend del proyecto</i>	121
Figura 43 <i>Página de inicio/login y pagina de registro de usuario</i>	149
Figura 44 <i>Páginas de creación de categorías y productos</i>	152
Figura 45 <i>Página de elaboración de factura y modal de factura creada</i>	155
Figura 46 <i>Visualización de factura generada</i>	159

Lista de Anexos

Anexo 1 <i>Caso de negocio (Business Case)</i>	181
Anexo 2 <i>Visión del proyecto</i>	182
Anexo 3 <i>Acta de constitución de proyecto</i>	183
Anexo 4 <i>Formulario para entrevista de toma de requisitos</i>	184
Anexo 5 <i>Formulario pruebas unitarias proyecto ADIIC</i>	185
Anexo 6 <i>Desarrollo del proyecto por épicas</i>	186
Anexo 7 <i>Certificado de registro de soporte lógico – software</i>	187
Anexo 8 <i>Manual de usuario versión 1.0</i>	188

Introducción

La gestión eficiente de inventarios es un factor crítico para la sostenibilidad operativa de las pequeñas y medianas empresas, pues impacta directamente en la disponibilidad de productos, la satisfacción del cliente y el control financiero. En países como Colombia, donde gran parte del tejido empresarial corresponde a pymes, la persistencia de registros manuales o sistemas fragmentados representa un obstáculo recurrente para la competitividad y el crecimiento.

La empresa ADIIC Dotaciones Institucionales, dedicada a la comercialización de dotaciones para entidades públicas y privadas, enfrenta limitaciones significativas derivadas del uso de métodos manuales. Entre ellas se encuentran la actualización tardía del inventario, inconsistencias en los registros, retrasos en la facturación, pérdida de información crítica, ausencia de reportes en tiempo real y falta de indicadores que permitan evaluar la productividad o el nivel de servicio. Estas deficiencias generan reprocesos, errores recurrentes y dificultades para responder a auditorías o participar en licitaciones que exigen trazabilidad confiable, lo que impacta de manera directa en su competitividad.

El objetivo del proyecto fue diseñar e implementar una aplicación web para optimizar la gestión de inventarios, productos, categorías, clientes y facturación, mejorando la eficiencia de las operaciones comerciales. Para ello, se aplicó un enfoque basado en Business Process Management (BPM) y la metodología ágil Scrum, con el soporte tecnológico de la pila MERN (MongoDB, Express.js, React.js y Node.js).

El impacto esperado del prototipo desarrollado trasciende la simple digitalización de registros. Se proyecta una reducción de tiempos en el proceso de facturación y en la consulta de inventario, una mejora en la precisión de los datos, mayor trazabilidad de productos, fortalecimiento de la toma de decisiones estratégicas y la posibilidad de escalar la solución hacia

módulos adicionales como reportes analíticos o integración con facturación electrónica. En consecuencia, el sistema no solo contribuye a optimizar la operación interna de ADIIC, sino que también le otorga una ventaja competitiva al mejorar su capacidad de respuesta, su cumplimiento normativo y su eficiencia en la gestión empresarial.

En síntesis, este trabajo de grado busca aportar una solución tecnológica viable y contextualizada a las necesidades reales de la empresa, fortaleciendo la trazabilidad de sus operaciones y, al mismo tiempo, generando un aporte académico al estudio y aplicación de metodologías ágiles y tecnologías web en el contexto empresarial colombiano.

Planteamiento del Problema

El gobierno de Colombia, a través del Ministerio de Trabajo, ha establecido el Sistema de Gestión de Seguridad y Salud en el Trabajo (SG-SST) como política obligatoria para todos los empleadores públicos y privados, con el propósito de prevenir accidentes laborales, enfermedades profesionales y garantizar ambientes laborales seguros (Congreso de la República de Colombia, 2012a; Presidencia de la República de Colombia, 2015b). En este sentido, dicha normativa exige el desarrollo de un proceso estructurado que involucre desde la planificación hasta la mejora continua (Colombia. Ministerio del Trabajo, 2017). De manera complementaria, el SG-SST integra acciones sistemáticas y coordinadas que promueven la gestión efectiva del riesgo laboral en todas las organizaciones.

En concordancia con este marco normativo, la empresa ADIIC Dotaciones Institucionales S.A.S., con más de 14 años de trayectoria en el mercado nacional, ha consolidado su operación en la confección y comercialización de dotaciones institucionales, elementos de protección personal (EPP) y calzado de seguridad industrial. A través de su portafolio de productos, la empresa cumple un rol fundamental como proveedora de insumos esenciales que permiten a otras organizaciones dar cumplimiento efectivo a los lineamientos establecidos por el SG-SST, contribuyendo así a la prevención de riesgos laborales y la promoción de entornos seguros.

Actualmente, la empresa gestiona sus procesos operativos mediante registros manuales y hojas de cálculo, tanto para el control de inventarios como para el registro de ventas. El inventario se actualiza de forma manual una vez finalizadas las ventas diarias, lo que genera desfases entre la disponibilidad real de productos y la información registrada. Este procedimiento carece de sincronización con el área comercial, ya que los pedidos se reciben por distintos canales sin una integración automatizada que permita la actualización en tiempo real. Como

resultado, se presentan errores frecuentes, pérdida de información, reprocesos y dificultades en el acceso oportuno a los datos operativos. Esta problemática no solo afecta la eficiencia interna de ADIIC Dotaciones Institucionales S.A.S., sino que también refleja una situación común en muchas pequeñas y medianas empresas (PYMES) colombianas, donde la ausencia de herramientas tecnológicas limita la gestión efectiva de los recursos y procesos críticos (González Miller, 2024; Tovar-Cardozo & Sierra-García, 2021).

No obstante, ADIIC Dotaciones Institucionales S.A.S. enfrenta limitaciones operativas significativas debido a la falta de una solución tecnológica que respalde una gestión integrada de inventario y ventas. Esta desconexión entre procesos impide conocer en tiempo real el estado de las existencias y afecta directamente la trazabilidad de los productos, generando fallos en la atención a clientes, duplicidad de tareas y pérdida de eficiencia en la cadena operativa. La empresa carece de un sistema que permita alinear dinámicamente la oferta de productos con la demanda registrada por el área comercial, lo que limita la toma de decisiones basadas en datos y dificulta el cumplimiento de objetivos estratégicos.

De no abordarse oportunamente esta deficiencia tecnológica, la empresa podría enfrentar múltiples consecuencias adversas en el mediano y largo plazo. En primer lugar, la ausencia de un sistema moderno de gestión dificulta el control eficiente del inventario, comprometiendo la trazabilidad de los productos y generando inconsistencias que afectan la calidad del servicio ofrecido. Esto, a su vez, limita la capacidad de respuesta ante auditorías, procesos de certificación o licitaciones que exigen evidencias documentadas del flujo de productos, impactando negativamente en la imagen institucional frente a clientes y aliados estratégicos. Aunque ADIIC Dotaciones Institucionales S.A.S. no está obligada a cumplir con los estándares del SG-SST, dado que su rol es de proveedora de insumos y no de usuaria directa del sistema, sí

podría enfrentar consecuencias normativas en otros frentes. Particularmente, el no contar con una solución tecnológica que garantice la emisión adecuada de la factura electrónica, conforme a lo estipulado en la Resolución 000165 de 2023, podría derivar en sanciones tributarias por parte de la DIAN (Unidad Administrativa Especial Dirección de Impuestos y Aduanas Nacionales (DIAN), 2023). Además, desde una perspectiva competitiva, la falta de digitalización impide la optimización de los procesos internos, lo que puede traducirse en sobrecostos, pérdida de oportunidades comerciales y disminución de la participación en el mercado frente a competidores que sí han adoptado soluciones tecnológicas innovadoras.

En respuesta a este panorama, el proyecto aplicado que aquí se desarrollará busca cerrar esta brecha mediante el diseño e implementación de un prototipo web desarrollado con la pila MERN (MongoDB, Express.js, React.js y Node.js), orientado a optimizar la gestión de inventarios y ventas en la empresa. Esta solución tecnológica permitirá integrar de forma eficiente los procesos operativos, mejorar la trazabilidad de los productos y fortalecer la toma de decisiones basada en datos, contribuyendo así a la eficiencia operativa y al cumplimiento de los requisitos funcionales y comerciales propios del entorno empresarial actual.

A partir de lo anterior, se formula la siguiente pregunta de investigación: ¿Cómo optimizar la gestión de inventarios y ventas en la empresa ADIIC Dotaciones Institucionales S.A.S. mediante un prototipo web desarrollado con la pila MERN (MERN Stack)?

Justificación

La situación actual de ADIIC Dotaciones Institucionales presenta registros dispersos, doble digitación y dependencia de hojas de cálculo, lo que expone a inconsistencias, reprocesos y pérdida de trazabilidad en inventarios y ventas. Este contexto limita la visibilidad operativa y eleva el riesgo de errores que impactan la continuidad del negocio.

Asimismo, la organización enfrenta exigencias crecientes de trazabilidad, auditoría y reporte que requieren evidencia sistemática, controles de acceso y registros transaccionales confiables. Mantener el statu quo incrementa la probabilidad de desalineación entre el stock físico y el registrado, demoras en el ciclo de venta y mayor exposición a incumplimientos en procesos administrativos y comerciales.

Desde la perspectiva estratégica, consolidar la información en una plataforma transaccional habilita medición objetiva de procesos, estandarización de flujos y preparación para integraciones futuras (por ejemplo, facturación electrónica y analítica de datos). La elección de un MVP reduce el riesgo de adopción, acelera el aprendizaje organizacional y concentra el esfuerzo en las funcionalidades que realmente generan valor.

En el plano académico-profesional, el proyecto documenta la aplicación de BPM y Scrum con la pila MERN en un escenario real de PyME colombiana, aportando un caso de estudio transferible y buenas prácticas para desarrollos iterativos en contextos similares. Este aporte metodológico fortalece la formación de los desarrolladores y ofrece una referencia útil para trabajos posteriores.

En síntesis, la justificación radica en mitigar riesgos operativos y de cumplimiento, y en crear capacidades organizacionales medibles mediante una solución incremental, sostenible y alineada con los objetivos estratégicos de la empresa.

Objetivos

Objetivo General

Implementar un prototipo orientado a la Web para la gestión de la información en la empresa de “*ADIIC Dotaciones Institucionales*” mediante MERN Stack.

Objetivos Específicos

Analizar la gestión del inventario y el proceso de ventas en la empresa “*ADIIC Dotaciones Institucionales*”.

Diseñar la arquitectura de solución del sistema para la gestión de inventario y ventas en la empresa “*ADIIC Dotaciones Institucionales*”.

Desarrollar el sistema para la gestión de inventario y ventas en la empresa “*ADIIC Dotaciones Institucionales*”.

Marco Referencial

En el presente apartado se establecen los fundamentos que orientan y sustentan el desarrollo del proyecto. Para ello, se abordan de manera articulada cinco componentes: el estado del arte, que recopila y analiza antecedentes relevantes; el marco contextual, que describe el entorno social, organizacional y tecnológico en el cual se sitúa la investigación; el marco conceptual, que define los términos y categorías clave empleados; el marco teórico, que sustenta las teorías y enfoques relacionados; y el marco legal, que identifica las normas y regulaciones aplicables relacionados con la gestión de inventarios y ventas en el contexto de la empresa ADIIC Dotaciones Institucionales.

De esta manera, este marco referencial proporciona una visión integral de las diferentes dimensiones que influyen en la implementación exitosa de un sistema de gestión de inventarios y venta en entornos empresariales.

Estado del Arte

En el panorama actual del desarrollo web, la pila MERN (MongoDB, Express.js, React.js y Node.js) **se consolida** como una opción versátil y eficiente. Su arquitectura modular **favorece** la escalabilidad y la separación de responsabilidades, habilitando aplicaciones robustas capaces de manejar altos volúmenes de datos y concurrencia (p. ej., comercio electrónico o gestión de inventarios). La curva de aprendizaje y la amplitud de su comunidad **facilitan** la conformación de equipos y la evolución continua del stack, lo que fortalece su adopción en contextos empresariales diversos. (Abbas et al., 2022; Chandrashekhhar Bakale et al., 2023; Prajapati et al., 2023; Teja Labba et al., 2023).

En paralelo, la gestión de inventarios se posiciona como proceso crítico en PyMEs, impulsando soluciones web y móviles para asegurar trazabilidad, disponibilidad y control. La

literatura reporta mejoras en eficiencia, reducción de errores y soporte a la toma de decisiones mediante aplicaciones específicas de inventario y facturación, tanto en entornos de servicios como en comercio minorista (Díaz Pereira, 2020; González Miller, 2024; Tovar-Cardozo & Sierra-García, 2021). Otros trabajos destacan el papel de la transformación digital y los sistemas de gestión (ERP) para centralizar información y automatizar procesos, potenciando la competitividad y productividad (Lazo Díaz et al., 2021; Mera Rodríguez et al., 2019; Soledispa-Rodríguez et al., 2021).

Asimismo, se observa una tendencia consistente al uso de frameworks y metodologías ágiles con énfasis en iteraciones, validación temprana y despliegues incrementales para acelerar la entrega de valor (Capece et al., 2021; Herrero Romero, 2022; Mex Álvarez et al., 2024; Moragas Pardell, 2023). Estos referentes sustentan la elección de un enfoque basado en MVP con pila MERN para ADIIC Dotaciones Institucionales.

Con el fin de aterrizar la revisión al contexto de ADIIC y evaluar alternativas de adopción o integración, se presenta a continuación una comparación sintética de soluciones relevantes para PyMEs colombianas en inventario y facturación.

Tabla 1

Comparación de soluciones disponibles

Solución	Modelo	Alcance funcional (resumen)	Integraciones/API (ejemplos)	Adecuación a ADIIC	Esquema de costo (general)	Observaciones
Odoo	Open source / SaaS u on-prem	Inventario, Ventas, Compras, Facturación, Reportes	Módulos e integraciones; API REST	Media-Alta	Suscripción por usuario/módulo o auto-hosting	Flexible y robusto; requiere parametrización y gobierno de cambios.
ERPNext	Open source / self-hosted/cloud	Inventario, Ventas, CRM, Contabilidad	API; apps ecosistema Frappe	Media	Infraestructura + soporte opcional	Bajo costo total de propiedad; exige equipo

						técnico y curva de aprendizaje alta.
Alegra	SaaS (LatAm)	Facturación, Inventario, Contabilidad básica	Integraciones contables; conectores	Alta	Suscripción mensual	Adopción rápida; orientado a operación y cumplimiento diario.
Siigo	SaaS (LatAm)	Facturación, Inventario, Nómina/Contable	Integraciones contables; conectores	Alta	Suscripción mensual	Amplia base en Colombia; enfoque administrativo y fiscal.
Vendty	SaaS POS	POS, Inventario, Clientes, Reportes	Integraciones POS/e-commerce	Media-Alta	Suscripción mensual	Despliegue ágil retail; reportes avanzados limitados.
Shopify POS	SaaS (POS omnicanal)	POS, catálogo, inventario omnicanal	Ecosistema Shopify; API e-commerce	Media	Suscripción + add-ons	Fuerte en omnicanalidad; depende del ecosistema Shopify.

Nota. La estimación de “Adecuación” ponderó facilidad de adopción, cobertura mínima del MVP y dependencia de recursos técnicos internos; los costos se muestran como esquema general, sin cifras específicas.

Para complementar la comparación de soluciones con evidencia en contexto local, se incluye a continuación una matriz de tecnologías y estudios de caso en PyMEs colombianas, con hallazgos prácticos y una valoración de adecuación al caso ADIIC.

Tabla 2*Matriz de tecnologías y casos en PyMEs colombianas*

Caso / Sector (tamaño)	Solución / Stack	Hosting	Alcance / Integraciones clave	Hallazgos	Adecuación a ADIIC	Fuente
Retail de dotaciones (pequeña)	SaaS de facturación + inventario (Alegra/Siigo)	Nube (proveedor)	Inventario + ventas + reportes básicos	Adopción rápida; soporte local	Alta	Entrevistas internas / experiencia sector
Ferretería (mediana)	Odoo (módulos Inventario/Ventas)	On-premise / VPS	Flujos complejos; personalizaciones	Gran flexibilidad; mayor mantenimiento	Media	Literatura técnica / casos públicos
Tienda de calzado (pequeña)	Vendty (POS + inventario)	Nube (proveedor)	POS integrado; reportes	Despliegue ágil; reportes avanzados limitados	Media-Alta	Experiencia de campo / reseñas técnicas
E-commerce PyME (pequeña)	Shopify + POS	Nube (Shopify)	Catálogo omnicanal; pagos	Excelente omnicanalidad; dependencia del ecosistema	Media	Documentación técnica / práctica sector
Servicios industriales (pequeña)	ERPNext (Frappe)	Self-hosted / cloud	Inventario + compras + ventas	TCO bajo; requiere equipo técnico	Media	Comunidad FOSS / documentación

Nota. La matriz resume evidencias provenientes de literatura técnica, documentación de

producto y experiencias reportadas en PyMEs colombianas.

La revisión comparativa evidencia que las soluciones SaaS locales (Alegra, Siigo) y los POS especializados (Vendty) resultan altamente adecuadas para PyMEs como ADIIC por su rápida adopción, soporte regional y cumplimiento normativo. Por su parte, los ERP open source (Odoo, ERPNext) ofrecen flexibilidad y control, aunque implican mayores requerimientos técnicos y de mantenimiento. Adicionalmente, alternativas como Shopify POS destacan en estrategias de omnicanalidad, pero dependen de ecosistemas externos. En conjunto, estas evidencias reflejan un panorama donde la elección tecnológica debe balancear costo, facilidad de

implementación y escalabilidad. Para el caso de ADIIC, el desarrollo de un MVP con pila MERN representa una estrategia viable que permite validar la solución en el corto plazo, con la posibilidad de integrarse o evolucionar hacia estas plataformas en escenarios de mayor madurez digital.

Marco Contextual

Bogotá D.C., capital de Colombia, se consolida como el epicentro económico, industrial y empresarial del país. Esta ciudad alberga una alta concentración de pequeñas y medianas empresas (PYMES), desempeñando un papel protagónico en el tejido empresarial nacional gracias a su dinamismo, diversidad sectorial y elevada competitividad (Díaz Pereira, 2020; Rolón Ramírez, 2024). El mercado bogotano se caracteriza por su capacidad de adaptación a las tendencias globales y tecnológicas, así como por el surgimiento de empresas orientadas a satisfacer necesidades específicas de sectores productivos y de servicios.

El sector de dotaciones institucionales, elementos de protección personal (EPP) y calzado de seguridad industrial es estratégico para la economía colombiana, pues suministra insumos esenciales para la protección de los trabajadores en diversos ámbitos laborales (Ortiz Fernández et al., 2024). La importancia de este sector se ha visto potenciada por la implementación y el fortalecimiento del Sistema de Gestión de la Seguridad y Salud en el Trabajo (SG-SST), que exige altos estándares de calidad y la promoción de ambientes laborales seguros.

Adicionalmente, la digitalización, la globalización y la demanda de trazabilidad han generado nuevos desafíos y oportunidades para las empresas dedicadas a este rubro, requiriendo una constante actualización y adaptación a las tendencias tecnológicas y regulatorias vigentes.

ADIIC Dotaciones Institucionales S.A.S. es una empresa familiar, clasificada como PYME, ubicada en Bogotá D.C. Su actividad principal se centra en la comercialización de

dotaciones institucionales, elementos de protección personal (EPP) y calzado de seguridad industrial, acumulando una trayectoria de catorce años en el mercado. Bajo la dirección de Adriana Castañeda y Fabio Uribe, la empresa ha logrado consolidarse gracias a su compromiso con la calidad, el cumplimiento y la satisfacción del cliente (Ortiz Fernández et al., 2024). La estructura organizacional contempla áreas fundamentales como almacén, logística y administración, apoyadas en una infraestructura física y tecnológica acorde a sus necesidades operativas.

Marco Conceptual

A continuación, se definen los principales términos que sustentan el desarrollo del proyecto: estableciendo así un lenguaje común, evitar ambigüedades y comprender la solución propuesta. De esta manera, se integran definiciones clave relacionadas con las metodologías de desarrollo, los modelos de arquitectura de software, las herramientas tecnológicas empleadas y los conceptos propios del dominio del negocio.

La Gestión de Inventarios

La gestión de inventarios es un aspecto fundamental para el éxito de las empresas. Una gestión eficaz de los inventarios permite a las empresas identificar y prevenir la pérdida de materias primas, optimizar los niveles de inventario, mejorar la eficiencia operativa, reducir costos y tomar mejores decisiones.

Navarrete Lozada (2019) destaca algunos de los beneficios de una gestión eficaz de inventarios, como la reducción de costos de almacenamiento, la optimización de los gastos de pedido, la mejora de la eficiencia de la cadena de suministro y la minimización del desperdicio de materias primas.

La gestión de inventarios también ayuda a las empresas a mantener niveles de inventario óptimos para garantizar que tengan suficientes productos para satisfacer la demanda de los clientes sin incurrir en costos de almacenamiento excesivos. Esto se puede lograr mediante la implementación de estrategias como el método FIFO (First In, First Out), que supone que los productos que se compran primero se venden primero.

En general, el autor en su artículo proporciona una visión completa de la importancia de la gestión de inventarios en las empresas, destaca los beneficios de una gestión eficaz de los inventarios y describe los objetivos y estrategias clave para lograr estos beneficios.

Pila MERN (MERN Stack)

La pila MERN (MERN Stack) constituye un entorno de desarrollo full stack basado en JavaScript, integrado por MongoDB, Express.js, React.js y Node.js. Su versatilidad permite abordar una amplia gama de aplicaciones web y móviles, optimizando tanto la capa de presentación como la lógica de negocio y la persistencia de datos.

Entre los usos más frecuentes de esta tecnología se encuentran:

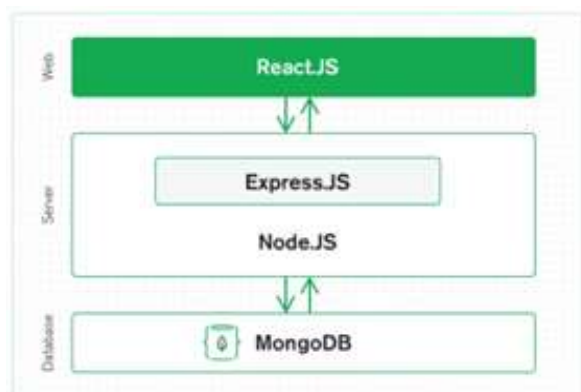
- **Plataformas de comercio electrónico:** aplicaciones de tiendas en línea que ofrecen experiencias de compra interactivas y personalizadas.
- **Redes sociales:** aplicaciones que facilitan la interacción, la comunicación y el intercambio de contenidos entre usuarios.
- **Herramientas de gestión de proyectos:** soluciones que apoyan la colaboración, el control de tareas y la organización de equipos de trabajo.
- **Aplicaciones de streaming:** plataformas para la transmisión de música, vídeos o podcast en tiempo real.

- **Sistemas de gestión de contenido (CMS):** herramientas destinadas a la creación, edición y administración de sitios web de manera eficiente.

En síntesis, la pila MERN (MERN Stack) se ha consolidado como una de las soluciones más relevantes para el desarrollo de aplicaciones web completas en JavaScript. Su popularidad se debe a la facilidad de aprendizaje, la coherencia tecnológica y la amplia disponibilidad de recursos de apoyo, tales como tutoriales, repositorios de código abierto, comunidades activas y entornos de despliegue automatizado.

Figura 1

¿Como funciona la pila MERN?



Fuente. MongoDB. (2024).

Business Process Management (BPM)

Es una metodología que integra principios, técnicas y herramientas orientadas a la mejora continua de los procesos empresariales. Esta disciplina tiene como objetivo alinear los procesos organizacionales con las metas estratégicas, promoviendo la eficiencia, la calidad y la adaptabilidad frente a los cambios del entorno. BPM considera a las organizaciones como sistemas interconectados en los que cada proceso transforma insumos en resultados que generan

valor para el cliente. Su implementación abarca fases clave, como la identificación, el modelado, la ejecución, el monitoreo y la optimización de procesos, apoyadas por tecnologías avanzadas, como la automatización y la inteligencia artificial. Más allá de ser una técnica de gestión, BPM fomenta una visión sistémica y colaborativa dentro de la empresa, destacando la importancia de la satisfacción tanto del cliente interno como del externo. Su adopción estratégica no solo contribuye a la reducción de costos y al incremento de la rentabilidad, sino que también posiciona a las organizaciones en un entorno competitivo y dinámico (Ortiz Fernández et al., 2024).

Patrón Modelo-Vista-Controlador (MVC)

Es una arquitectura de software que organiza las aplicaciones en tres componentes fundamentales: el Modelo, la Vista y el Controlador. El Modelo representa la lógica de negocio y los datos que manipula la aplicación, gestionando su acceso y actualizaciones. La Vista corresponde a la interfaz de usuario, encargada de mostrar información y recibir las interacciones del usuario. Por último, el Controlador actúa como intermediario entre el Modelo y la Vista, procesando las entradas del usuario y actualizando los datos o la presentación según sea necesario. Esta separación permite un desarrollo modular, facilita el mantenimiento del software y promueve la reutilización de componentes. Según Pinzón Núñez et al. (2019), una ventaja clave de este patrón es que los cambios en la interfaz no afectan la lógica de negocio, lo que mejora la escalabilidad y la adaptabilidad del sistema. No obstante, su implementación puede incrementar la complejidad inicial del desarrollo, ya que requiere una estructura bien definida.

Marco Teórico

El presente marco teórico contiene los enfoques, modelos y teorías que respaldan el diseño y desarrollo de la solución propuesta. A través de este apartado se analizan los referentes

teóricos de la ingeniería de software, permitiendo comprender los fundamentos que orientan la implementación del prototipo web en ADIIC Dotaciones Institucionales.

Inventarios

La teoría del inventario se relaciona con la planificación y el control de la cantidad y el flujo de bienes a lo largo de la cadena de suministro de una empresa, incluidos los insumos de los proveedores y los bienes entregados a los consumidores. En cada una de las principales funciones empresariales como las compras, producción, finanzas y distribución, el control de inventarios juega un papel importante y fundamental, aunque genera costos, el control de inventario puede interpretarse como una especie de remedio, un aspecto esencial de una administración eficaz, especialmente cuando la gestión del inventario es un asunto costoso.

Así, el control de inventario contribuye a la salida y entrada de productos de diversa índole en una empresa, por lo que se debe tener un control y registro de los bienes resguardados en el almacén, que es importante, ya que permite conocer la existencia real de los productos con los que cuenta la empresa. La eficiencia del control de almacén, resguardo y conservación permite tener un control para determinar el menor movimiento de la mercancía y poder detectar posibles robos.

Teniendo en cuenta lo anterior, podemos decir que contamos con dos tipos de inventario los cuales son:

Contable, es aquel que se realiza en el sistema, determinando lo que se tiene, lo que entra y lo que sale, por otro lado, tenemos el extracontable, es aquel que se realiza en la empresa por la persona encargada de esta área, el cual debe de realizar un conteo físico de los productos para confirmar los datos que arroja el sistema.

En el inventario extracontable, se deben tener en cuenta dos etapas, una es de carácter

cíclico y el otro es periódico, el levantamiento cíclico, se debe realizar siguiendo un programa calendario el cual es realizado por el jefe de almacén y consiste en contar físicamente lo que se tiene de determinado producto, este conteo está determinado por el sistema, ya que se debe ir a él, para verificar a qué producto se le debe realizar dicho conteo. Después de estar determinado, se debe dirigir a él y realizar el conteo para comparar la información que se encuentra en el sistema, después de realizar dicha comparación y no encontrar inconsistencias, se deben llevar los resultados al jefe de almacén.

El segundo levantamiento es el periódico, llamado físico, en el que participan todos los trabajadores y debe ser supervisado por auditores. El conteo se realizará mínimo dos veces, para corregir o tener seguridad de que la información registrada sea idónea, hay que considerar que mientras se haga este conteo no se permite salida ni entrada de productos de la empresa, en casos extraordinarios de recibir productos, y no se haya dado de alta este ingreso, hay que considerar la factura para incluirlos en el conteo. Debemos considerar que un buen control de inventario permite corregir errores en el sistema y tener la seguridad de que los productos de la empresa están registrados en el sistema. (Arenal Laza, 2020)

Ingeniería de Software.

La Ingeniería de Software es una disciplina que ha revolucionado la forma en que se desarrolla y utiliza el software en la vida cotidiana. Desde simples aplicaciones móviles hasta sistemas empresariales complejos, la Ingeniería de Software ha desempeñado un papel fundamental en la evolución de la tecnología.

Es probable que, la Ingeniería de Software como disciplina formal comenzó a tomar forma en 1968, en una conferencia realizada para discutir lo que entonces se llamaba la “crisis del software” (Robert M. McClure, 2001). Antes de eso, el desarrollo de software se basaba en

enfoques ad hoc y no existían estándares ni métodos establecidos. A medida que los sistemas de software se volvían cada vez más complejos y críticos para las organizaciones, surgió la necesidad de aplicar principios de ingeniería para su desarrollo. Los primeros esfuerzos se centraron en definir procesos y métodos estructurados para desarrollar software de manera más eficiente y confiable.

Es por esta razón que, *“La ingeniería de software es una disciplina de ingeniería que se interesa por todos los aspectos de la producción de software, desde las primeras etapas de la especificación del sistema hasta el mantenimiento del sistema después de que se pone en operación”* (Sommerville, 2020). Implica la aplicación de principios de ingeniería para diseñar, construir, probar y mantener software de calidad. La Ingeniería de Software no solo se trata de escribir código, sino también de comprender las necesidades del cliente, diseñar soluciones efectivas, colaborar en equipo y asegurar la calidad del producto final.

Además de lo anterior, Pressman & Maxim (2021) menciona en su libro a David Hooker, quien propuso siete principios que se centran en la práctica de la ingeniería de software como un todo (pág. 14). Estos principios son los siguientes:

- La razón de que exista todo
- MSE (Mantenlo sencillo, estúpido...)
- Mantener la visión
- Otros consumirán lo que usted produce
- Ábrase al futuro
- Planee por anticipado la reutilización
- ¡Piense!

Estos principios se enfocan en la importancia de agregar valor al sistema, mantener la visión, comunicarse y coordinarse efectivamente, adaptarse a los cambios, centrarse en la calidad del producto final y tener en cuenta las necesidades de los usuarios finales.

Por todo lo anterior, La Ingeniería de Software trae beneficios para los desarrolladores y los usuarios finales. Algunos de los beneficios más destacados incluyen:

- **Eficiencia:** La aplicación de metodologías y procesos estructurados permite un desarrollo de software más eficiente, reduciendo el tiempo y los recursos necesarios.
- **Calidad:** La Ingeniería de Software se enfoca en garantizar la calidad del software mediante pruebas rigurosas y estándares de desarrollo.
- **Mantenibilidad:** Los principios de ingeniería aplicados en el desarrollo de software facilitan su mantenimiento y actualización a largo plazo.
- **Mayor satisfacción del cliente:** Al seguir procesos bien definidos y garantizar la calidad, la Ingeniería de Software mejora la satisfacción del cliente al entregar productos confiables y funcionales.
- **Innovación continua:** La Ingeniería de Software fomenta la innovación al proporcionar un marco estructurado para la creación de nuevas soluciones tecnológicas.

Para finalizar, la Ingeniería de Software ha permitido el rápido avance de la tecnología y ha revolucionado la forma de interactuar con el mundo digital. A través de la aplicación de principios de ingeniería y la adopción de procesos estructurados, la Ingeniería de Software ha mejorado la calidad, eficiencia y mantenibilidad del software. Desde sus inicios hasta hoy, la disciplina sigue evolucionando y es fundamental en el desarrollo de soluciones tecnológicas innovadoras. En resumen, la Ingeniería de Software es una disciplina clave que impulsa el progreso y la transformación en el mundo digital.

Paradigmas de Programación

Los paradigmas de programación han evolucionado a lo largo del tiempo para abordar diferentes enfoques y necesidades en el desarrollo de software. Desde sus inicios hasta hoy, se han introducido enfoques y paradigmas para abordar los desafíos de desarrollo de software de manera más eficiente y efectiva. Estos modelos de programación son respuestas a las necesidades cambiantes de la industria y a los avances tecnológicos, e influyeron en su forma de desarrollar y diseñar software actualmente. (Joyanes Aguilar, 2020)

Los dos principales paradigmas de programación tenemos: el paradigma imperativo y el declarativo. En el paradigma imperativo se describe paso a paso un conjunto de instrucciones que deben ejecutarse para variar el estado del programa y hallar la solución y en el paradigma declarativo se programa diciendo lo que se quiere resolver a nivel de usuario, pero no las instrucciones necesarias para solucionarlo. La programación imperativa se divide en tres amplias categorías: procedural, orientada a objetos y orientada a eventos y la programación declarativa se divide en lógica, funcional, base de datos y heurístico. (Timarán Pereira et al., 2009)

Programación Orientada a Objetos – POO, Joyanes Aguilar (2020), en su libro indica que la Programación Orientada a Objetos (POO) es un paradigma de programación que se basa en el concepto de "objetos". Es una de las metodologías más utilizadas en el desarrollo de software por su capacidad para modelar y simular situaciones del mundo real eficientemente estructurada.

Por lo tanto, la POO es un estilo de programación que se centra en la creación de "objetos", que son instancias de una clase. Una clase es una plantilla o un plano que define las propiedades y comportamientos que un objeto puede tener. Los objetos interactúan entre sí a través de métodos, que son funciones asociadas a una clase.

En la POO, los objetos se consideran como entidades independientes que encapsulan tanto los datos como los métodos asociados a ellos. Esto permite una mayor modularidad y reutilización de código, ya que los objetos pueden ser utilizados en diferentes partes de un programa sin necesidad de volver a escribir el mismo código una y otra vez.

En ese mismo contexto la POO se rige por cuatro principios fundamentales, conocidos como los "cuatro pilares" de la POO:

Abstracción: Consiste en identificar las características esenciales de un objeto y omitir los detalles irrelevantes. Permite crear clases que representan conceptos abstractos y complejos, lo que facilita la comprensión y el diseño del software.

Encapsulación: El proceso de ocultar los detalles internos de un objeto y proporcionar una interfaz para interactuar con él. Esto se logra mediante el uso de modificadores de acceso, como públicos, privados y protegidos, que controlan la visibilidad de los atributos y métodos de una clase.

Herencia: Permite crear nuevas clases basadas en clases existentes, heredando sus propiedades y comportamientos. Esto promueve la reutilización de código y facilita la organización jerárquica de las clases.

Polimorfismo: Permite que un objeto pueda tomar diferentes formas y comportarse de diferentes maneras según el contexto. Esto se logra usando herencia y la capacidad de sustituir una clase base por una clase derivada.

En paralelo, la POO tiene numerosos beneficios en el desarrollo de software:

Reutilización de código: Es uno de los principales beneficios de la POO. Al encapsular la lógica en objetos, se pueden usar en diferentes partes de un programa sin volver a escribir el mismo código una y otra vez.

Modularidad: Es otra ventaja importante de la POO. Al dividir un programa en objetos independientes, se puede desarrollar, probar y mantener cada componente de manera individual, lo que facilita el mantenimiento y la escalabilidad del software.

Flexibilidad: La POO permite una mayor flexibilidad en el diseño y desarrollo de software. Los objetos pueden ser modificados o reemplazados fácilmente sin afectar al resto del sistema, lo que facilita la adaptación a los cambios y requisitos del negocio.

Facilidad de mantenimiento: Debido a la modularidad y reutilización de código, el mantenimiento de software basado en POO es más sencillo. Los errores pueden ser localizados y corregidos más rápidamente, lo que reduce el tiempo y los costos asociados.

Finalmente, la programación orientada a objetos es un paradigma de programación poderoso y ampliamente utilizado en el desarrollo de software. Permite un mayor modularidad, reutilización de código y mayor flexibilidad en el diseño y desarrollo de aplicaciones. Al comprender y aplicar los conceptos de programación orientada a objetos, los desarrolladores pueden crear software más estructurado, fácil de usar y escalable.

Desarrollo Orientado a la Web (DOW)

El Desarrollo Orientado a la Web (DOW), también conocido como Desarrollo Web, constituye un campo fundamental en la informática actual, representando una disciplina esencial para la creación y el mantenimiento de sitios web y aplicaciones web (Pressman & Maxim, 2021). Este proceso abarca diversas etapas, desde la planificación y el diseño conceptual hasta la implementación y el mantenimiento posterior. Se basa en la utilización de distintos lenguajes de programación, herramientas y tecnologías, permitiendo la construcción de experiencias web interactivas y funcionales (Larman, 2003). En el ámbito del DOW, diversas metodologías guían el proceso de desarrollo, asegurando la calidad y eficiencia del proyecto. Entre las más utilizadas

se encuentran la Metodología en Cascada, caracterizada por su enfoque lineal y secuencial (Pressman & Maxim, 2021); las Metodologías Ágiles como Scrum o Kanban, que promueven un enfoque iterativo y flexible (Larman, 2003); y la Metodología DevOps, que integra las prácticas de desarrollo y operaciones, promoviendo una colaboración estrecha entre ambos equipos para mejorar la entrega continua de software (Kim et al., 2016). El DOW involucra el uso de diversas tecnologías, que se pueden agrupar en las siguientes categorías:

- Lenguajes de programación: Lenguajes de marcado (HTML5, XML), Lenguajes de scripting (JavaScript, TypeScript), Lenguajes de hojas de estilo (CSS3).
- Bases de datos (MySQL, PostgreSQL, MongoDB).
- Frameworks de desarrollo (React, Angular, Vue.js, Node.js).
- Librerías y herramientas (jQuery, Bootstrap, Git, Docker).

En la actualidad, el DOW resulta esencial en el contexto de la amplia presencia de internet y la necesidad de contar con sitios web y aplicaciones web funcionales, accesibles y seguras. Estas herramientas permiten la creación de plataformas digitales para diversos fines, como:

- Comercio electrónico: Permiten la venta de productos y servicios en línea de manera segura y eficiente.
- Educación: Facilitan el acceso a la educación a través de plataformas de aprendizaje en línea, cursos virtuales y recursos educativos digitales.
- Entretenimiento: Ofrecen diversas opciones de entretenimiento, como juegos en línea, videos, música y streaming de contenido.
- Gobierno electrónico: Permiten a los ciudadanos acceder a servicios públicos en línea, simplificando trámites y mejorando la eficiencia de la administración pública.

El DOW también desempeña un papel crucial en la transformación digital de las empresas, permitiéndoles:

- Mejorar sus procesos: Automatizar tareas, optimizar flujos de trabajo y reducir costos operativos.
- Aumentar su productividad: Facilitar la colaboración entre equipos, mejorar la comunicación interna y externa, y agilizar la toma de decisiones.
- Competir en el mercado global: Llegar a nuevos clientes, expandir su mercado objetivo y ofrecer una experiencia de usuario superior.

El Desarrollo Orientado a la Web (DOW) se ha convertido en una disciplina esencial en la era digital, impulsando el desarrollo de sitios web y aplicaciones web que transforman la forma en que vivimos, trabajamos y nos comunicamos. Su relevancia se intensifica en el contexto de la transformación digital, permitiendo a las empresas y organizaciones adaptarse a las demandas del mercado global y mejorar su competitividad.

Marco Legal

Este apartado sintetiza el marco normativo aplicable al proyecto y a la operación de ADIIC Dotaciones Institucionales, con énfasis en: seguridad y salud en el trabajo, dotaciones, control de inventarios, facturación electrónica y protección de datos personales. Su objetivo es dar soporte regulatorio a los procesos implementados y a los controles de seguridad descritos en la arquitectura.

Sistema de Gestión de Seguridad y Salud en el Trabajo

El Sistema de Gestión de Seguridad y Salud en el Trabajo (SG-SST) es un conjunto de normas y procedimientos establecidos por el gobierno colombiano con el objetivo de garantizar la protección de los trabajadores y prevenir accidentes laborales y enfermedades ocupacionales.

El SG-SST, implementado en 2012, se ha convertido en una herramienta fundamental para promover ambientes laborales seguros y saludables en el país.

Por lo tanto, el SG-SST tiene sus raíces en la Ley 9 de 1979, la cual estableció las bases para la promoción de la seguridad y salud en el trabajo (Congreso de la República de Colombia, 1979). Sin embargo, fue a partir de la Ley 1562 de 2012 que se promulgó el SG-SST como un sistema integral y obligatorio para todas las empresas y organizaciones del país. Esta ley estableció la responsabilidad de los empleadores en la implementación y mantenimiento de un SG-SST efectivo. (Congreso de la República de Colombia, 2012a)

Por consiguiente, el SG-SST es un modelo de gestión que busca identificar, evaluar y controlar los riesgos laborales presentes en los diferentes entornos de trabajo. Esto implica la adopción de medidas preventivas, la capacitación de los trabajadores, la promoción de una cultura de seguridad y salud, y la implementación de programas de vigilancia epidemiológica y control de enfermedades ocupacionales.

No obstante, el SG-SST se basa en una serie de principios fundamentales que guían su implementación y funcionamiento. Estos principios incluyen:

1. Liderazgo y compromiso de la alta dirección: Los líderes de las organizaciones deben asumir un rol activo en la promoción de la seguridad y salud en el trabajo.
2. Participación de los trabajadores: Los trabajadores deben ser involucrados en la identificación y control de los riesgos laborales, así como en la toma de decisiones relacionadas con su seguridad y salud.
3. Identificación y evaluación de riesgos: Es necesario realizar una evaluación exhaustiva de los riesgos laborales presentes en cada lugar de trabajo, con el fin de implementar medidas preventivas adecuadas.

4. Prevención y control de riesgos: Se deben implementar medidas de control para eliminar o reducir los riesgos laborales identificados.
5. Mejora continua: El SG-SST es un proceso dinámico que requiere de una constante revisión y mejora de sus componentes.

Es decir, la implementación efectiva del SG-SST en las empresas y organizaciones colombianas trae beneficios para empleadores y trabajadores. Algunos de estos beneficios incluyen:

- Reducción de accidentes y enfermedades laborales: El SG-SST contribuye a minimizar los riesgos laborales, lo que se traduce en una disminución de los accidentes y enfermedades ocupacionales.
- Mejora del clima laboral: La implementación de medidas de seguridad y salud en el trabajo promueve un ambiente laboral más seguro, saludable y agradable para los trabajadores.
- Cumplimiento legal: El SG-SST ayuda a las empresas a cumplir con la legislación vigente en materia de seguridad y salud en el trabajo, evitando sanciones y multas.

Teniendo en cuenta lo anterior, el SG-SST, posee los siguientes antecedentes normativos:

- Ley 9 de 1979 Establece los lineamientos generales para la promoción de la seguridad y salud en el trabajo en Colombia. (Congreso de la República de Colombia, 1979)
- Resolución 1016 de 1989 Por la cual se reglamenta la organización, funcionamiento y forma de los Programas de SO que deben desarrollar los patronos o empleadores en el país. (Ministerio de Trabajo y Seguridad Social & Ministerio de Salud, 1989)
- Ley 1562 de 2012 Promulga el SG-SST como un sistema integral y obligatorio para todas las empresas y organizaciones del país. (Congreso de la República de Colombia, 2012a)
- Decreto 1072 de 2015 por medio del cual se expide el decreto único reglamentario del sector

trabajo. (Presidencia de la República de Colombia, 2015b)

- Decreto 52 de 2017 Por medio del cual se modifica el artículo 2.2.4.6.37. del Decreto 1072 de 2015 Decreto Único Reglamentario del Sector Trabajo. (Presidencia de la República de Colombia, 2017)
- Resolución 1111 de 2017 Establece los lineamientos técnicos para la implementación del SG-SST en Colombia. (Colombia. Ministerio del Trabajo, 2017)

Por consiguiente, el Sistema de Gestión de Seguridad y Salud en el Trabajo (SG-SST) es una herramienta fundamental para promover ambientes laborales seguros y saludables en Colombia. Su implementación efectiva ayuda a reducir accidentes y enfermedades laborales, mejora el clima laboral y garantiza el cumplimiento de la legislación vigente. Es responsabilidad de los empleadores adoptar y mantener un SG-SST efectivo para proteger la integridad y bienestar de los trabajadores.

Dotaciones Institucionales.

Las dotaciones a los empleados en Colombia son un aspecto importante en el ámbito laboral. Estas dotaciones se refieren a los beneficios y suministros que una empresa brinda a sus empleados para garantizar su bienestar y comodidad en el lugar de trabajo.

Los inicios de las dotaciones a los empleados en Colombia se remontan a la década de 1920, cuando surgieron las primeras legislaciones laborales para proteger los derechos de los trabajadores. En ese momento, se estableció la obligación de las empresas de proveer uniformes, herramientas y equipos necesarios para realizar el trabajo. Con el tiempo, estas dotaciones se expandieron para incluir otros elementos como calzado de seguridad, implementos de protección personal y suministros de oficina.

Los principios fundamentales de las dotaciones a los empleados en Colombia se basan en la equidad y la seguridad laboral. Es importante que todas las empresas brinden a sus empleados las mismas condiciones y oportunidades en términos de dotación. Esto garantiza un ambiente de trabajo justo y evita posibles conflictos o discriminaciones entre los empleados.

Los beneficios de las dotaciones a los empleados son tanto para la empresa como para los trabajadores. Para la empresa, estas dotaciones contribuyen a mejorar la imagen corporativa, fortalecer la identidad de la organización y aumentar la productividad de los empleados. Al contar con las herramientas y suministros adecuados, los empleados pueden desempeñar su trabajo de manera más eficiente y segura. Para los empleados, las dotaciones representan una mejora en su calidad de vida laboral. Al recibir uniformes, calzado de seguridad y otros elementos necesarios, se sienten valorados y respaldados por la empresa. Además, estas dotaciones promueven la seguridad y previenen accidentes laborales, lo que se traduce en un ambiente de trabajo más seguro y saludable.

En Colombia, las dotaciones a los empleados están reguladas por una serie de leyes, decretos y resoluciones. A continuación, se presenta un listado con algunas de las normativas relevantes:

Resolución 2400 de 1979: Esta resolución establece las normas de seguridad e higiene en el trabajo, y contempla la obligación de proporcionar dotaciones de acuerdo con las condiciones laborales y riesgos a los que están expuestos los empleados. (Colombia. Ministerio de Trabajo y Seguridad Social, 1979)

Ley 50 de 1990: Esta ley introduce reformas al Código Sustantivo del Trabajo y se dictan otras disposiciones, en su artículo 78 obligando a las empresas de servicios temporales asumir su responsabilidad con el trabajador cuando este preste oficios o actividades

particularmente riesgosas, adiestrándolo de forma particular en cuanto a prevención de riesgos, o sea necesario el suministro de elementos de protección especial. (Congreso de la República de Colombia, 1990)

Ley 789 de 2002: Esta ley regula el salario mínimo en Colombia, el cual es un factor importante para determinar el valor de las dotaciones que deben ser entregadas a los empleados. (Congreso de la República de Colombia, 2002)

Decreto 1072 de 2015: Este decreto regula el Sistema de Gestión de la Seguridad y Salud en el Trabajo (SG-SST), el cual incluye disposiciones para la entrega de elementos de protección personal a los empleados. (Presidencia de la República de Colombia, 2015b)

Es importante tener en cuenta que existen otras normativas y disposiciones que pueden aplicarse en casos específicos, dependiendo del sector laboral y las condiciones de trabajo.

En conclusión, las dotaciones a los empleados en Colombia son fundamentales para garantizar el bienestar y la seguridad laboral. Estas dotaciones han evolucionado a lo largo del tiempo, pasando de ser simplemente uniformes y herramientas para abarcar una amplia gama de suministros y beneficios. Tanto para la empresa como para los empleados, las dotaciones representan una inversión que contribuye a la eficiencia, la seguridad y el bienestar en el lugar de trabajo. Es importante que las empresas cumplan con la normatividad establecida y brinden a sus empleados las dotaciones adecuadas para fomentar un ambiente laboral justo y seguro.

Control de Inventarios

El control de inventarios en Colombia está compuesto por diversas normas, entre las más importantes se encuentran:

- Ley 1314 de 2009: Esta ley establece los principios y normas generales de contabilidad financiera y de información financiera, y ordena la convergencia con las Normas Internacionales de Contabilidad. (Congreso de la República de Colombia, 2009)
- Decreto 2649 de 1993: Este decreto reglamenta la Ley 1314 de 2009 y establece los requisitos específicos para la valoración y presentación de los inventarios en los estados financieros. (Presidencia de la República de Colombia, 1993)
- Decreto 302 de 2015: Este decreto, reglamentario de la Ley 1314, establece en su Artículo 10 que "los auditores independientes deberán observar las Normas Internacionales de Auditoría al realizar auditorías de estados financieros". Complementa y desarrolla los principios establecidos en la Ley 1314, y su Artículo 10 reitera la obligatoriedad de aplicar las NIA para los auditores. (El Presidente de La República de Colombia, 2015)
 - Código de Comercio: El Código de Comercio contiene algunas disposiciones relacionadas con el control de inventarios, como las normas sobre la compraventa de mercancías y la prenda mercantil.

Facturación Electrónica

La legislación colombiana que regula la facturación electrónica se fundamenta en varios decretos y resoluciones que establecen su obligatoriedad y los requisitos que deben cumplir las facturas emitidas. El Decreto 2242 de 2015 constituyó el primer marco normativo que instituyó el modelo de facturación electrónica, definiendo las condiciones para su expedición y promoviendo su uso como herramienta para mejorar la interoperabilidad y el control fiscal en el país (Presidencia de la República de Colombia, 2015c). Posteriormente, la Resolución 000042 de 2020 actualizó los plazos y las condiciones para su implementación, estableciendo que todos los contribuyentes debían registrarse como facturadores electrónicos y comenzar a expedir facturas

electrónicas a partir del 1 de noviembre de 2020 (Colombia. Unidad Administrativa Especial Dirección de Impuestos y Aduanas Nacionales (DIAN), 2020). Además, el artículo 617 del Decreto Ley 624 de 1989 (Estatuto Tributario) detalla los elementos que debe contener una factura electrónica, entre ellos la descripción de los bienes o servicios, el valor total, la forma de pago y la firma digital (Presidencia de la República de Colombia, 1989). Esta normativa no solo busca facilitar los procesos administrativos de las empresas, sino también combatir la evasión fiscal mediante un registro más eficiente de las transacciones comerciales.

Protección de datos personales

El tratamiento de datos personales en el sistema de ADIIC se rige por la Ley 1581 de 2012 y su reglamentación (Decreto 1377 de 2013), además de las obligaciones asociadas al Registro Nacional de Bases de Datos (RNBD) ante la Superintendencia de Industria y Comercio, cuando aplique (Congreso de la República de Colombia, 2012b; Presidencia de la República de Colombia, 2013).

Principios: legalidad, finalidad, libertad, veracidad/calidad, transparencia, acceso y circulación restringida, seguridad y confidencialidad.

Derechos del titular: conocer, actualizar, rectificar, suprimir y revocar autorización; atención de consultas y reclamos dentro de los plazos legales.

Otras Normas

- Normas técnicas colombianas: Las normas técnicas colombianas (NTC) son elaboradas por el Instituto Colombiano de Normas y Certificación (ICONTEC) y establecen los requisitos mínimos de calidad para los productos y procesos. Algunas NTC pueden ser relevantes para el control de inventarios, como las NTC relacionadas con el almacenamiento y la manipulación de mercancías.

Además de las normas mencionadas anteriormente, existen otras disposiciones legales que pueden ser relevantes para el control de inventarios en Colombia, dependiendo del sector de actividad de la empresa. Por ejemplo, las empresas que se dedican a la venta de alimentos y bebidas deben cumplir con las normas sanitarias establecidas por el Instituto Nacional de Vigilancia Sanitaria (INVIMA).

El control de inventarios es un proceso fundamental para las empresas, ya que les permite optimizar sus recursos, reducir costos y mejorar su eficiencia operativa. Un buen control de inventarios debe basarse en un marco legal sólido y en prácticas adecuadas de gestión.

Artefactos de cumplimiento exigidos por el proyecto:

- Política de Tratamiento de Datos Personales (pública, versionada) y Aviso de Privacidad con mecanismo de consentimiento informado cuando la base jurídica no sea contractual u obligatoria.
- Procedimiento de consultas y reclamos (canales, plazos, responsable).
- Matriz de datos personales por proceso/base (finalidad, base jurídica, conservación, ubicación/encargado).
- Acuerdos de encargo de tratamiento con terceros (hosting, correo, proveedor de facturación, etc.).
- Verificación y registro/actualización en el RNBD, si corresponde.

Metodología

Tipo de Estudio

El proyecto correspondió a un estudio aplicado y descriptivo, orientado a resolver una problemática real detectada en la gestión de inventarios y venta de la empresa ADIIC Dotaciones Institucionales S.A.S. El enfoque aplicado se justificó porque el objetivo central fue implementar un prototipo funcional orientado a la web mediante el uso de la pila MERN (MERN stack), mientras que la naturaleza descriptiva radicó en la caracterización detallada de los procesos actuales y en la documentación de las fases de análisis, diseño, desarrollo e implementación del sistema (Vásquez Ramírez et al., 2023).

Enfoque Metodológico

Se adoptó un enfoque mixto, integrando métodos cualitativos y cuantitativos para la recolección y validación de información durante las diferentes fases del proyecto. Inicialmente, se emplearon técnicas cualitativas, como entrevistas semiestructuradas y observación directa, para identificar requerimientos y analizar los procesos empresariales existentes. Posteriormente, se incorporaron técnicas cuantitativas, específicamente encuestas de satisfacción y pruebas de usabilidad, para evaluar de manera objetiva la experiencia de los usuarios y la eficacia de la solución tecnológica implementada. Este enfoque mixto permitió obtener una comprensión profunda del fenómeno estudiado y generó inferencias integradas que incrementaron la validez y la robustez de los resultados (Storey et al., 2025).

Técnicas e Instrumentos de Recolección de Información

Durante la fase de diagnóstico y análisis, se utilizaron entrevistas semiestructuradas (Anexos 1, 2, 3 y 4) a actores clave del proceso, observación directa de procedimientos operativos y análisis documental de registros e informes internos de la organización. En la etapa

de validación del prototipo, se aplicaron pruebas de usabilidad y encuestas tipo System Usability Scale (SUS) a usuarios finales, lo que permitió obtener retroalimentación objetiva sobre funcionalidad, facilidad de uso y pertinencia de la solución propuesta. La selección de estas técnicas respondió a la necesidad de comprender tanto la perspectiva operativa de los usuarios como la eficiencia técnica del sistema.

Metodologías Implementadas

Para el desarrollo del prototipo web, se implementó una estrategia metodológica dual que combinó la Gestión por Procesos de Negocio (Business Process Management, BPM) para el análisis y la optimización de flujos de trabajo, y la metodología ágil Scrum para el desarrollo de software.

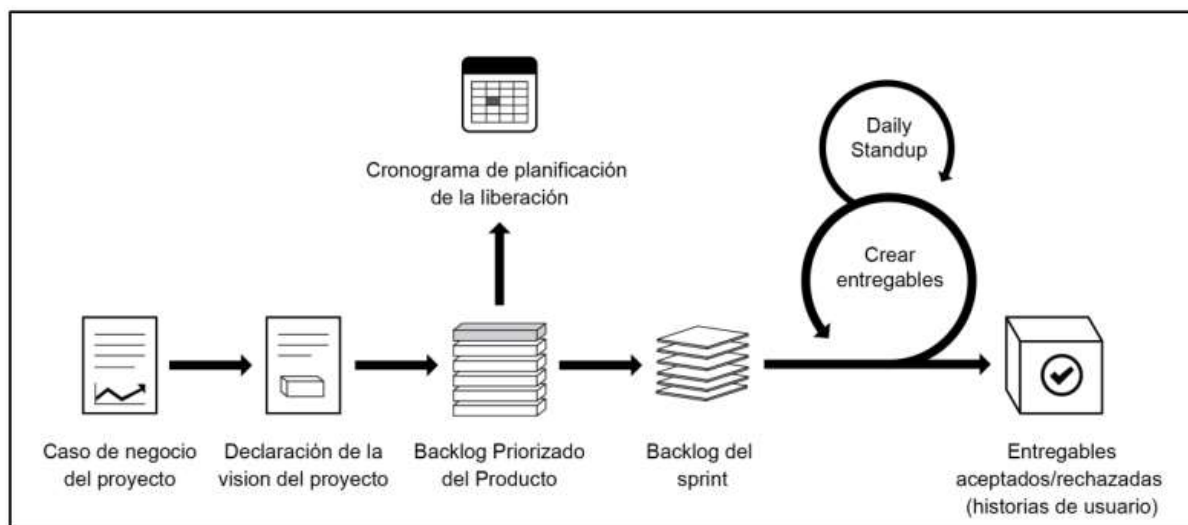
Gestión por Procesos de Negocio (Business Process Management - BPM): La metodología BPM, estructurada en cinco fases (diseño, modelado, ejecución, monitoreo y optimización), fue aplicada parcialmente. En particular, se abordaron las fases de diseño y modelado. Se realizó un análisis detallado de los procesos de adquisición y venta empleando la notación BPMN; como resultado, se modelaron los procesos actuales (AS-IS), lo cual permitió identificar ineficiencias y puntos críticos del flujo operativo. A partir de estos hallazgos, se diseñaron procesos optimizados (TO-BE) que sirvieron como base para el levantamiento de requisitos funcionales y técnicos (Ortiz Fernández et al., 2024).

Scrum: Posterior a la definición de procesos optimizados, se implementó la metodología ágil Scrum para el desarrollo iterativo e incremental del sistema de gestión de inventarios. El trabajo se estructuró en sprints de dos a cuatro semanas, con entregas funcionales al final de cada ciclo, planificación colaborativa, retrospectivas y reuniones diarias que garantizaron transparencia, adaptación continua y entrega de valor (Satpathy, 2022). Como se ilustra en la

Figura 2, este flujo de trabajo caracterizó la dinámica central en cada sprint.

Figura 2

Flujo de Scrum para un sprint



Fuente. SCRUMstudy. (2022, p. 2)

Asimismo, el método Scrum se estructuró en torno al Sprint, núcleo esencial del proceso, que permitió realizar entregas periódicas mediante un enfoque iterativo e incremental. De acuerdo con SCRUMstudy (2022), las etapas fundamentales incluyeron:

- **Planificación del Sprint (Sprint Planning):** al inicio de cada sprint, se definieron objetivos, se seleccionaron elementos del Product Backlog y se estableció un Sprint Backlog detallado.
- **Desarrollo del Sprint (Sprint Execution):** el equipo ejecutó tareas acordadas, con reuniones diarias (Daily Scrum) para coordinar avances, detectar obstáculos y definir acciones.
- **Evaluación del Sprint (Sprint Review):** al término de cada sprint, el equipo expuso resultados a clientes e interesados, y recogió observaciones para mejoras.
- **Análisis retrospectivo del Sprint (Sprint Retrospective):** se revisó el proceso, se identificaron

oportunidades de mejora operativa y colaborativa y se plantearon acciones para iteraciones siguientes.

- Ajuste y actualización del Backlog (Backlog Refinement): se realizó un proceso continuo de revisión y preparación del Product Backlog.

Esta combinación de BPM y Scrum proporcionó al proyecto una visión integrada y flexible, garantizó soluciones técnicas ágiles alineadas con el análisis de procesos y favoreció una generación sostenida de valor.

Fases del Proyecto

En consonancia con la integración metodológica, el desarrollo del proyecto se estructuró en las siguientes fases:

Análisis de Procesos de Inventario y Ventas con Enfoque BPM

Se aplicó BPM para levantar información mediante entrevistas, observación directa y análisis documental. Se modelaron los procesos actuales en diagramas AS-IS, con identificación de puntos críticos y oportunidades de mejora.

Diseño de la Arquitectura del Sistema de Gestión de Inventario y Ventas

A partir del modelado TO-BE, se propusieron y validaron flujos optimizados que sirvieron como base para definir la arquitectura general del sistema. Esta se estructuró bajo un enfoque cliente-servidor de tres capas con la pila MERN (MERN Stack) y el patrón MVC, lo que garantizó modularidad, escalabilidad y mantenibilidad. Se diseñó un modelo desacoplado entre frontend y backend, comunicados mediante APIs RESTful para facilitar integraciones futuras y el trabajo colaborativo (Teja Labba et al., 2023). De manera complementaria, se adoptó un enfoque multivista (lógica, desarrollo, procesos y física) orientado a distintos perfiles técnicos e interesados.

Desarrollo Iterativo del Sistema de Inventario y Ventas

El prototipo fue desarrollado e implementado de manera incremental bajo Scrum. A través de entregas iterativas en cada sprint, se incorporaron mejoras y retroalimentación, asegurando la alineación de las funcionalidades con las necesidades de la empresa.

Evaluación de Usabilidad del Prototipo del Sistema de Inventario y Ventas

Se llevaron a cabo pruebas de usabilidad para evaluar la interacción de los usuarios con el sistema. Los participantes realizaron tareas concretas con el prototipo en un entorno controlado y, posteriormente, respondieron encuestas SUS. Esto permitió recopilar observaciones sobre facilidad de uso, desempeño, utilidad y pertinencia, y se realizaron ajustes en funcionalidades conforme a la retroalimentación obtenida.

Análisis de los Procesos de Adquisición y Venta

En este apartado se presenta el análisis de la gestión de inventarios en ADIIC Dotaciones Institucionales, con énfasis en los procesos de adquisición y venta. Este análisis se realizó mediante entrevistas individuales con actores clave de la empresa, lo que permitió obtener una visión integral de la operación actual. A partir de esta información, se aplicó la metodología de Gestión por Procesos de Negocio (Business Process Management – BPM), mediante la cual se mapearon los flujos actuales (AS-IS) y se propusieron mejoras orientadas a su optimización (TO-BE).

Como resultado de este análisis, y en coherencia con el enfoque ágil adoptado en el proyecto, se definieron el Producto Mínimo Viable (MVP), las épicas funcionales y las historias de usuario que guían el desarrollo del sistema. De esta manera, los requisitos no se modelaron bajo enfoques tradicionales, sino a través de artefactos propios de Scrum que permiten una planeación iterativa y centrada en el valor para el usuario.

Plan de Obtención de Requisitos

Este plan se enmarca en la metodología ágil Scrum y tiene como objetivo principal establecer una base sólida para el desarrollo del prototipo, asegurando que cumpla con los objetivos, el alcance y las expectativas de las partes interesadas. Para ello, se definieron técnicas de levantamiento de información orientadas a identificar y documentar los requerimientos funcionales y no funcionales del sistema. En este caso, se emplearon las siguientes técnicas de obtención de información:

Entrevistas Individuales

Se llevaron a cabo entrevistas individuales con tres actores clave de la empresa ADIIC Dotaciones Institucionales: la gerente general, el gerente operativo y la vendedora. El propósito

de estas entrevistas fue comprender sus expectativas, identificar necesidades operativas y recoger criterios clave para el diseño del sistema. Para ello, se utilizaron preguntas abiertas, acompañadas de técnicas de escucha activa y reformulación, con el fin de garantizar la captura precisa de los requerimientos. La información recolectada también sirvió como insumo para el análisis de procesos y la construcción de los diagramas AS-IS y TO-BE descritos en el capítulo correspondiente.

Identificación de Roles en ADIIC Dotaciones Institucionales

La identificación de roles permitió definir las responsabilidades y áreas de trabajo de cada participante en el proyecto. Esta definición es clave para entender la relación entre los actores empresariales y el equipo de desarrollo. En este proyecto, se identificaron los siguientes roles, descritos en la tabla 1.

Tabla 3

Asignación de roles

Nombres y apellidos	Cargo	Responsabilidad
Adriana Castañeda Calvo	Gerente General de “ADIIC Dotaciones Institucionales”	<ul style="list-style-type: none"> – Ostenta la máxima responsabilidad sobre el proyecto en su adquisición. – Aporta su visión estratégica y liderazgo al proyecto.
Fabio Uribe Guerrero	Gerente Operativo de “ADIIC Dotaciones Institucionales”	<ul style="list-style-type: none"> – Supervisa y ejecuta las operaciones diarias de la empresa. – Es clave en la obtención de los requerimientos del proyecto.
Sara Isabela Zalasar Castañeda	Vendedora de “ADIIC Dotaciones Institucionales”	<ul style="list-style-type: none"> – Es clave en la obtención de los requerimientos del proyecto.
Juan Carlos Arrieta Bustos	Product Owner y Miembro del Equipo de Desarrollo	<ul style="list-style-type: none"> – Comprende las necesidades de los usuarios y define los requisitos del producto. – Aporta sus conocimientos técnicos en ingeniería de sistemas para la implementación del producto.

Richard Pardo Cardona	Scrum Master y Miembro del Equipo de Desarrollo	<ul style="list-style-type: none"> – Facilita las reuniones y eventos Scrum. – Promueve la comunicación y colaboración entre los miembros del equipo. – Vela por el cumplimiento de las metodologías ágiles. – Aporta su experiencia en ingeniería de sistemas para la implementación del producto.
Proveedor	Representante de la empresa proveedora	<ul style="list-style-type: none"> – Suministra productos y materiales necesarios para el inventario. – Asegura la calidad y disponibilidad de los insumos. – Mantiene una comunicación fluida con el equipo de operaciones para garantizar entregas oportunas.

Nota. La tabla presenta la asignación de roles de los participantes involucrados en el desarrollo del proyecto para la empresa “ADIIC Dotaciones Institucionales”.

Descripción del Proceso de Adquisición y Ventas

El proceso de adquisición está basado en el modelo Just-in-Time (JIT) (Ladrón de Guevara, 2020) y gestionado manualmente, se organiza en fases que buscan optimizar la eficiencia y reducir los tiempos de almacenamiento:

- Identificación de productos y rotación en el inventario, se realiza un análisis detallado para identificar los productos disponibles en el inventario, verificando su rotación con base en las necesidades de la empresa.
- Monitoreo manual de niveles de inventario por producto, se lleva a cabo un control manual para verificar los niveles de inventario, asegurando que no existan desbalances y anticipando posibles necesidades de reposición.
- Alerta de reabastecimiento al alcanzar niveles mínimos, cuando los niveles de inventario llegan al punto mínimo establecido, se genera una alerta que indica la necesidad de realizar un reabastecimiento.

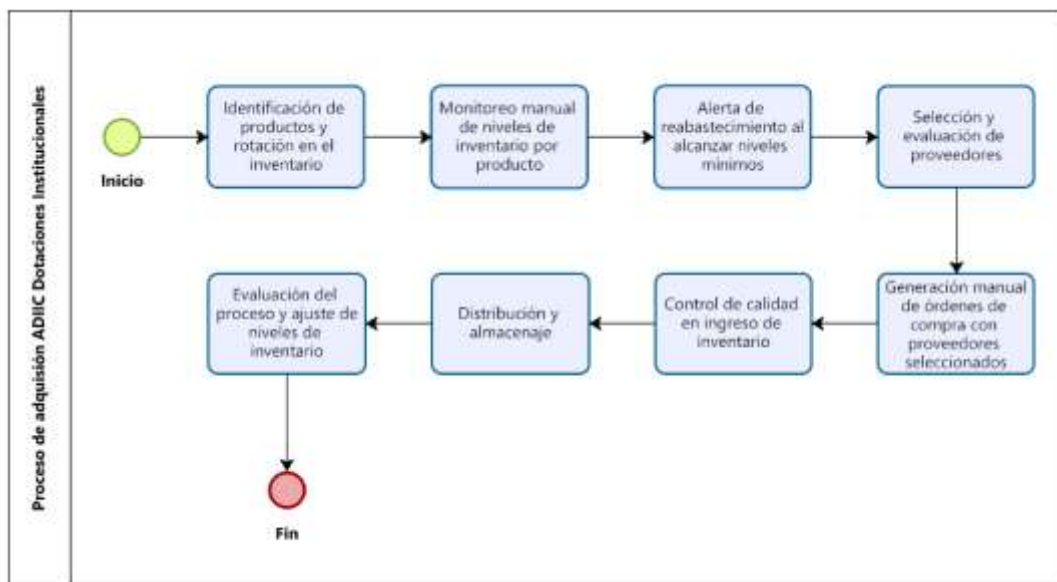
- Selección y evaluación de proveedores, se eligen y evalúan los proveedores que cumplan con los requisitos de calidad, tiempo de entrega y precio, garantizando la mejor opción para el almacén.
- Generación manual de órdenes de compra con proveedores seleccionados, una vez seleccionados los proveedores, se emiten manualmente las órdenes de compra necesarias para reabastecer el inventario.
- Control de calidad en ingreso de inventario, al recibir los productos adquiridos, se realiza un control de calidad para verificar que cumplan con las especificaciones requeridas.
- Distribución y almacenaje, los productos aprobados durante el control de calidad son distribuidos y almacenados de manera organizada en el inventario, listos para ser utilizados.
- Evaluación del proceso y ajuste de niveles de inventario, finalmente, se evalúa el proceso completo para identificar oportunidades de mejora y se ajustan los niveles de inventario según las tendencias de consumo y necesidades futuras.

Este procedimiento busca mantener un flujo eficiente en la gestión del inventario y garantizar la disponibilidad de productos esenciales para la operación de la empresa.

La secuencia de estas fases se ilustra en la figura 3.

Figura 3

Diagrama de flujo proceso de adquisiciones.



Fuente. Autoría Propia.

El proceso de venta consiste en:

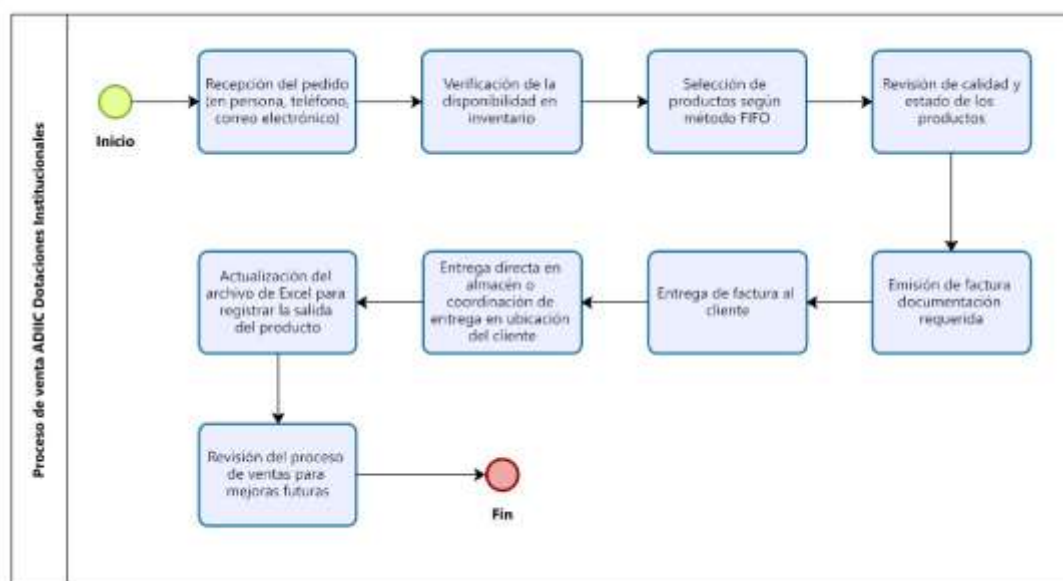
- El proceso inicia con la recepción del pedido, que puede realizarse de forma presencial, telefónica o mediante correo electrónico.
- Se revisa la disponibilidad de los productos solicitados en el inventario para asegurar que se puedan satisfacer los requerimientos del cliente.
- Los productos son seleccionados siguiendo el método FIFO (First In, First Out) (Ladrón de Guevara, 2020), garantizando la rotación adecuada del inventario.
- Los productos seleccionados son sometidos a una inspección para verificar su calidad y estado, asegurando que cumplan con los estándares requeridos.

- Se elabora la factura correspondiente junto con cualquier otra documentación necesaria para formalizar la venta.
- La factura es entregada al cliente como comprobante de la transacción realizada.
- El cliente puede optar por retirar directamente los productos en el almacén o coordinar la entrega en la ubicación que indique.
- Se actualiza un archivo de Excel para registrar la salida de los productos vendidos, manteniendo un control sobre el inventario disponible.
- Finalmente, se revisa el proceso de ventas con el objetivo de identificar oportunidades de mejora y optimizar futuras operaciones.
- El proceso concluye una vez completadas todas las actividades mencionadas.

La secuencia de estas fases se representa en la figura 4.

Figura 4

Diagrama de flujo proceso de ventas.



Fuente. Autoría Propia.

El análisis de los procesos de adquisición y venta permite reconocer una gestión de inventario aceptable, aunque también revela diversos hallazgos que ponen de manifiesto su complejidad y los retrasos que la acompañan:

Falencias en el proceso de adquisición.

- Gestión manual de inventarios: identificación y monitoreo realizados sin herramientas automatizadas, incrementando el tiempo y riesgo de errores.
- Generación manual de alertas y órdenes de compra: procesos dependientes de acciones humanas que ralentizan la respuesta y reposición de inventarios.
- Selección y evaluación de proveedores no digitalizada.

Falencias en el Proceso de venta.

- Recepción de pedidos en formatos variados: integración manual de información procedente de diferentes canales, lo que genera demoras.
- Verificación manual de disponibilidad de inventario: ralentiza la atención a los clientes y afecta la eficiencia en ventas.
- Registro manual de salidas en hojas de cálculo: actualización lenta y propensa a errores, afectando la precisión del inventario.
- Falta de integración entre procesos: la ausencia de un sistema centralizado fragmenta las operaciones, aumentando los tiempos y dificultando la coordinación.

Rediseño del Proceso de Adquisición y Venta

Con el objetivo de reducir la complejidad, minimizar los tiempos de operación y eliminar los riesgos asociados a la gestión manual, se propone un rediseño integral de los procesos de adquisición y venta. Con el propósito de implementar una herramienta tecnológica para

automatizar estas tareas e integrar las operaciones en un sistema centralizado, garantizando la disponibilidad de productos. A continuación, se anuncian las mejoras a los procesos mediante la aplicación Web:

Rediseño Proceso de Adquisición.

- Gestión de adquisición de productos a través de un sistema en línea que permita registrar, actualizar, eliminar los niveles de inventario de cada producto, organizado por categoría.
- Añadir funcionalidades de búsqueda y filtrado para facilitar la localización rápida de los productos.
- Control de stock de los productos en tiempo real mediante la integración del proceso de venta.
- Incorporar en el sistema una sección de alertas automáticas que notifique al administrador cuando los niveles de inventario de un producto alcancen un punto mínimo.

Rediseño Proceso de Venta.

- Incorporar en el sistema un módulo de ventas que permita buscar rápidamente un producto a vender el cual muestre sus características y precio y en caso de realizarse la venta genere la factura electrónica al cliente.
- El sistema debe contar con un módulo de clientes, para el caso de una venta de un nuevo cliente el sistema debe permitir agregarlo rápidamente.
- Actualización automática del stock una vez se realice la venta.

- El sistema debe permitir generar informes sobre las ventas realizadas de forma diaria, semanal y mensual por productos y categorías lo que permitirá realizar un seguimiento a las tendencias de venta y así ajustar el inventario según sea necesario.

Esta propuesta apunta a crear procesos más ágiles, eficientes y confiables, asegurando una operación óptima y una experiencia de cliente satisfactoria.

Definición de las Necesidades

A continuación, se presenta una tabla que define las necesidades del estado futuro para los procesos de adquisición y venta, en función de las mejoras propuestas. Esta tabla detalla las funcionalidades esperadas del sistema de información y cómo estas contribuyen a alcanzar el estado deseado.

Tabla 4*Necesidades del estado futuro para los procesos de adquisición y venta*

Proceso	Necesidad	Funcionalidad de la aplicación Web	Estado deseado
Adquisición	Gestión de adquisición de productos	Sistema en línea para registrar, actualizar y eliminar niveles de inventario, organizado por categorías y subcategorías.	Inventario actualizado y organizado, facilitando la gestión de productos.
	Búsqueda y filtrado de productos	Funcionalidades de búsqueda y filtrado para localizar productos rápidamente.	Localización rápida de productos, mejorando la eficiencia en la adquisición.
	Control de stock en tiempo real	Integración del proceso de venta para reflejar cambios en el inventario en tiempo real.	Visibilidad constante del stock, evitando sobreventas y faltantes.
	Alertas automáticas de inventario	Sección de alertas que notifique al administrador cuando los niveles de inventario alcancen un punto mínimo.	Proactividad en la gestión de inventario, evitando quiebres de stock.
Venta	Búsqueda rápida de productos a vender	Módulo de ventas que permita buscar productos y mostrar características y precios.	Proceso de venta ágil y eficiente, mejorando la experiencia del cliente.
	Generación de factura electrónica	Generación automática de factura electrónica al realizar una venta.	Cumplimiento normativo y reducción de errores en la facturación.
	Registro rápido de nuevos clientes	Módulo de clientes que permita agregar rápidamente nuevos clientes durante el proceso de venta.	Facilita la incorporación de nuevos clientes, mejorando la atención al cliente.
	Actualización automática del stock	Actualización del inventario en tiempo real tras cada venta.	Inventario siempre actualizado, evitando discrepancias entre stock y ventas.
	Generación de informes de ventas	Generación de informes diarios, semanales y mensuales sobre ventas por productos y categorías.	Análisis de tendencias de venta, permitiendo ajustes proactivos en el inventario.

Nota. La tabla presenta las necesidades y funcionalidades del sistema de información para optimizar los procesos de adquisición y venta, mejorando la gestión del inventario y la eficiencia operativa.

Esta tabla proporciona una visión clara de cómo las funcionalidades del sistema de información están alineadas con las necesidades de mejora en los procesos de adquisición y venta, contribuyendo a un estado futuro más ágil, eficiente y confiable.

Diagrama Estado Futuro del Proceso de Adquisición TO-BE

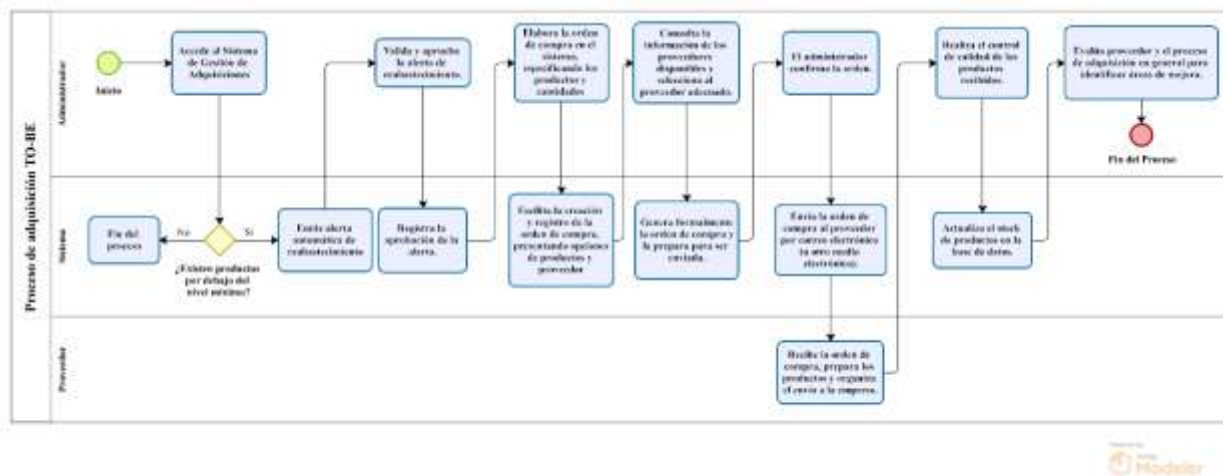
En la propuesta de estado futuro (TO-BE) del proceso de adquisición, se implementa un sistema automatizado de gestión que optimiza la verificación de inventarios y el registro de productos. Las mejoras incluyen la activación de alertas automáticas cuando los niveles de inventario son bajos, lo que permite tomar decisiones oportunas para garantizar la disponibilidad de los productos. Además, se integran flujos de trabajo más eficientes para actualizar los datos de productos existentes o registrar nuevos artículos, asegurando la trazabilidad y la consistencia de la información. En la figura 5 se detalla el flujo de este proceso optimizado, destacando los puntos clave en los que se aplican las mejoras.

- El proceso inicia cuando el administrador accede al Sistema de Gestión de Adquisiciones.
- Si hay productos por debajo del nivel mínimo, el sistema emite una alerta automática de reabastecimiento para el administrador.
- El administrador valida y aprueba la alerta de reabastecimiento.
- El administrador realiza consulta y selección de proveedores.
- Una vez seleccionado el proveedor el administrador elabora orden de compra para ser generado por el sistema.

- El sistema envía orden de compra al proveedor por correo electrónico.
- El proveedor recibe la orden de compra por parte del almacén, prepara los productos y los envía a la empresa.
- Una vez el pedido es recibido en almacén se realiza el control de calidad y si los productos cumplen con las especificaciones, se registran y se actualiza el stock.
- Se realiza evaluación del proveedor y del proceso para mejoras.
- Fin del Proceso.

Figura 5

Diagrama de flujo proceso de adquisición TO-BE



Fuente. Autoría Propia.

Diagrama Estado Futuro del Proceso de Venta TO-BE

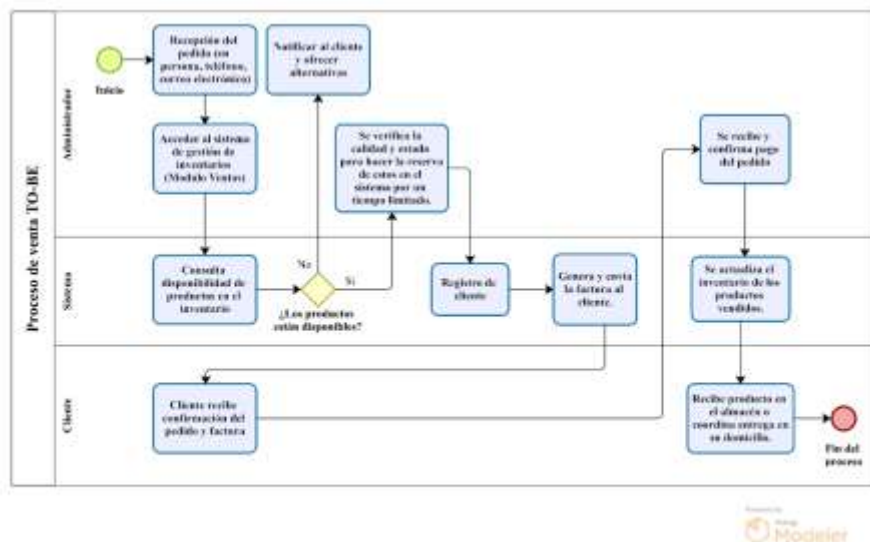
La propuesta del estado futuro (TO-BE) para el proceso de venta redefine el flujo operativo para la verificación de inventarios. Entre las mejoras propuestas destaca la automatización de la consulta de disponibilidad de productos, lo que garantiza que solo se

vendan artículos en stock y se reduzcan los errores en los pedidos. Además, se incorpora un registro en tiempo real de las transacciones, lo que permite la actualización inmediata del inventario tras cada venta. En la figura 6 se describe el proceso revisado, señalando los puntos donde las nuevas implementaciones mejoran la eficiencia y la experiencia del usuario.

- El proceso inicia cuando el administrador recibe el pedido mediante una de las siguientes opciones: en persona, por teléfono o por correo electrónico.
- El usuario accede al sistema de gestión de inventarios (Módulo Ventas) para procesar el pedido.
- El administrador verifica en el módulo de ventas si los productos solicitados por el cliente están disponibles y su cantidad.
 - Si los productos no están disponibles el administrador notifica al cliente sobre la falta de disponibilidad y ofrece alternativas.
 - Si los productos están disponibles se verifica la calidad y estado de los productos para hacer la reserva de estos en el sistema por un tiempo limitado.
- El administrador genera y envía la factura al cliente.
- Cuando se genere confirmación de pago por parte del cliente el sistema actualiza el inventario de los productos vendidos.
- El cliente elige entre retirar en el almacén el producto o coordinar con el vendedor la entrega de este en su domicilio.
- El proceso concluye tras la entrega o retiro del producto.

Figura 6

Diagrama de flujo proceso de venta TO-BE



Fuente. Autoría Propia.

Definición del Producto Mínimo Viable (MVP) del Sistema

Para que el sistema resulte viable y entregable dentro del alcance definido para el proyecto, es imprescindible contar con un conjunto mínimo de funcionalidades que permitan cubrir los procesos clave de operación y gestión. En primer lugar, debe contemplarse un módulo de inicio de sesión y autenticación seguro, capaz de validar identidades y controlar el acceso según roles. A continuación, el sistema ha de incorporar la gestión de usuarios, que posibilite la creación de cuentas, así como la asignación de permisos. Asimismo, será necesario implementar la gestión de categorías y de productos, de manera que el administrador pueda organizar el catálogo y registrar las características esenciales de cada artículo. En el ámbito de venta, se requiere para que el sistema resulte viable y entregable dentro del alcance definido para el

proyecto, es imprescindible contar con un conjunto mínimo de funcionalidades que permitan cubrir los procesos clave de operación y gestión. En primer lugar, debe contemplarse un módulo de inicio de sesión y autenticación seguro, capaz de validar identidades y controlar el acceso según roles. A continuación, el sistema ha de incorporar la gestión de usuarios, que posibilite la creación de cuentas, así como la asignación de permisos. Asimismo, será necesario implementar la gestión de categorías y de productos, de manera que el administrador pueda organizar el catálogo y registrar las características esenciales de cada artículo. En el ámbito de venta, se requiere un módulo para parametrización de cabeceras de factura, que defina atributos como datos de la empresa y formatos de impresión, y otro para la gestión de facturas, que registre operaciones de venta, calcule totales, impuestos y descuentos, y mantenga un historial consultable. Finalmente, la gestión de clientes debe permitir el alta, edición y consulta de información de quienes adquieren productos, asegurando la integridad de los datos de contacto y facturación. Con este conjunto de funcionalidades mínimas, el sistema podrá satisfacer las necesidades básicas de inventario, ventas y control administrativo, garantizando su entrega como un producto funcional, escalable y validable por los usuarios finales, que defina atributos como datos de la empresa y formatos de impresión, y otro para la gestión de facturas, que registre operaciones de venta, calcule totales, impuestos y descuentos, y mantenga un historial consultable. Finalmente, la gestión de clientes debe permitir el alta, edición y consulta de información de quienes adquieren productos, asegurando la integridad de los datos de contacto y facturación. Con este conjunto de funcionalidades mínimas, el sistema podrá satisfacer las necesidades básicas de inventario, ventas y control administrativo, garantizando su entrega como un producto funcional y alineado con los objetivos del proyecto.

Definición de Épicas

En el contexto del desarrollo de software bajo metodologías ágiles, las épicas se definen como unidades funcionales de alto nivel que agrupan características o procesos clave del sistema. Estas constituyen el punto de partida para la planificación estructurada del proyecto, ya que permiten visualizar funcionalidades amplias que posteriormente se descomponen en historias de usuario más específicas, viables y gestionables dentro de los ciclos iterativos de desarrollo. La adecuada formulación de las épicas permite garantizar trazabilidad, control del alcance y alineación con los objetivos funcionales del sistema. Según la Guía de los fundamentos de Scrum (Guía del SBOK®), las épicas permiten capturar requisitos complejos que deben ser fragmentados para facilitar su análisis, diseño e implementación de forma ágil y eficiente (SCRUMstudy, 2022). A continuación, se presenta una tabla en la que se describen las principales épicas definidas para el sistema de gestión de inventarios y venta, detallando el propósito y los componentes funcionales de cada una.

Tabla 5**Épicas funcionales del sistema de gestión de inventario y venta**

ID	Épica	Descripción
EP01	Inicio de sesión y autenticación	Esta épica abarca el diseño e implementación de los mecanismos de acceso seguro al sistema de gestión de inventarios y venta. Se incluyen la creación de la interfaz de inicio de sesión, la validación de credenciales mediante encriptación de contraseñas (por ejemplo, con bcrypt), la emisión y renovación de tokens de acceso (JWT) para el mantenimiento de la sesión. El propósito central es garantizar la confidencialidad, integridad y disponibilidad de la plataforma, estableciendo una base sólida de seguridad que sustente todas las operaciones posteriores.
EP02	Gestión de usuarios	Esta épica se centra en la administración integral de las cuentas de usuario dentro de la aplicación, contemplando la creación de perfiles. Incluye la asignación de roles (administrador, vendedor), la definición de permisos granulares para cada rol y la auditoría de cambios mediante registro de actividad. Adicionalmente, comprende la validación de datos de usuario (por ejemplo, unicidad de correo electrónico), con el fin de asegurar un control riguroso de acceso y un adecuado seguimiento de la gestión de usuarios.
EP03	Gestión de categorías	Engloba la creación y mantenimiento del catálogo de categorías de productos, incluyendo la definición de atributos y la jerarquización de niveles si aplica. Se contempla la validación de la unicidad de los nombres, la edición de descripciones y la eliminación segura que preserve la integridad referencial con los productos asociados. Asimismo, se desarrollan mecanismos de filtrado y búsqueda eficiente, orientados a optimizar la experiencia del usuario en la segmentación y consulta del inventario.
EP04	Gestión de productos	Esta épica aborda el módulo de administración del catálogo de productos, permitiendo el registro y modificación de atributos críticos como nombre, descripción, código interno, precio, impuestos aplicables (IVA), descuentos y valores de stock. Incluye la vinculación de cada producto a una categoría, la gestión de imágenes. Su objetivo es proporcionar una visión precisa y actualizada del estado del inventario, facilitando tanto la gestión interna como la generación de reportes comerciales.
EP05	Cabeceras para facturas	En esta épica se define y configura la sección inicial de las facturas, cabecera, que contiene los datos estáticos y de contexto necesarios para la emisión de documentos oficiales. Se incluyen elementos como la razón social del almacén, la identificación tributaria (NIT), la dirección física, el logotipo corporativo, el número de factura correlativo, la fecha de

		expedición y la información de contacto. La correcta parametrización de esta información asegura la conformidad con los requisitos legales y tributarios, así como la presentación uniforme de los documentos generados.
EP06	Gestión de facturas	Esta épica comprende el ciclo completo de emisión, edición, visualización de facturas de venta. Se implementa el cálculo automático de subtotales, descuentos, impuestos (IVA) y totales generales, integrando los datos de cabecera y el cuerpo de factura (detalle de productos vendidos). Además, incluye la generación de la representación en PDF para distribución al cliente y el almacenamiento del historial en la base de datos, permitiendo consultas filtradas. El alcance también abarca la validación.
EP07	Gestión de clientes	Enfocada en la administración de la base de datos de clientes, contemplando el registro y actualización de la información de contacto, datos fiscales (NIT o cedula), direcciones y demás. Se desarrollan funcionalidades de búsqueda avanzada, segmentación por criterios comerciales y generación de reportes de historial de compras. Asimismo, se asegura la integridad y consistencia de los datos mediante validaciones de formato y unicidad, de modo que la plataforma ofrezca una vista confiable y actualizada de la cartera de clientes.

Nota. La tabla resume las épicas del sistema (ID, nombre y descripción): autenticación, gestión de usuarios, categorías, productos, facturación (cabeceras y emisión) y clientes.

Historias de Usuario

En el marco del desarrollo ágil, las historias de usuario constituyen una herramienta fundamental para capturar los requerimientos desde la perspectiva del cliente o usuario final. Para el presente trabajo, las historias de usuario se representarán mediante el formato de tarjeta bajo el criterio INVEST (independientes, negociables, valiosas, estimables, pequeñas y comprobables), como se muestra en la Figura No. 7.

Figura 7

Formato tarjeta de historia de usuario

Historias de usuario:		Titulo:	
Como .			
Quiero .			
Para poder .			
Criterios de aceptación:			
1.			
2.			
3.			
4.			
5.			
I	N	V	Valor:
□	□	□	Estimación:
□	□	□	¿Cumple con la definición de listo del equipo?

Fuente. Autoría Propia.

A continuación, en las figuras de la 8 a la 39 se presentan las diferentes historias de usuario utilizadas para la realización del proyecto. Estas historias fueron diseñadas con el propósito de definir de manera clara y concisa los requisitos funcionales y no funcionales del sistema, asegurando que cada necesidad del usuario sea atendida de manera efectiva. Cada historia detalla las funcionalidades esperadas, los actores involucrados y los criterios de aceptación, permitiendo una mejor planificación y desarrollo del proyecto.

Figura 8

Historia de usuario HU001 - Página de inicio con login integrado

Historias de usuario: HU001		Titulo: Página de inicio con Login Integrado	
Como usuario de la aplicación,			
Quiero contar con una página de inicio que muestre el logo, el nombre de la empresa, una breve descripción de los servicios y el formulario de inicio de sesión,			
Para poder obtener una experiencia clara y profesional que me permita acceder a la aplicación de manera intuitiva.			
Criterios de aceptación:			
1. La página de inicio debe mostrar el logo y nombre de la empresa en la parte superior central.			
2. Debe incluir un fondo representativo de un almacén o inventario.			
3. Debe contener una breve descripción de los servicios ofrecidos.			
4. Se debe mostrar el componente de Login que permita autenticarse.			
5. Debe adaptarse a diferentes tamaños de pantalla (móvil, tablet, escritorio).			
I	N	V	Valor:
☑	☑	☑	85/100
☑	☑	☑	Estimación:
☑	☑	☑	4 horas
☑	☑	☑	¿Cumple con la definición de listo del equipo?
☑	☑	☑	☑

Fuente. Autoría Propia.

Figura 9

Historia de usuario HU002 – Formulario de login con redirección por rol.

Historias de usuario: HU002	Título: Formulario de Login con redirección por rol.
<p>Como usuario registrado, Quiero acceder a la aplicación ingresando mi correo electrónico y contraseña, Para poder acceder a la plataforma según mi rol asignado.</p>	
<p>Criterios de aceptación:</p> <ol style="list-style-type: none"> 1. El sistema debe permitir el ingreso de un correo electrónico y una contraseña. 2. El correo electrónico debe tener un formato válido y pertenecer a un dominio permitido. 3. La contraseña debe tener al menos 5 caracteres. 4. Si las credenciales son correctas, el sistema debe autenticar al usuario y redirigirlo según su rol. 5. Si el usuario no está registrado, debe mostrarse un mensaje de error. 6. Si la contraseña es incorrecta, debe mostrarse un mensaje de error. 7. El sistema debe almacenar un token JWT en el almacenamiento local del navegador. 8. Si el usuario ya ha iniciado sesión, debe ser redirigido automáticamente a su respectiva página según su rol. 	
I	Valor: 95/100
N	Estimación: 6 horas
V	¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/>
E	
S	
T	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	

Fuente. Autoría Propia.

Figura 10

Historia de usuario HU003 – Sidebar para acceso a funcionalidades de gestión

Historias de usuario: HU003	Título: Sidebar para Acceso a Funcionalidades de Gestión
<p>Como administrador, Quiero un Sidebar con accesos directos a las funcionalidades de gestión, Para poder administrar eficientemente las categorías, productos, clientes, cabeceras y facturas del sistema.</p>	
<p>Criterios de aceptación:</p> <ol style="list-style-type: none"> 1. El Sidebar debe estar disponible en todas las páginas donde el administrador tenga acceso. 2. Debe contener enlaces de navegación hacia: categorías, crear categoría, crear producto, crear cuenta, clientes, cabeceras y facturas. 3. Los botones deben estar estilizados con diseño visual atractivo y responsivo. 4. El Sidebar debe mantenerse fijo en el lado izquierdo de pantallas de PC y adaptarse a diferentes tamaños de pantalla. 5. La navegación debe ser intuitiva, con botones grandes y texto en mayúsculas. 6. Si el administrador tiene una categoría seleccionada (idCategoría ≠ null), se debe habilitar el botón de "Crear Producto". 7. Si no hay categoría seleccionada (idCategoría = null), el botón "Crear Producto" debe estar deshabilitado visual y funcionalmente. 	
I	Valor: 80/100
N	Estimación: 3 horas
V	¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/>
E	
S	
T	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	

Fuente. Autoría Propia.

Figura 11

Historia de usuario HU004 – Encabezado de la aplicación Web

Historias de usuario: HU004		Título: Encabezado de la Aplicación Web	
Como usuario autenticado, Quiero visualizar un encabezado en la aplicación web que contenga el logo de la empresa, el nombre y descripción del negocio, y un botón para cerrar sesión, Para poder identificar rápidamente la plataforma y gestionar mi sesión de usuario.			
Criterios de aceptación: 1. El encabezado debe estar presente en todas las páginas de la aplicación después de iniciar sesión. 2. Debe mostrar: Logo de la empresa (izquierda), nombre y descripción del negocio (centro), botón de "Cerrar Sesión" (derecha). 3. El diseño debe ser responsivo, visible correctamente en diferentes tamaños de pantalla. 4. Al hacer clic en "Cerrar Sesión": Se debe eliminar el token JWT del almacenamiento local, el usuario debe ser redirigido automáticamente a la página de inicio.			
I	N	V	E S T
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
			Valor: 75/100
			Estimación: 5 horas
			¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Figura 12

Historia de usuario HU005 – Página de administración de categorías y productos

Historias de usuario: HU005		Título: Página de Administración de Categorías y Productos	
Como administrador del sistema, Quiero acceder a una página de administración donde pueda visualizar, actualizar y eliminar categorías, así como gestionar los productos asociados, Para poder mantener actualizada la organización de los artículos en la plataforma.			
Criterios de aceptación: 1. Si el usuario no está autenticado, debe ser redirigido a la página de inicio de sesión. 2. El acceso al módulo debe validarse mediante la existencia de un token JWT. 3. El administrador debe poder: Ver una lista de categorías con imagen y nombre, eliminar una categoría con confirmación previa, editar una categoría desde una página específica de edición, acceder a la gestión de productos por categoría. 4. Las categorías deben cargarse dinámicamente desde una API. 5. Deben mostrarse mensajes de éxito o error según el resultado de las acciones. 6. Mientras se cargan los datos, debe visualizarse un indicador de carga.			
I	N	V	E S T
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
			Valor: 90/100
			Estimación: 8 horas
			¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Figura 13

Historia de usuario HU006 – Registro de nuevos usuarios con asignación de rol

Historias de usuario: HU006		Título: Registro de Nuevos Usuarios con Asignación de Rol					
Como administrador, Quiero registrar nuevos usuarios en la plataforma asignándoles un rol específico, Para poder gestionar el acceso y permisos según sus funciones dentro del sistema.							
Criterios de aceptación: 1. Solo los usuarios con rol de "admin" pueden acceder al formulario de registro. 2. El formulario debe incluir los siguientes campos obligatorios: Nombre, Correo electrónico, Rol (admin o regular), Contraseña, Confirmación de contraseña. 3. Si la contraseña y su confirmación no coinciden, debe mostrarse un mensaje de error. 4. Si el correo electrónico ya existe en la plataforma, debe mostrarse un mensaje de error. 5. Si la información es válida, el sistema debe registrar al nuevo usuario y mostrar una confirmación. 6. Al completar el registro, sera redirigido al panel de administración.							
I	N	V	E	S	T	Valor: 85/100	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Estimación: 4 horas	
<input checked="" type="checkbox"/>						¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/>	

Fuente. Autoría Propia.

Figura 14

Historia de usuario HU007 – Creación de nuevas categorías

Historias de usuario: HU007		Título: Creación de Nuevas Categorías					
Como administrador, Quiero crear nuevas categorías en el sistema, Para poder organizar los productos de manera eficiente.							
Criterios de aceptación: 1. El administrador debe haber iniciado sesión para acceder a la página de creación de categorías. 2. El formulario debe incluir campos obligatorios: Nombre de la categoría, URL de la imagen representativa. 3. Debe existir un botón para registrar la categoría. 4. Si la categoría se crea correctamente: Mostrar mensaje de éxito, redirigir automáticamente a la vista principal de administración. 5. Si ocurre un error, debe mostrarse un mensaje de error. 6. El formulario debe ser intuitivo y responsivo, adaptado a distintos dispositivos.							
I	N	V	E	S	T	Valor: 80/100	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Estimación: 6 horas	
<input checked="" type="checkbox"/>						¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/>	

Fuente. Autoría Propia.

Figura 15

Historia de usuario HU008 – Actualización de categorías existentes

Historias de usuario: HU008		Título: Actualización de Categorías Existentes	
Como administrador, Quiero actualizar la información de una categoría existente, Para mantener la información de las categorías correctamente actualizada en el sistema.			
Criterios de aceptación:			
<ol style="list-style-type: none"> 1. El administrador debe estar autenticado para acceder a esta funcionalidad. 2. El sistema debe cargar automáticamente los datos actuales de la categoría seleccionada. 3. El formulario debe permitir modificar el: Nombre de la categoría e imagen representativa. 4. Al hacer clic en “Actualizar Categoría”, el sistema debe validar los campos antes de enviar. 5. Si la actualización es exitosa: Mostrar un mensaje de confirmación, redirigir al administrador a la vista de administración. 6. Si ocurre un error, se debe mostrar un mensaje de error claro. 7. Si el usuario no está autenticado, debe ser redirigido al login. 			
I		Valor: 75/100	
N		Estimación: 5 horas	
V		¿Cumple con la definición de listo del equipo?	
E		<input checked="" type="checkbox"/>	
S			
T			
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Figura 16

Historia de usuario HU009 – Actualización de información de un producto

Historias de usuario: HU009		Título: Actualización de Información de un Producto	
Como administrador, Quiero actualizar la información de un producto existente, Para mantener los datos correctos y actualizados en el inventario.			
Criterios de aceptación:			
<ol style="list-style-type: none"> 1. El sistema debe recuperar y mostrar los datos actuales del producto seleccionado basándose en su ID. 2. Si el producto no existe o hay error al cargarlo, debe mostrarse un mensaje de error. 3. El formulario debe permitir modificar los siguientes campos: Referencia, Nombre, Descripción, Talla, Color, Stock, Precio. 4. Todos los campos son obligatorios, excepto la imagen. 5. El stock debe ser un número entero positivo o cero. 6. El precio debe ser un número mayor a cero. 7. Si algún dato no es válido, el sistema debe mostrar un mensaje de error. 8. Al hacer clic en “Actualizar Producto”: Si los datos son válidos, debe enviarse la solicitud PUT al servidor, si es exitosa, debe mostrarse un mensaje de confirmación y redirigir al listado de productos de la categoría, si falla, debe mostrarse un mensaje de error. 			
I		Valor: 85/100	
N		Estimación: 3 horas	
V		¿Cumple con la definición de listo del equipo?	
E		<input checked="" type="checkbox"/>	
S			
T			
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Figura 17

Historia de usuario HU010 – Registro de nuevos productos asociados a categorías

Historias de usuario: HU010		Título: Registro de Nuevos Productos Asociados a Categorías	
<p>Como administrador, Quiero registrar nuevos productos en el sistema asignándolos a una categoría específica, Para que estos puedan ser visualizados y gestionados en la plataforma.</p>			
<p>Criterios de aceptación:</p> <ol style="list-style-type: none"> 1. El administrador debe estar autenticado para acceder a esta funcionalidad. 2. El formulario debe incluir los siguientes campos obligatorios: Referencia, Nombre, Descripción, Talla, Color, Stock, Precio, Imagen. 3. Los campos stock y precio deben aceptar solo valores numéricos válidos. 4. Al enviar el formulario: El producto debe registrarse en la base de datos, debe asociarse correctamente a la categoría seleccionada. 5. Si la creación es exitosa: Se debe mostrar un mensaje de confirmación, redirigir a la página de listado de productos de la categoría. 6. Si ocurre un error, debe mostrarse un mensaje con la causa. 7. Debe incluirse un Sidebar para navegación dentro del panel de administración. 8. Debe incluirse un header con botón de cierre de sesión. 			
		Valor:	90/100
		Estimación:	7 horas
I	N	V	E
S	T		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
¿Cumple con la definición de listo del equipo?			<input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Figura 18

Historia de usuario HU011 – Listado de productos por categoría

Historias de usuario: HU011		Título: Listado de Productos por Categoría	
<p>Como administrador, Quiero ver un listado de productos organizados por categoría, Para poder gestionar y visualizar los productos de manera eficiente.</p>			
<p>Criterios de aceptación:</p> <ol style="list-style-type: none"> 1. Se debe obtener el ID de la categoría a partir de la URL. 2. Si no se detecta un ID de categoría, debe mostrarse un mensaje de error en consola. 3. El sistema debe hacer una solicitud a la API para obtener los productos de la categoría correspondiente. 4. Mientras se cargan los productos, debe mostrarse un mensaje de "Cargando productos...". 5. Si hay productos disponibles: Deben mostrarse en la interfaz con sus respectivos datos. 6. Si no hay productos disponibles: Debe mostrarse el mensaje "No hay productos disponibles". 7. La interfaz debe incluir un encabezado, un sidebar y una lista bien estructurada. 8. El diseño debe ser responsivo y visualmente atractivo, usando colores y estilos adecuados. 			
		Valor:	80/100
		Estimación:	4 horas
I	N	V	E
S	T		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
¿Cumple con la definición de listo del equipo?			<input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Figura 19

Historia de usuario HU012 – Visualización y gestión de productos de una categoría específica

Historias de usuario: HU012	Título: Visualización y Gestión de Productos de una Categoría Específica
Como administrador, Quiero visualizar los productos de una categoría específica, Para poder gestionar su información y disponibilidad.	
Criterios de aceptación: 1. Al cargar la vista de productos, el sistema debe recuperar y mostrar la lista de productos de la categoría seleccionada. 2. Cada producto debe mostrar: referencia, nombre, descripción, talla, color, stock, precio e imagen. 3. Al hacer clic en "Editar", el usuario debe ser redirigido a la página de actualización con los datos del producto seleccionado. 4. Al hacer clic en "Eliminar", debe mostrarse una alerta de confirmación. 5. Si se confirma la eliminación: El producto debe eliminarse de la base de datos, la lista debe actualizarse en la interfaz. 6. Si se cancela la eliminación, no debe ocurrir ningún cambio. 7. Si ocurre un error al cargar productos o al eliminarlos, se debe mostrar un mensaje de error. 8. La interfaz debe ser responsiva y adaptarse a distintos dispositivos.	
I	Valor: 85/100
N	Estimación: 6 horas
V	
E	
S	
T	
<input checked="" type="checkbox"/>	¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Figura 20

Historia de usuario HU013 – Visualización de categorías por parte del usuario regular

(Vendedor)

Historias de usuario: HU013	Título: Visualización de Categorías por parte del Usuario Regular (Vendedor)
Como usuario regular (vendedor), Quiero visualizar las categorías de productos disponibles en el sistema, Para poder seleccionar una y gestionar sus productos.	
Criterios de aceptación: 1. Si el usuario no ha iniciado sesión, debe ser redirigido a la página de inicio de sesión. 2. El sistema debe validar la existencia del token en el almacenamiento local antes de mostrar la vista. 3. Las categorías deben obtenerse desde la API /api/categorias al cargar la página. 4. Si la respuesta de la API contiene datos válidos, las categorías deben mostrarse en la interfaz. 5. Si ocurre un error al obtener las categorías, debe mostrarse un mensaje con opción de reintento. 6. La página debe tener un encabezado con el título "Panel de vendedores". 7. Cada categoría debe mostrarse como tarjeta con: Imagen, nombre y botón para ver productos asociados. 8. El botón "Productos" debe dirigir a la página de gestión de productos de la categoría seleccionada. 9. La interfaz debe ser responsiva, adaptándose a distintos tamaños de pantalla. 10. Se debe aplicar una paleta de colores en tonos lima y verde.	
I	Valor: 80/100
N	Estimación: 8 horas
V	
E	
S	
T	
<input checked="" type="checkbox"/>	¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Figura 21

Historia de usuario HU014 – Visualización de productos por categoría para vendedores

Historias de usuario: HU014	Título: Visualización de Productos por Categoría para Vendedores												
<p>Como usuario regular (vendedor), Quiero ver los productos de una categoría específica, Para poder conocer su información y disponibilidad.</p>													
<p>Criterios de aceptación:</p> <ol style="list-style-type: none"> 1. Si el usuario no está autenticado, debe ser redirigido a la página de inicio de sesión. 2. El sistema debe obtener la información de la categoría seleccionada desde la API. 3. El sistema debe obtener la lista de productos asociados a la categoría desde la API. 4. Si ocurre un error al cargar datos, debe mostrarse un mensaje de error informativo. 5. El nombre de la categoría debe mostrarse en la parte superior de la interfaz. 6. Si hay productos en la categoría: Deben mostrarse en una cuadrícula con su imagen y detalles (nombre, precio, stock, etc.). 7. Si no hay productos: Debe mostrarse un mensaje indicando que no hay productos disponibles. 8. Debe incluirse un botón para regresar a la vista de categorías. 9. La interfaz debe ser responsiva y accesible desde distintos dispositivos. 													
Valor: 75/100													
Estimación: 6 horas													
<table border="0"> <tr> <td>I</td><td>N</td><td>V</td><td>E</td><td>S</td><td>T</td> </tr> <tr> <td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td> </tr> </table>	I	N	V	E	S	T	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<p>¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/></p>
I	N	V	E	S	T								
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>								

Fuente. Autoría Propia.

Figura 22

Historia de usuario HU015 – Creación de cabeceras para facturas por sucursal

Historias de usuario: HU015	Título: Creación de Cabeceras para Facturas por Sucursal												
<p>Como administrador, Quiero poder crear nuevas cabeceras para las facturas con información como nombre del local, NIT, dirección, teléfono y correo electrónico, Para poder personalizar la factura según la sucursal emisora.</p>													
<p>Criterios de aceptación:</p> <ol style="list-style-type: none"> 1. El sistema debe verificar que el usuario tiene permisos de administrador. 2. Los campos obligatorios del formulario son: Nombre del local, dirección, NIT, teléfono y correo electrónico. 3. Se debe validar que ningún campo esté vacío. 4. El formato de la Cabecera para la factura debe ser válido según normativas fiscales locales. 5. Debe verificarse que no exista otra cabecera con el mismo NIT. 6. Si los datos son válidos: La cabecera debe registrarse correctamente en la base de datos, debe mostrarse un mensaje de confirmación. 													
Valor: 85/100													
Estimación: 5 horas													
<table border="0"> <tr> <td>I</td><td>N</td><td>V</td><td>E</td><td>S</td><td>T</td> </tr> <tr> <td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td> </tr> </table>	I	N	V	E	S	T	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<p>¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/></p>
I	N	V	E	S	T								
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>								

Fuente. Autoría Propia.

Figura 23

Historia de usuario HU016 – Edición de cabeceras

Historias de usuario: HU016		Título: Edición de Cabeceras					
Como administrador, Quiero poder editar las cabeceras existentes, Para actualizar la información en caso de cambios en los datos de la sucursal.							
Criterios de aceptación: 1. El sistema debe verificar que el usuario tiene permisos de administrador. 2. El formulario de edición debe mostrar los datos actuales cargados automáticamente. 3. Se deben validar los siguientes campos obligatorios: Nombre del local, NIT, Dirección, teléfono y correo electrónico. 4. El formato de la cabecera para la factura debe mantenerse válido según las normas fiscales aplicables. 5. No debe permitirse la existencia de otra cabecera con el mismo NIT. 6. Si la actualización es exitosa: Se debe guardar la información modificada, debe mostrarse un mensaje de confirmación. 7. Si ocurre un error, se debe mostrar un mensaje de advertencia adecuado.							
I	N	V	E	S	T	Valor: 80/100	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Estimación: 6 horas	
¿Cumple con la definición de listo del equipo?							<input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Figura 24

Historia de usuario HU017 – Eliminación de cabeceras innecesarias

Historias de usuario: HU017		Título: Eliminación de Cabeceras Innecesarias					
Como administrador, Quiero poder eliminar cabeceras que ya no sean necesarias, Para mantener organizada la base de datos.							
Criterios de aceptación: 1. El sistema debe verificar que el usuario tenga permisos de administrador. 2. Se debe solicitar una doble confirmación al administrador antes de proceder con la eliminación. 3. Si la cabecera cumple con las condiciones y se confirma la acción: Debe eliminarse correctamente de la base de datos, debe mostrarse un mensaje de éxito. 4. Si no se cumplen las condiciones o se cancela la acción: La cabecera debe mantenerse intacta, se debe notificar adecuadamente al usuario.							
I	N	V	E	S	T	Valor: 70/100	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Estimación: 4 horas	
¿Cumple con la definición de listo del equipo?							<input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Figura 25

Historia de usuario HU018 – Búsqueda de cabeceras por NIT en gestión administrativa

Historias de usuario: HU018	Título: Búsqueda de Cabeceras por NIT en Gestión Administrativa
<p>Como administrador, Quiero buscar cabeceras existentes ingresando el NIT, Para localizarlas rápidamente y poder editarlas o eliminarlas con eficiencia.</p>	
<p>Criterios de aceptación:</p> <ol style="list-style-type: none"> 1. El campo de búsqueda debe aceptar únicamente valores válidos de NIT. 2. Si existe una cabecera con el NIT ingresado, debe mostrarse en los resultados. 3. Si no existe, debe mostrarse un mensaje claro de "No se encontraron resultados". 4. La búsqueda debe ofrecer retroalimentación inmediata o casi en tiempo real. 5. La funcionalidad debe integrarse dentro del módulo de Gestión de Cabeceras. 	
I	Valor: 70/100
N	Estimación: 3 horas
V	
E	
S	
T	
<input checked="" type="checkbox"/>	¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Figura 26

Historia de usuario HU019 – Selección de cabecera al crear una factura

Historias de usuario: HU019	Título: Selección de Cabecera al Crear una Factura
<p>Como usuario de la aplicación, Quiero poder seleccionar una cabecera específica al momento de crear una factura, Para asegurar que la información de la sucursal emisora sea la correcta.</p>	
<p>Criterios de aceptación:</p> <ol style="list-style-type: none"> 1. El sistema debe mostrar un menú desplegable con las cabeceras disponibles activas. 2. El usuario debe seleccionar una cabecera antes de completar la factura. 3. Solo se deben mostrar en el menú las cabeceras habilitadas y con datos completos. 4. Si el usuario intenta generar una factura sin haber seleccionado una cabecera: El sistema debe impedirlo, debe mostrarse un mensaje de advertencia. 5. La cabecera seleccionada debe reflejar: Nombre del local, NIT, dirección, teléfono y correo electrónico en la factura generada. 6. La funcionalidad debe ser intuitiva y rápida. 	
I	Valor: 90/100
N	Estimación: 6 horas
V	
E	
S	
T	
<input checked="" type="checkbox"/>	¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Figura 27

Historia de usuario HU020 – Creación de nueva factura

Historias de usuario: HU020		Titulo: Creación de Nueva Factura	
Como usuario de la aplicación, Quiero poder crear una nueva factura ingresando los datos de la cabecera, cliente y seleccionando los productos que vendo, Para registrar la venta y entregar un comprobante al cliente.			
Criterios de aceptación:			
1. El sistema debe verificar que se ha seleccionado una cabecera válida antes de permitir la creación.			
2. El formulario debe incluir campos obligatorios del cliente: Nombre completo, Identificación (NIT o cédula), dirección, ciudad, teléfono.			
3. La identificación debe validarse con formato correcto según tipo.			
4. El vendedor debe poder seleccionar uno o varios productos disponibles en el inventario.			
5. Se debe validar que al menos un producto haya sido agregado antes de permitir la facturación.			
6. El sistema debe calcular dinámicamente el resumen de la venta: Subtotal, Descuento, IVA, Total.			
7. Al hacer clic en el botón "Generar Factura", el sistema debe: Validar todos los campos, registrar la factura en la base de datos, mostrar un mensaje de confirmación.			
8. Si algún dato es inválido o falta información, debe mostrarse un mensaje de error claro y específico.			
I	N	V	Valor: 95/100
E	S	T	Estimación: 5 horas
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Figura 28

Historia de usuario HU021 – Agregar productos con cantidad, descuento e IVA a la factura

Historias de usuario: HU021		Titulo: Agregar Productos con Cantidad, Descuento e IVA a la Factura	
Como usuario de la aplicación, Quiero poder agregar múltiples productos a una factura indicando su cantidad, descuento e IVA, Para reflejar con precisión lo que el cliente está comprando y cumplir con la normativa fiscal.			
Criterios de aceptación:			
1. El vendedor debe poder seleccionar productos del inventario y agregarlos uno por uno a la factura.			
2. Para cada producto agregado, debe poder ingresarse: Cantidad (número entero positivo > 0), porcentaje de descuento (entre 0% y 100%), porcentaje de IVA (por lo general 0%, 3%, 19% u otro definido por el sistema fiscal).			
3. El sistema debe verificar que: La cantidad no exceda el stock disponible, el descuento y el IVA estén dentro de los rangos permitidos.			
4. El sistema debe calcular automáticamente por cada producto: Subtotal sin IVA ni descuento, monto descontado, monto de IVA, total final del producto.			
5. El resumen general de la factura debe actualizarse dinámicamente al modificar productos.			
6. La interfaz debe permitir: Editar cantidades, descuentos e IVA antes de finalizar o eliminar productos de la factura.			
7. Si algún dato ingresado es inválido, el sistema debe: Bloquear la acción, mostrar un mensaje de error o advertencia clara.			
I	N	V	Valor: 95/100
E	S	T	Estimación: 3 horas
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Figura 29

Historia de usuario HU022 – Cálculo automático de totales en la factura

Historias de usuario: HU022		Titulo: Cálculo Automático de Totales en la Factura	
Como usuario de la aplicación, Quiero que el sistema calcule automáticamente los subtotales, descuentos, IVA y total final mientras agrego productos, Para visualizar el monto a cobrar en tiempo real y asegurar precisión en la factura.			
Criterios de aceptación:			
1. Cada vez que se agregue, edite o elimine un producto de la factura, el sistema debe recalcular automáticamente: Subtotal sin descuentos ni IVA, monto total de descuento aplicado, monto total de IVA, total final de la factura.			
2. Los cálculos deben actualizarse en tiempo real o con mínima latencia perceptible.			
3. El porcentaje de IVA aplicado debe corresponder al configurado para cada producto.			
4. El descuento por producto debe integrarse correctamente antes de aplicar el IVA.			
5. Los resultados deben ser numéricamente precisos y libres de errores de redondeo.			
6. Cualquier modificación en cantidad, descuento o eliminación de un producto debe actualizar el resumen.			
7. La interfaz debe mostrar un resumen detallado y claro de los montos calculados.			
8. Si los datos de algún producto son inválidos, el sistema no debe recalcular y debe mostrar un mensaje de error.			
I N V E S T		Valor:	95/100
		Estimación:	8 horas
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
¿Cumple con la definición de listo del equipo?			<input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Figura 30

Historia de usuario HU023 – Verificación automática del stock al facturar

Historias de usuario: HU023		Titulo: Verificación Automática del Stock al Facturar	
Como usuario de la aplicación, Quiero que el sistema verifique automáticamente si hay suficiente stock de los productos que estoy añadiendo a la factura, Para evitar vender más de lo disponible.			
Criterios de aceptación:			
1. Al intentar agregar un producto a la factura, el sistema debe consultar en tiempo real el stock actual disponible en inventario.			
2. Si la cantidad ingresada supera el stock, el sistema debe: Bloquear la acción, mostrar un mensaje claro de error (por ejemplo: "Solo hay 'X' unidades disponible").			
3. El sistema debe bloquear la posibilidad de aumentar la cantidad por encima del stock desde cualquier control numérico.			
4. Si se modifica el stock en la base de datos (por otra venta), el sistema debe actualizar automáticamente el valor disponible en la interfaz antes de generar la factura.			
5. Solo debe permitirse facturar productos con cantidad mayor que cero y dentro del stock.			
6. Si hay problemas de conexión o falla al consultar el stock: Debe mostrarse un mensaje de error, no debe permitirse continuar con la operación.			
I N V E S T		Valor:	95/100
		Estimación:	6 horas
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
¿Cumple con la definición de listo del equipo?			<input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Figura 31

Historia de usuario HU024 – Edición de factura antes de finalizar

Historias de usuario: HU024		Título: Edición de Factura Antes de Finalizar	
Como usuario de la aplicación, Quiero poder editar una factura al momento de estar creandola, Para corregir errores o realizar cambios solicitados por el cliente antes de finalizar la venta.			
Criterios de aceptación:			
1. El sistema debe verificar que la factura aún no ha sido impresa ni finalizada para permitir su edición.			
2. El vendedor debe poder: Modificar cantidades de productos, cambiar productos agregados, editar descuentos, actualizar los datos del cliente si es necesario.			
3. Al editar las cantidades: El sistema debe verificar que no se exceda el stock disponible actualizado.			
4. Cada modificación debe recalcular automáticamente los subtotales, descuentos, IVA y total final de la factura.			
5. Todos los campos obligatorios deben mantenerse completos y válidos tras la edición.			
6. Al generar los cambios, el sistema debe: Actualizar la información de los datos de la factura que se esta generando para ser guardados en la base de datos.			
7. Si ocurre un error, debe mostrarse un mensaje explicativo y no guardar cambios inconsistentes.			
I		Valor: 85/100	
N		Estimación: 5 horas	
V		¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/>	
E			
S			
T			
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Figura 32

Historia de usuario HU025 – Descarga de factura finalizada en PDF

Historias de usuario: HU025		Título: Descarga de Factura Finalizada en PDF	
Como usuario de la aplicación, Quiero poder descargar la factura en formato PDF una vez finalizada, Para entregarle al cliente un comprobante de su compra que pueda imprimirse o enviarse digitalmente.			
Criterios de aceptación:			
1. El sistema debe verificar que la factura esté en estado “finalizada” antes de habilitar la descarga.			
2. Al hacer clic en el botón “Generar factura”, el sistema debe: Validar que los datos del cliente y los productos estén completos, calcular y mostrar el resumen de la venta (subtotal, descuentos, IVA, total), generar un archivo PDF de la factura con diseño adecuado.			
3. El archivo debe incluir: Cabecera (nombre del local, NIT, dirección, teléfono y correo electrónico), datos del cliente, Fecha y hora de emisión, detalles de los productos vendidos (cantidad, precio, IVA, descuento), Total final.			
4. El archivo PDF debe descargarse automáticamente en el dispositivo del usuario.			
5. El sistema debe marcar la factura como “descargada” o “emitida” para evitar ediciones posteriores.			
6. Si ocurre un error al generar el archivo, debe mostrarse un mensaje claro y evitar marcar la factura como finalizada.			
I		Valor: 95/100	
N		Estimación: 6 horas	
V		¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/>	
E			
S			
T			
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Figura 33

Historia de usuario HU026 – Búsqueda de facturas por número o NIT (Administrador)

Historias de usuario: HU026	Título: Búsqueda de Facturas por Número o NIT (Administrador)						
<p>Como administrador, Quiero poder buscar facturas anteriores por su número o por el NIT del cliente, Para atender solicitudes de cambios, devoluciones o auditorías y facilitar la trazabilidad de las ventas.</p>							
<p>Criterios de aceptación:</p> <ol style="list-style-type: none"> 1. El campo de búsqueda debe permitir ingresar: El número de factura o el NIT del cliente. 2. El sistema debe validar que el campo: No esté vacío. 3. Si existen resultados coincidentes: Deben mostrarse con información relevante (número, cliente, fecha, total). 4. Si no existen coincidencias: Debe mostrarse un mensaje claro como "No se encontraron resultados". 5. El sistema debe realizar la búsqueda de forma rápida y eficiente. 6. El administrador debe tener acceso completo a todas las facturas del sistema, sin restricciones por usuario. 7. Desde el resultado, debe poder acceder al detalle completo de la factura seleccionada. 							
I	N	V	E	S	T	Valor:	90/100
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Estimación:	4 horas
¿Cumple con la definición de listo del equipo?							<input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Figura 34

Historia de usuario HU027 – Descuento automático del stock al generar factura

Historias de usuario: HU027	Título: Descuento Automático del Stock al Generar Factura						
<p>Como usuario de la aplicación, Quiero que al generar una factura de venta, el sistema descuente automáticamente del stock la cantidad de productos vendidos, Para mantener el inventario actualizado y reflejar las ventas en tiempo real.</p>							
<p>Criterios de aceptación:</p> <ol style="list-style-type: none"> 1. El sistema debe ejecutar el descuento del stock únicamente después de que la factura haya sido generada exitosamente. 2. La cantidad descontada debe corresponder exactamente a los valores registrados por producto en la factura. 3. El descuento debe reflejarse inmediatamente en la base de datos del inventario. 4. No debe permitirse que el stock llegue a valores negativos. 5. Si hay error durante la transacción de descuento, la factura no debe completarse, y debe mostrarse un mensaje de error. 6. Si la factura contiene múltiples productos, deben descontarse todos correctamente en la misma operación (transacción segura). 							
I	N	V	E	S	T	Valor:	95/100
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Estimación:	5 horas
¿Cumple con la definición de listo del equipo?							<input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Figura 35

Historia de usuario HU028 – Consulta del historial de facturación

Historias de usuario: HU028		Título: Consulta del Historial de Facturación						
<p>Como administrador, Quiero consultar el historial de facturación, Para realizar revisiones, auditorías internas o gestionar datos contables.</p>								
<p>Criterios de aceptación:</p> <ol style="list-style-type: none"> 1. El sistema debe proporcionar al administrador una vista completa de todas las facturas generadas en el sistema. 2. Esta vista debe incluir información clave por factura: Fecha de emisión, Número de factura, Subtotal, Descuento y Total. 3. El sistema ordenara las facturas por fecha (de la más reciente a la más antigua). 4. La interfaz debe cargar los datos sin demoras significativas. 5. Debe haber una opción para ver el detalle completo de una factura específica desde esta lista (ver factura en PDF en el navegador). 6. Solo los usuarios con rol de administrador deben tener acceso a esta vista. 								
I	N	V	E	S	T	Valor: 85/100		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Estimación: 4 horas		
							¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/>	

Fuente. Autoría Propia.

Figura 36

Historia de usuario HU029 – Visualización de lista de clientes con compras (Administrador)

Historias de usuario: HU029		Título: Visualización de Lista de Clientes con Compras						
<p>Como administrador, Quiero ver la lista de todos los clientes que han realizado compras en la tienda, Para consultar su historial, verificar sus datos y gestionar correctamente la relación comercial.</p>								
<p>Criterios de aceptación:</p> <ol style="list-style-type: none"> 1. El sistema debe permitir al administrador acceder a una vista de clientes registrados mediante compras realizadas. 2. La lista debe mostrar por cada cliente: Nombre completo, NIT o cédula, dirección, ciudad, teléfono y compras realizadas. 3. Deben incluirse únicamente clientes que tengan al menos una factura registrada. 4. La interfaz debe: Ser ordenada, permitir búsqueda por cedula o NIT. 5. Desde esta lista, el administrador debe poder: Acceder al detalle del cliente, consultar el numero de compras realizadas por el cliente. 6. El acceso debe estar restringido solo a usuarios con rol de administrador. 								
I	N	V	E	S	T	Valor: 80/100		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Estimación: 3 horas		
							¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/>	

Fuente. Autoría Propia.

Figura 37

Historia de usuario HU030 – Búsqueda de clientes por NIT o identificación (Administrador)

Historias de usuario: HU030	Título: Búsqueda de Clientes por NIT o Identificación (Administrador)
<p>Como administrador, Quiero poder buscar un cliente específico por su NIT o número de identificación, Para localizarlo rápidamente en la base de datos sin necesidad de recorrer toda la lista.</p>	
<p>Criterios de aceptación:</p> <ol style="list-style-type: none"> 1. El campo de búsqueda debe aceptar tanto: El NIT o número de identificación. 2. Si el campo está vacío, el sistema debe: Bloquear la búsqueda, mostrar un mensaje como "Por favor, ingrese un número de cédula o NIT para buscar". 3. Si no hay coincidencias: Debe mostrarse el mensaje "No hay clientes registrados". 4. Si hay resultados, deben mostrarse en una lista con: Nombre completo, cedula o NIT, dirección, ciudad, teléfono y numero de compras. 5. La búsqueda debe ofrecer retroalimentación rápida y correcta. 6. Solo los usuarios con rol de administrador deben tener acceso a esta funcionalidad. 	
I	Valor: 80/100
N	Estimación: 6 horas
V	¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/>
E	
S	
T	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	

Fuente. Autoría Propia.

Figura 38

Historia de usuario HU031 – Visualización del número de compras por cliente

Historias de usuario: HU031	Título: Visualización del Número de Compras por Cliente
<p>Como administrador, Quiero poder ver el número de compras realizadas por cada cliente, Para identificar a los clientes frecuentes y ofrecerles promociones especiales.</p>	
<p>Criterios de aceptación:</p> <ol style="list-style-type: none"> 1. El sistema debe verificar que el usuario tenga rol de administrador antes de permitir el acceso a esta funcionalidad. 2. El contador de compras por cliente debe considerar solo facturas finalizadas. 3. La información debe mostrarse en una tabla o lista, incluyendo: Nombre del cliente, cedula o NIT, Dirección, Ciudad, teléfono y Cantidad total de compras realizadas. 4. El sistema debe ofrecer una interfaz de carga rápida incluso con volúmenes grandes de datos. 	
I	Valor: 85/100
N	Estimación: 5 horas
V	¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/>
E	
S	
T	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	

Fuente. Autoría Propia.

Figura 39

Historia de usuario HU032 – Registro automático de cliente al realizar la primera compra

Historias de usuario: HU032	Título: Registro Automático de Cliente al Realizar la Primera Compra
<p>Como usuario de la aplicación, Quiero que el sistema registre automáticamente a un nuevo cliente cuando realiza su primera compra, Para evitar tener que ingresar sus datos en un formulario separado y agilizar el proceso de facturación.</p>	
<p>Criterios de aceptación:</p> <ol style="list-style-type: none"> 1. El sistema debe verificar que el campo de identificación (cédula o NIT) tenga el formato correcto, según el tipo seleccionado. 2. Antes de registrar al cliente, el sistema debe consultar si ya existe una identificación igual en la base de datos. 3. Si el NIT o cédula ya están registrados, el sistema debe: Asociar la venta al cliente existente, evitar crear un duplicado. 4. Si no existe, se debe registrar automáticamente un nuevo cliente con los datos ingresados en la factura. 5. Los campos obligatorios a validar son: Nombre completo, cédula o NIT, dirección, ciudad y teléfono. 6. El cliente debe quedar registrado en la base de datos al finalizar la venta, sin pasos adicionales para el vendedor. 7. Esta funcionalidad debe estar disponible solo para usuarios con rol de vendedor o superior. 	
Valor: 85/100	
Estimación: 4 horas	
I N V E S T <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	¿Cumple con la definición de listo del equipo? <input checked="" type="checkbox"/>

Fuente. Autoría Propia.

Mapa de Trazabilidad del Producto Mínimo Viable (MVP)

Este mapa vincula cada épica definida con sus historias de usuario para evidenciar el alcance mínimo funcional y facilitar la verificación de entregables durante el desarrollo.

Tabla 6

Mapa de trazabilidad del MVP: relación entre épicas e historias de usuario

MVP	Épica	Historia de Usuario
Inicio de sesión y autenticación	EP01: Inicio de sesión y autenticación	HU001: Página de Inicio con Login Integrado. HU002: Formulario de Login con redirección por rol.
Gestión de usuarios	EP02: Gestión de usuarios	HU003: Sidebar para Acceso a Funcionalidades de Gestión. HU004: Encabezado de la Aplicación Web. HU005: Página de Administración de Categorías y Productos.

		HU006: Registro de Nuevos Usuarios con Asignación de Rol.
Gestión de categorías	EP03: Gestión de categorías	HU007: Creación de Nuevas Categorías. HU008: Actualización de Categorías Existentes.
Gestión de productos	EP04: Gestión de productos	HU009: Actualización de Información de un Producto. HU010: Registro de Nuevos Productos Asociados a Categorías. HU011: Listado de Productos por Categoría. HU012: Visualización y Gestión de Productos de una Categoría Específica.
Cabeceras para facturas	EP05: Cabeceras para facturas	HU015: Creación de Cabeceras para Facturas por Sucursal. HU016: Edición de Cabeceras. HU017: Eliminación de Cabeceras Innecesarias.
Gestión de facturas	EP06: Gestión de facturas	HU020: Creación de Nueva Factura. HU021: Agregar Productos con Cantidad, Descuento e IVA a la Factura. HU022: Cálculo Automático de Totales en la Factura. HU023: Verificación Automática del Stock al Facturar. HU024: Edición de Factura Antes de Finalizar. HU025: Descarga de Factura Finalizada en PDF. HU026: Búsqueda de Facturas por Número o NIT (Administrador). HU027: Descuento Automático del Stock al Generar Factura. HU029: Visualización de Lista de Clientes con Compras (Administrador).
Gestión de clientes	EP07: Gestión de clientes	HU013: Visualización de Categorías por parte del Usuario Regular (Vendedor). HU014: Visualización de Productos por Categoría para Vendedores. HU018: Búsqueda de Cabeceras por NIT en Gestión Administrativa. HU019: Selección de Cabecera al Crear una Factura. HU028: Consulta del Historial de Facturación.

HU030: Búsqueda de Clientes por NIT
o Identificación (Administrador).
HU031: Visualización del Número de
Compras por Cliente.
HU032: Registro Automático de
Cliente al Realizar la Primera Compra.

Nota. La tabla relaciona, por módulo del MVP, cada épica (EP01–EP07) con las historias de usuario (HU001–HU032) que la implementan.

Diseño de la Arquitectura del Sistema

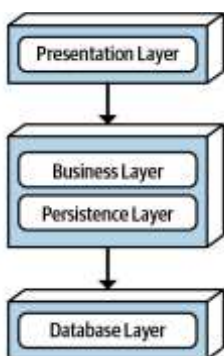
El diseño arquitectónico del software es una etapa clave en el desarrollo de soluciones tecnológicas. La diferencia entre un diseño bien estructurado y uno deficiente puede marcar el éxito o el fracaso de un proyecto. Esto se debe a que, más allá de satisfacer los requisitos funcionales, también debe incorporar los atributos de calidad, que responden a las necesidades no funcionales del sistema (González Clavijo et al., 2022). En este sentido, la arquitectura establece la estructura sobre la cual se construyen los componentes del sistema, garantizando propiedades como escalabilidad, mantenibilidad e integración futura. En este apartado se documenta la arquitectura propuesta para un prototipo web de gestión de inventarios y ventas, desarrollado para la empresa ADIIC Dotaciones Institucionales, empleando la pila tecnológica MERN y guiado por la metodología ágil SCRUM.

Tipo de Arquitectura Aplicada

La arquitectura seleccionada adopta un modelo cliente-servidor de tres capas, adecuado para aplicaciones web que requieren modularidad, separación de responsabilidades y una estructura organizada de comunicación entre subsistemas (Richards & Ford, 2020).

Figura 40

Modelo cliente-servidor de tres capas



Fuente. Tomado de *Fundamentals of Software Architecture* (p. 139), por M. Richards y N. Ford, 2020, O'Reilly Media. Copyright 2020 por Mark Richards y Neal Ford.

Las tres capas identificadas son:

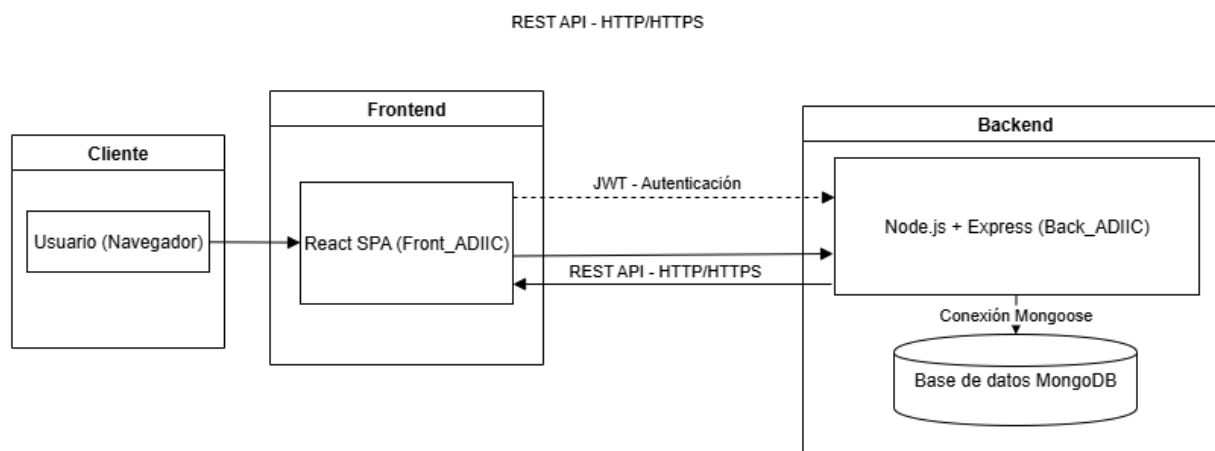
- Capa de presentación: desarrollada en React.js, permite la interacción del usuario a través de una interfaz dinámica, responsiva y desacoplada del backend.
- Capa de lógica de negocio: implementada con Node.js y Express, se encarga del procesamiento de datos, la gestión de reglas de negocio y la orquestación del flujo de información entre la presentación y la base de datos.
- Capa de base de datos y persistencia: basada en MongoDB, almacena la información en documentos estructurados bajo el modelo NoSQL, lo cual facilita consultas eficientes y favorece la escalabilidad horizontal.

La comunicación entre capas se realiza mediante APIs RESTful que operan bajo los protocolos HTTP/HTTPS, con mecanismos de autenticación basados en JSON Web Tokens (JWT), garantizando la integridad, confidencialidad y seguridad del intercambio de información (Bass et al., 2021).

Dentro de la capa de lógica de negocio, se aplicó el patrón de diseño Modelo-Vista-Controlador (MVC), estructurando el backend en módulos de rutas (controladores), lógica de negocio y modelos de datos definidos mediante Mongoose. Esta decisión favorece la mantenibilidad del sistema y refuerza la separación de responsabilidades internas, complementando adecuadamente la arquitectura en capas adoptada (Pinzón Núñez et al., 2019). Lo anteriormente explicado lo podemos observar gráficamente en la figura 41:

Figura 41

Diagrama de arquitectura del sistema



Fuente. Autoría Propia.

Selección del Modelo de Vista de Arquitectura

Para capturar de forma exhaustiva los distintos aspectos de la arquitectura del sistema, se adoptó el modelo de viewpoints de Rozanski & Woods (2012). Este enfoque estructurado define seis perspectivas (*viewpoints*), cada una orientada a resolver las preocupaciones de grupos de interesados específicos. Con ello se logra:

- Clarificar los límites y el contexto de operación del sistema.
- Visualizar la funcionalidad y el flujo de información.
- Analizar el comportamiento concurrente y las dependencias de ejecución.
- Describir la organización del código y sus dependencias.
- Documentar la asignación de componentes en la infraestructura de despliegue.

De este modo, se potencia la comunicación entre el equipo técnico y las demás partes interesadas, y se garantiza la trazabilidad entre requisitos, decisiones de diseño y despliegue.

Siguiendo el enfoque multivista propuesto por Rozanski & Woods (2012), se seleccionaron siete vistas arquitectónicas: contexto, información, despliegue, funcional, desarrollo, concurrencia y seguridad y autenticación.

Tabla 7

Descripción de vistas arquitectónicas adoptadas

Punto de Vista	Descripción	Interesados	Representación
Contexto	Define el alcance y los límites del sistema, y muestra sus interacciones con actores y sistemas externos.	Patrocinadores; usuarios finales; arquitectos	Diagrama de contexto (System Context)
Funcional	Descompone el sistema en responsabilidades y servicios principales, organizando la funcionalidad en módulos cohesionados.	Arquitectos de software; analistas de negocio	Diagrama de componentes
Información	Modela las estructuras de datos, las relaciones entre entidades y las reglas de persistencia y transformación de la información.	Diseñadores de bases de datos; integradores de datos	Diagrama de clases orientado a datos
Concurrencia	Expone los procesos, hilos de ejecución y mecanismos de sincronización, así como los patrones de comunicación y colaboración en tiempo real.	Arquitectos de rendimiento; ingenieros de pruebas	Diagrama de secuencia; diagrama de actividad
Desarrollo	Frontend – Estructura generada con Create React App: public/ (assets estáticos, index.html) y src/ (punto de entrada index.js, componentes, páginas y estilos).– Librerías principales: React 18, React Router DOM, Axios, React Hook Form + Yup, JWT-Decode, React-Modal, jsPDF + jsPDF-AutoTable, Lucide-React y SweetAlert; estilado con Tailwind CSS. Backend – Servidor Express: arranque en index.js, configuración de express.json() y CORS, conexión a MongoDB Atlas (config/db.js).– Rutas REST en /routes; controladores en /controllers; modelos Mongoose en /models	Desarrolladores frontend y backend	Diagrama de paquetes (rutas, controladores, modelos); estructura de carpetas

	(Usuarios, Categorías, Productos, Clientes, Facturas, etc.).– Utilidades: bcryptjs, jsonwebtoken, dotenv, moment-timezone, decimal.js, uuid.		
Despliegue	<p>Frontend: compilación con npm run build → build/, apto para hosting estático (Vercel, Netlify, S3, etc.).</p> <p>Backend: Node.js en puerto 4000, conectado a MongoDB Atlas; configuración mediante variables.env (URI, claves JWT).–</p> <p>Contenerización opcional con Docker y pipelines de CI/CD en GitHub Actions para despliegue automático.– Topología: cliente React ↔ API Express ↔ MongoDB Atlas.</p>	Administradores de sistemas; equipo de infraestructura	Diagrama de despliegue
Seguridad	<p>Define los controles para proteger la confidencialidad, integridad y disponibilidad de la información y gobernar el acceso al sistema. Incluye: autenticación con JWT de vida corta (entregado preferiblemente en cookie HttpOnly/Secure/SameSite), autorización por roles (RBAC) validada en middleware del backend, cifrado en tránsito (TLS) y cifrado selectivo en reposo, gestión segura de secretos (.env/servicio de secretos), validación y sanitización de entradas, CORS por lista blanca, cabeceras de seguridad (CSP, HSTS, X-Content-Type-Options, X-Frame-Options, Referrer-Policy), auditoría de eventos críticos y respaldos con objetivos de continuidad (RPO ≤ 24 h, RTO ≤ 4 h). Alinea la implementación con el marco de protección de datos personales (Ley 1581/2012) y la trazabilidad exigida por facturación electrónica.</p>	Dirección; Responsable/Oficial de Protección de Datos; Arquitectos de software; Desarrolladores (frontend/backend); DevOps/Infraestructura; Auditoría/Calidad; Usuarios administradores.	Diagrama de secuencia (UML 2.5) de autenticación y autorización (login → emisión JWT → verificación RBAC → logout/blacklist); diagrama de componentes de seguridad (Frontend, API/Auth, JWTService, UsuarioDB, TokenBlacklist); matriz de riesgos y mitigaciones (OWASP Top 10) y referencia a la matriz de datos personales (Marco legal).

Nota. Tabla adaptada de Rozanski & Woods (2012) e implementaciones reales verificadas en los repositorios *Front_ADIIC* y *Back_ADIIC*.

Esta forma de estructurar la arquitectura mediante vistas específicas permite describir de manera clara y organizada la composición del sistema, facilitando a su vez la comunicación entre los distintos actores del proyecto. Gracias a la utilización de representaciones visuales adaptadas a las necesidades de cada grupo de interés, se logra una comprensión más efectiva de la solución propuesta, en coherencia con el enfoque orientado a stakeholders planteado por Rozanski & Woods (2012).

Selección de Puntos de Vista para la Arquitectura

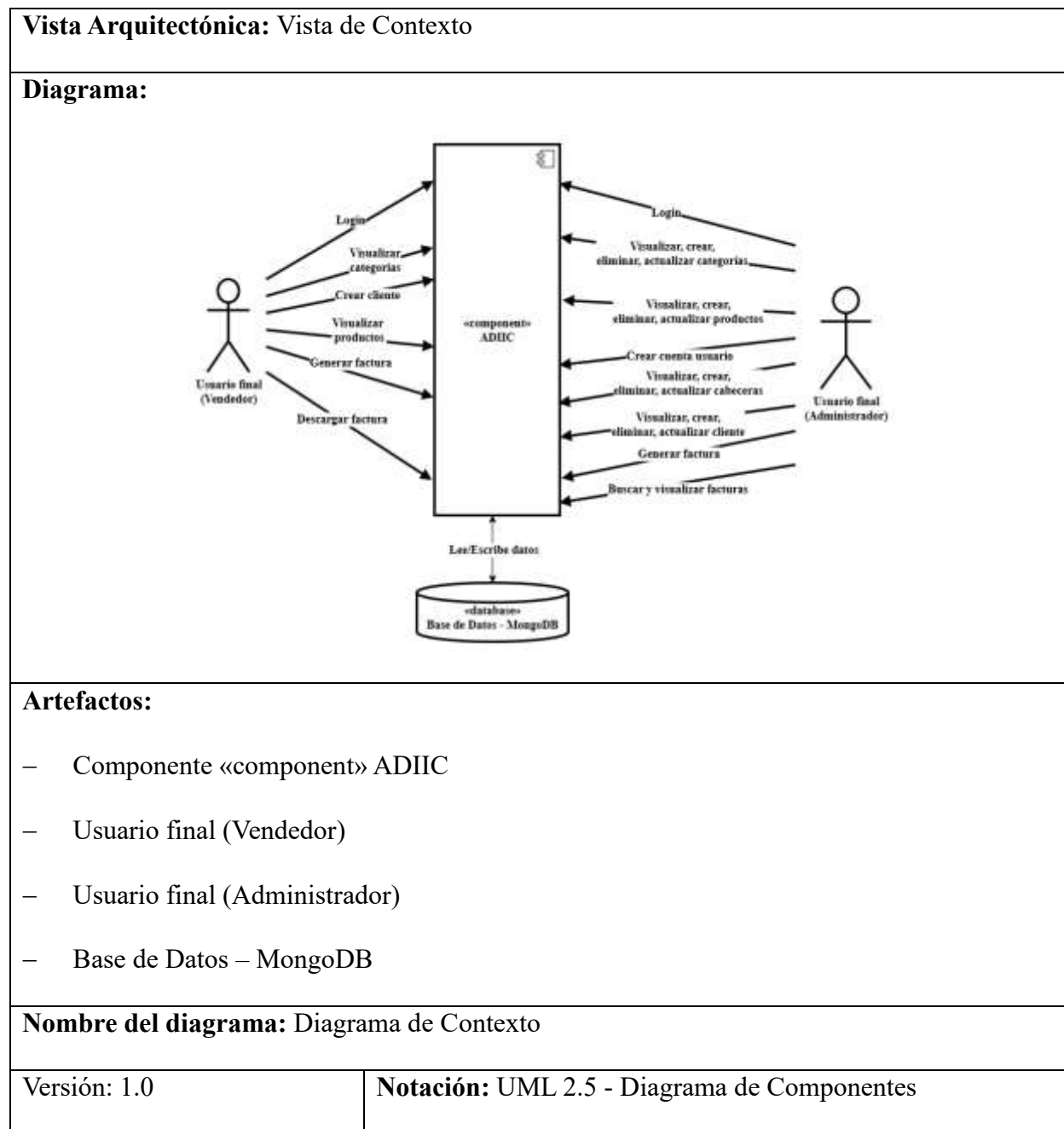
Siguiendo el enfoque multivista propuesto por Rozanski & Woods (2012), se seleccionaron seis vistas arquitectónicas para documentar la solución propuesta: vista de contexto, funcional, información, concurrencia, desarrollo y despliegue. Esta selección responde a la necesidad de representar tanto los aspectos estructurales como los comportamientos dinámicos del sistema, garantizando una comprensión integral del diseño. Las vistas fueron elegidas en función de las principales preocupaciones de los diferentes interesados, incluyendo desarrolladores, arquitectos, usuarios técnicos y personal de infraestructura, con el objetivo de facilitar una comunicación clara y efectiva entre ellos. Aunque el modelo contempla una séptima vista dedicada a escenarios, esta no fue considerada en el desarrollo del presente proyecto, dado que no se realizó una evaluación formal de la arquitectura mediante casos de uso específicos o escenarios de calidad. La documentación arquitectónica se centró en aquellas vistas que permitieran representar de forma clara las decisiones estructurales y comportamentales relevantes para los interesados.

Vista de Contexto

El diagrama fue elaborado con la herramienta Draw.io, La vista de contexto que se muestra al final muestra una representación de alto nivel del sistema desarrollado, denominado Sistema ADIIC, en su entorno operativo. En total, el diagrama está compuesto por cinco elementos principales: dos actores externos (Usuario final administrador y vendedor), el sistema central (Sistema ADIIC), una base de datos (MongoDB) y las interacciones entre ellos representadas por flechas etiquetadas.

- En el lado izquierdo se encuentra el Usuario final, quien puede realizar múltiples interacciones con el sistema:
 - Solicitud de ingreso, consulta de inventario y registro de ventas, representadas por flechas dirigidas hacia el sistema.
 - Solicitud de factura (línea discontinua hacia el sistema) y entrega de factura (línea discontinua desde el sistema), indicando un intercambio bidireccional relacionado con la emisión de comprobantes de venta.
- En el lado derecho se encuentra el Administrador, quien se comunica con el sistema mediante la acción Administra configuración, lo cual alude a la gestión de parámetros y datos administrativos del sistema.
- En la parte inferior, se representa la Base de Datos - MongoDB, indicando que el sistema lee y escribe datos de forma directa sobre esta fuente de persistencia.
- El elemento central, el Sistema ADIIC, encapsula toda la lógica del sistema y actúa como intermediario entre los usuarios y la base de datos.

Este diagrama establece los límites del sistema y sus relaciones clave con los actores externos, permitiendo a los interesados entender de forma rápida quiénes interactúan con el sistema, cómo lo hacen, y qué tipo de datos se gestionan.



Autor(es): Juan Carlos Arrieta Bustos Richard Pardo Cardona	Fecha de creación: _10 / 01 / 2025_
--	---

Vista de Información

El diagrama fue elaborado con la herramienta Draw.io, La vista de información está compuesta por ocho entidades principales, cada una representando una colección del sistema implementado con MongoDB. Estas entidades están organizadas y conectadas mediante relaciones de tipo 1:N (uno a muchos), lo que permite ilustrar las dependencias lógicas entre los distintos tipos de datos gestionados por el sistema.

Las entidades modeladas son: Usuarios, Clientes, Categorías, Cabeceras, Productos, Facturas, CuerpoFacturas, Relaciones entre entidades (1:N y N:1).

La entidad Usuarios contiene los campos: nombre, email, password y rol. Esta entidad se relaciona con:

- Clientes, mediante el campo creador_id, ya que cada cliente registrado está vinculado a un usuario del sistema.
- Cabeceras y Categorías, que también son gestionadas por usuarios administradores.

La entidad Clientes almacena datos como nombre, nit, dirección, ciudad, teléfono, número de compras, entre otros.

- Se relaciona con Facturas, ya que un cliente puede tener múltiples facturas asociadas (relación 1:N).

Cabeceras representa la información de la empresa que emite las facturas (local, NIT, dirección, etc.), y se relaciona también con Facturas, ya que una cabecera puede usarse para múltiples documentos emitidos.

Categorías agrupa a los productos y contiene un campo nombre y imagen.

- La relación con Productos es 1:N, ya que una categoría puede tener múltiples productos.

La entidad Productos incluye campos como nombre, descripción, stock, precio, imagen, talla, color, referencia, entre otros.

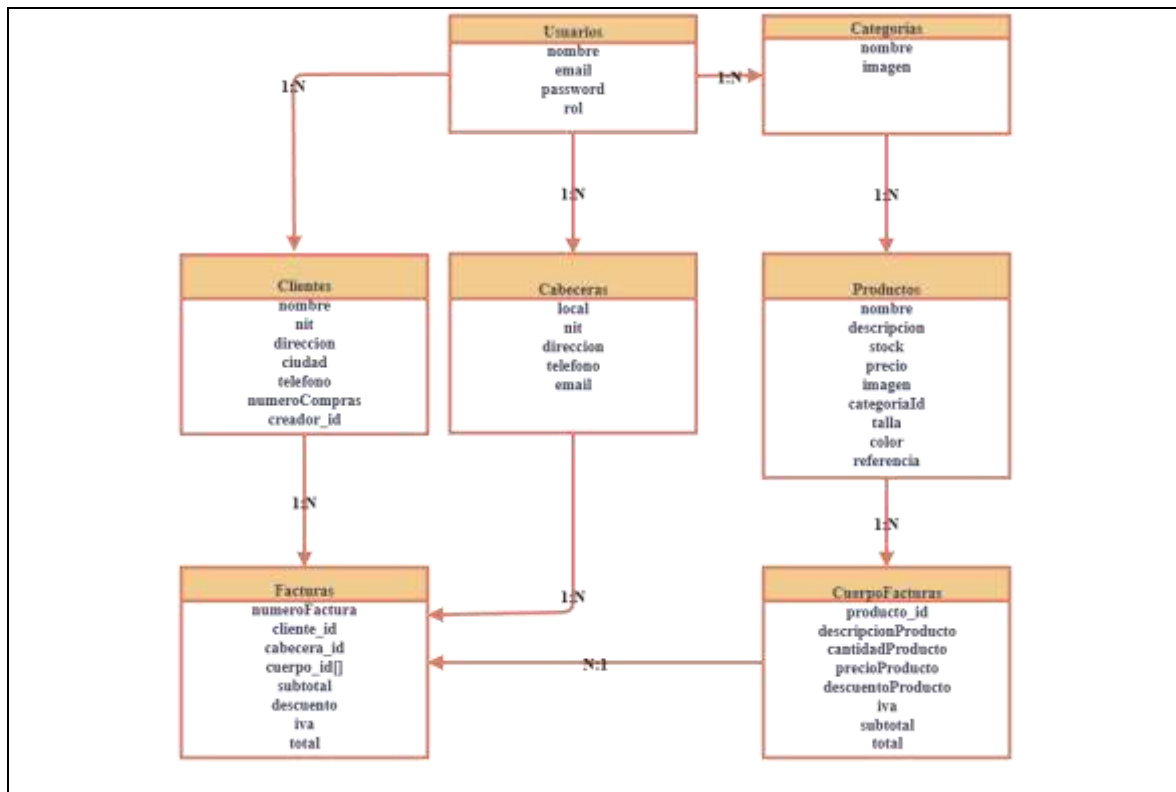
- Está relacionada con CuerpoFacturas, ya que un producto puede formar parte de múltiples detalles de factura.

Facturas representa la cabecera de cada comprobante de venta. Contiene campos como numeroFactura, cliente_id, cabecera_id, cuerpo_id[], subtotal, descuento, iva, total.

Finalmente, CuerpoFacturas representa el detalle de los productos vendidos en cada factura, incluyendo producto_id, cantidadProducto, precioProducto, descuentoProducto, subtotal, iva y total.

Este modelo de datos refleja una estructura documental y relacional orientada a MongoDB, en la que se conservan las dependencias necesarias para mantener la integridad de la información y garantizar una trazabilidad adecuada de los procesos de inventario, facturación y gestión de clientes.

Vista Arquitectónica: Vista de Información
Diagrama:



Artefactos:

- package.json y package-lock.json (manifiestos de dependencias)
- variables.env (variables de entorno: URI, credenciales, puertos)
- index.js (punto de entrada, conexión BD y arranque servidor)
- Carpeta config (archivos de configuración del proyecto)
- Carpeta models (definiciones de esquemas Mongoose: Usuario.js, Producto.js, Factura.js, Cliente.js, Categoria.js, Cabecera.js, CuerpoFactura.js)
- Carpeta controllers (lógica CRUD)
- Carpeta routes (routers y endpoints)
- Carpeta middleware (JWT, validaciones, manejo de errores)
- Carpeta node_modules (dependencias instaladas tras npm install)

<ul style="list-style-type: none"> – ModeloMongoDB_ADIIC_completo_con_usuarios.drawio (diagrama UML de clases) – plantilla_para_vistas.docx (plantilla de la vista) 	
Nombre del diagrama: Vista de Información del modelo de datos MongoDB para ADIIC Dotaciones Institucionales	
Versión: 1.0	Notación: UML 2.5 – Diagrama de clases
Autor(es): Juan Carlos Arrieta Bustos Richard Pardo Cardona	Fecha de creación: _10 / 01 / 2025_

Vista de Despliegue

El diagrama fue elaborado con draw.io, esta vista muestra cómo se distribuyen físicamente los componentes del sistema ADIIC en su infraestructura de ejecución, especificando las plataformas de despliegue y los canales de comunicación entre ellos.

El diagrama está compuesto por cuatro elementos principales: Frontend (Vercel), Backend (Node.js + Express) (Railway), Base de datos (MongoDB Atlas), Variables de entorno encriptadas.

Las interacciones entre estos elementos están representadas mediante flechas con la etiqueta HTTPS/TLS, lo que indica que toda la comunicación entre componentes ocurre a través de conexiones seguras.

De izquierda a derecha, el diagrama inicia con el Frontend, que está desplegado en la plataforma Vercel. Esta interfaz, desarrollada con React.js, representa el punto de acceso para los usuarios del sistema desde sus navegadores.

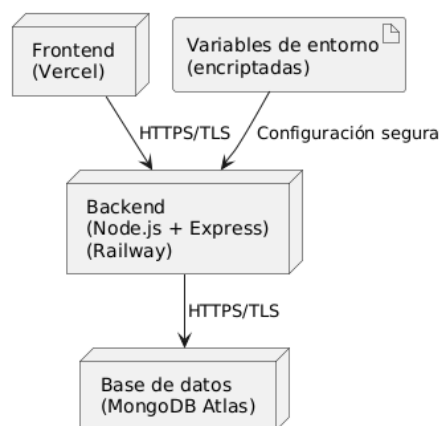
Desde el frontend, se establece una conexión segura hacia el Backend, que ha sido desarrollado con Node.js y Express, y se encuentra alojado en Railway. Este módulo intermedio contiene la lógica del negocio, así como los endpoints que permiten gestionar usuarios, productos e interacciones del sistema.

El backend, a su vez, se conecta con la Base de datos MongoDB, la cual está desplegada en la nube mediante el servicio MongoDB Atlas. En esta base de datos se almacena de forma persistente la información central del sistema.

Finalmente, el diagrama incluye un nodo adicional para las Variables de entorno encriptadas, que alimentan al backend con configuraciones sensibles como cadenas de conexión, claves secretas y rutas protegidas. Estas variables son gestionadas de forma segura desde la plataforma Railway.

Vista Arquitectónica: Vista de Despliegue

Diagrama:



Artefactos:

- Frontend Application (artefacto desplegado en Vercel)
- Backend Application (artefacto Node.js + Express desplegado en Railway)

<ul style="list-style-type: none"> – Variables de entorno (archivo cifrado de configuración) – Base de datos (instancia de MongoDB Atlas) 	
Nombre del diagrama: Diagrama de Despliegue del Sistema	
Versión: 1.0	Notación: UML 2.5 – Diagrama de Despliegue
Autor(es): Juan Carlos Arrieta Bustos Richard Pardo Cardona	Fecha de creación: _10 / 01 / 2025_

Vista Funcional

La vista funcional fue elaborada mediante la herramienta Draw.io, El diagrama funcional del sistema está compuesto por tres secciones principales:

- Frontend (parte superior) con nueve componentes React.
- Backend (parte inferior) con seis controladores Node.js/Express.
- Elementos de soporte, como la verificación JWT y la configuración de base de datos.

En total, el diagrama cuenta con 17 bloques funcionales, interconectados mediante líneas que representan las rutas de interacción (endpoints API) entre el cliente y el servidor.

En la parte superior, bajo el título Frontend, se representan los componentes del sistema implementados en React. Entre ellos se encuentran:

- Componentes de navegación y estructura: Navbar, Sidebar, Dashboard.
- Componentes funcionales: Login, Lista de Clientes, Lista de Productos, Lista de Categorías, CabeceraModal, FacturaModal.

Cada uno de estos componentes se comunica con el backend a través de endpoints

RESTful:

- /api/auth desde Login.
- /api/clientes desde Lista de Clientes.
- /api/productos desde Lista de Productos.
- /api/categorias desde Lista de Categorías.
- /api/cabecera desde CabeceraModal.
- /api/factura desde FacturaModal.

En el Backend, cada ruta está gestionada por un controlador correspondiente desarrollado con Node.js y Express:

- AuthController maneja la autenticación.
- ClientController, ProductController, CategoryController, CabeceraController y FacturaController gestionan las entidades funcionales del sistema.

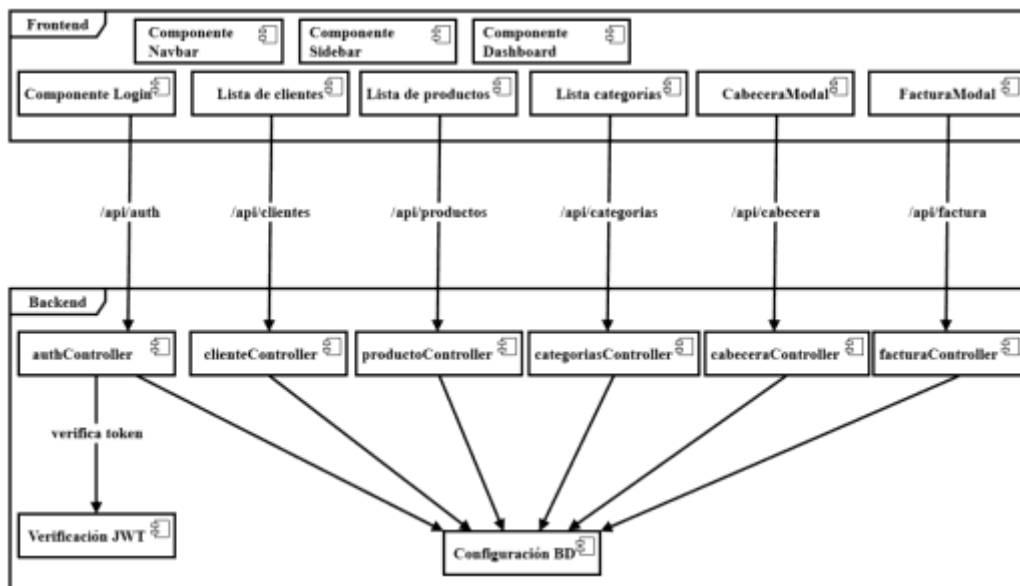
Adicionalmente, se incluye un componente para la Verificación JWT, que actúa como middleware de seguridad, especialmente conectado al flujo de autenticación.

Todos los controladores dependen de la Configuración de Base de Datos, que representa la conexión y manejo centralizado con MongoDB.

Este diagrama refleja claramente la separación de responsabilidades entre el cliente (frontend) y el servidor (backend), así como la forma en que los datos fluyen entre componentes a través de endpoints RESTful, garantizando un diseño modular, escalable y mantenible.

Vista Arquitectónica: Vista funcional

Diagrama:



Artefactos:

- authController.js (módulo de autenticación)
- clienteController.js (gestión de clientes)
- productoController.js (gestión de productos)
- categoriasController.js (gestión de categorías)
- cabeceraController.js (módulo del modal de cabecera)
- facturaController.js (módulo del modal de factura)
- verificacionJWT.js (middleware de verificación de token JWT)
- configBD.js (archivo de configuración de conexión a MongoDB)

Nombre del diagrama: Diagrama de la Vista Funcional

Versión: 1.0

Notación: UML 2.5.1 - Diagrama de Componentes

Autor(es): Juan Carlos Arrieta Bustos Richard Pardo Cardona	Fecha de creación: _10 / 01 / 2025_
--	---

Vista de Desarrollo

La Vista de Desarrollo presenta la estructura modular del sistema ADIIC, diseñado con la pila MERN (MongoDB, Express.js, React.js y Node.js). Esta vista permite comprender la organización interna del código fuente, así como las relaciones entre los módulos que conforman el frontend y el backend del sistema.

El frontend fue desarrollado con React.js, y se estructura de la siguiente forma:

- public/: contiene archivos estáticos como index.html, el logo del sistema y configuraciones del manifiesto (manifest.json).
- src/: agrupa los archivos principales del núcleo de la aplicación (App.js, index.js, App.css, entre otros).
- src/componentes/: organiza los distintos módulos funcionales en subcarpetas:
 - cabeceras/: componentes relacionados con la cabecera del sistema (Cabecera.js, CabeceraModal.js, crearCabecera.js).
 - categorias/: componentes para la gestión de categorías (ActualizarCategoria.js, CrearCategoria.js).
 - clientes/: incluye interfaces para clientes (Clientes.js, ClienteModal.js).
 - facturas/: incluye la lógica para la creación y visualización de facturas (Factura.js, FacturaModal.js).

- productos/: gestiona los productos del sistema (ActualizarProducto.js, CrearProducto.js, HomeProductos.js, ViewProductos.js, RegularProductos.js).
- admin/: funcionalidades específicas de administración (Admin.js, CrearCuenta.js).
- Componentes generales de navegación e interfaz (Header.js, Sidebar.js, Login.js, Home.js, Regular.js).
- src/conexiones/: módulo encargado de gestionar las conexiones con la API RESTful del backend.

El backend fue construido con Node.js y Express.js, y sigue un patrón estructurado en capas:

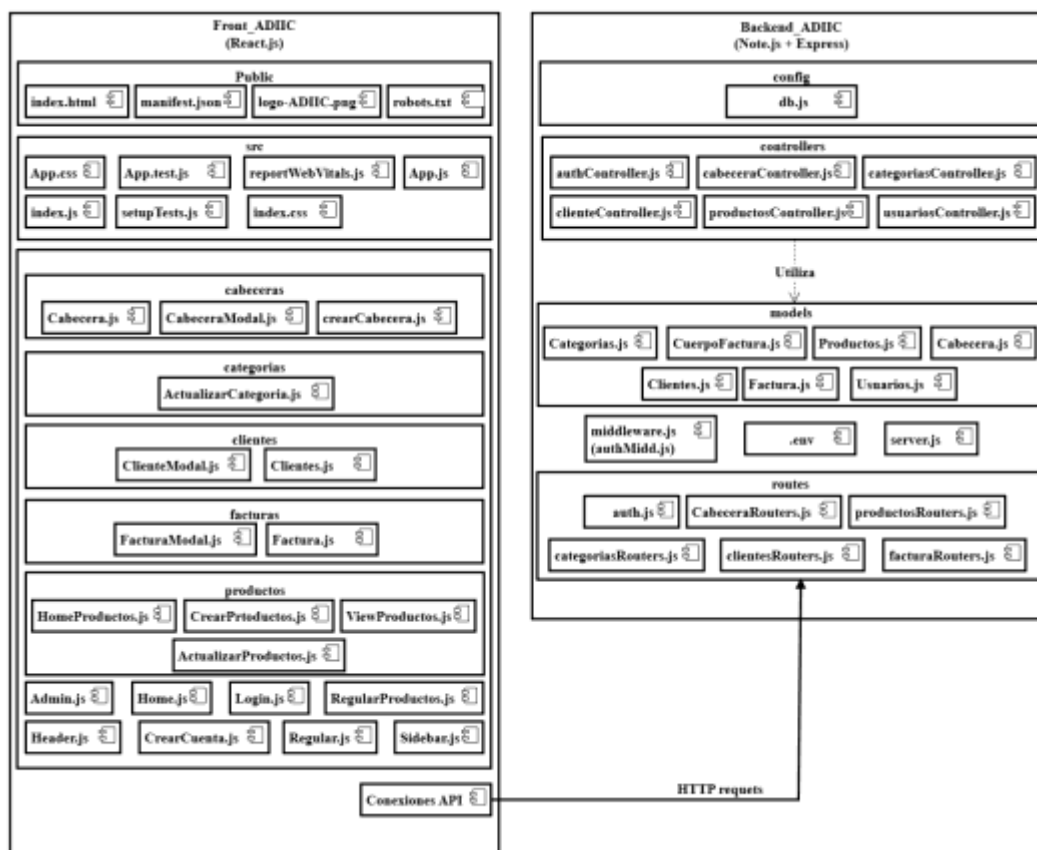
- config/: contiene la configuración de la base de datos (db.js).
- models/: define los modelos de datos usados en MongoDB mediante Mongoose (Cabecera.js, Categorías.js, Clientes.js, CuerpoFactura.js, Factura.js, Productos.js, Usuarios.js).
- controllers/: agrupa la lógica de negocio y controla las operaciones sobre los modelos (authController.js, cabeceraController.js, categoriasController.js, clienteController.js, facturaController.js, productosController.js, usuariosController.js).
- routes/: contiene las rutas expuestas por la API (auth.js, cabeceraRouters.js, categoriasRouters.js, clientesRouters.js, productosRouters.js, facturaRouters.js), las cuales comunican el frontend con los controladores correspondientes.
- middleware/: incluye funcionalidades intermedias como la autenticación (authMidd.js).
- server.js: archivo principal que inicia el servidor Express y configura las rutas y middlewares.

- `variables.env`: archivo de entorno que almacena credenciales y configuraciones sensibles.

Esta organización modular permite una clara separación de responsabilidades y facilita el mantenimiento y la escalabilidad del sistema.

Vista Arquitectónica: Vista de Desarrollo

Diagrama:



Artefactos:

Front_ADIIC (React.js)

- *Public*: `index.html`, `manifest.json`, `logo-ADIIC.png`, `robots.txt`
- *src*: `App.css`, `App.test.js`, `reportWebVitals.js`, `App.js`, `index.js`, `setupTests.js`, `index.css`
- *src/componentes*

- cabeceras: Cabecera.js, CabeceraModal.js, crearCabecera.js
- categorias: ActualizarCategoria.js
- clientes: ClienteModal.js, Clientes.js
- facturas: FacturaModal.js, Factura.js
- productos: HomeProductos.js, CrearPrductos.js, ActualizarProductos.js, ViewProductos.js
- UI genérica: Admin.js, Home.js, Login.js, RegularProductos.js, Header.js, CrearCuenta.js, Regular.js, Sidebar.js
- Conexiones API: peticiones HTTP al backend

Backend_ADIIC (Node.js + Express)

- config: db.js
- controllers: authController.js, cabeceraController.js, categoriasController.js, clienteController.js, productosController.js, usuariosController.js
- models: Categorias.js, CuerpoFactura.js, Productos.js, Cabecera.js, Clientes.js, Factura.js, Usuarios.js
- middleware: authMidd.js
- .env
- server.js
- routes: auth.js, CabeceraRouters.js, productosRouters.js, categoriasRouters.js, clientesRouters.js, facturaRouters.js
- Conexiones API: recepción de peticiones HTTP desde el frontend

Nombre del diagrama: Diagrama de la Vista de Desarrollo

Versión: 1.0

Notación: UML 2.5.1 - Diagrama de Componentes

Autor(es): Juan Carlos Arrieta Bustos Richard Pardo Cardona	Fecha de creación: _10 / 01 / 2025_
--	---

Vista de Concurrencia

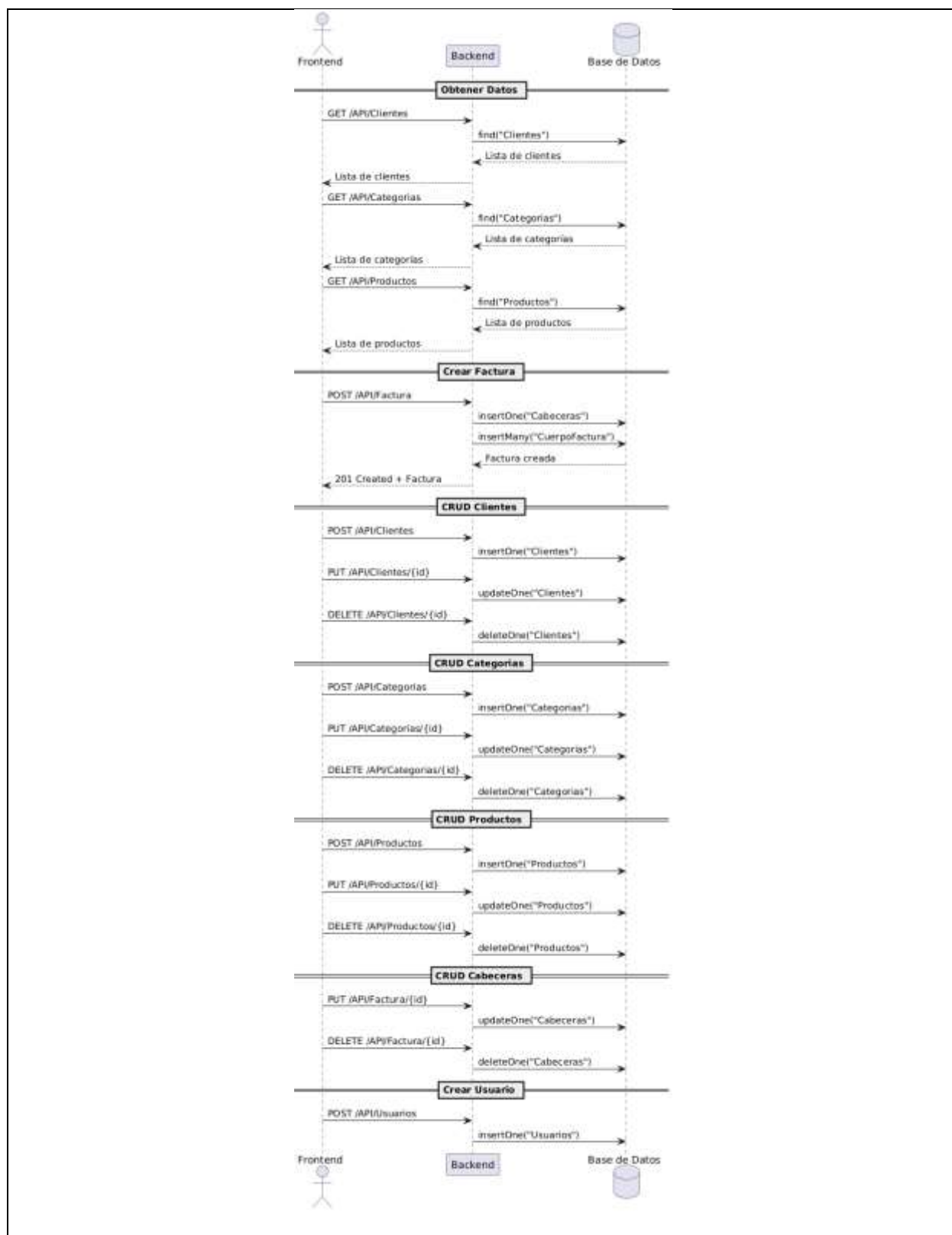
El diagrama de secuencia se elaboró con Draw.io, empleando la notación UML para representar las interacciones concurrentes entre los tres componentes principales del sistema: Frontend, Backend y Base de Datos (MongoDB). Consta de tres líneas de vida y agrupa siete bloques funcionales, con más de treinta interacciones (solicitudes HTTP y operaciones en la base de datos) que describen el flujo completo de las distintas operaciones:

- Obtener Datos
 - El Frontend realiza tres solicitudes GET (/API/Clientes, /API/Categorias, /API/Productos) al Backend.
 - El Backend invoca find("Clientes"/"Categorias"/"Productos") en MongoDB y, tras recibir cada lista, la reenvía al Frontend.
- Crear Factura
 - El Frontend envía un POST a /API/Factura.
 - El Backend persiste primero la cabecera con insertOne("Cabeceras") y luego los detalles con insertMany("CuerpoFactura").
 - Finalmente retorna un 201 Created y la entidad factura generada al Frontend.
- CRUD Clientes
 - POST /API/Clientes → insertOne("Clientes")

- PUT /API/Clientes/{id} → updateOne("Clientes")
- DELETE /API/Clientes/{id} → deleteOne("Clientes")
- CRUD Categorías
 - POST /API/Categorias → insertOne("Categorias")
 - PUT /API/Categorias/{id} → updateOne("Categorias")
 - DELETE /API/Categorias/{id} → deleteOne("Categorias")
- CRUD Productos
 - POST /API/Productos → insertOne("Productos")
 - PUT /API/Productos/{id} → updateOne("Productos")
 - DELETE /API/Productos/{id} → deleteOne("Productos")
- CRUD Cabeceras
 - PUT /API/Factura/{id} → updateOne("Cabeceras")
 - DELETE /API/Factura/{id} → deleteOne("Cabeceras")
- Crear Usuario
 - POST /API/Usuarios → insertOne("Usuarios")

Este diagrama refleja el comportamiento asíncrono y no bloqueante característico de la pila MERN (MERN stack), garantizando eficiencia en el procesamiento de múltiples solicitudes y consistencia de los datos en escenarios de alta concurrencia.

Vista Arquitectónica: Vista de Concurrencia
Diagrama:



Artefactos:	
<ul style="list-style-type: none"> – Interfaz Web (Frontend): cliente HTTP construido en React.js que inicia las solicitudes al servidor. – API REST (Backend): servicio en Node.js con Express que recibe peticiones y aplica la lógica de negocio. <p>MongoDB (Base de Datos): instancia de MongoDB Atlas donde se almacenan y recuperan los datos de clientes, categorías, productos, facturas y usuarios.</p>	
Nombre del diagrama:	
Diagrama de Secuencia de Concurrencia del Sistema ADIIC	
Versión: 1.0	Notación: UML 2.5.1 – Diagrama de Secuencia
Autor(es):	Fecha de creación:
Juan Carlos Arrieta Bustos	_10_ / _01_ / _2025_
Richard Pardo Cardona	

Vista de Seguridad y Autenticación

Describir las decisiones y controles de seguridad que protegen la confidencialidad, integridad y disponibilidad de la información, así como la autenticación y autorización de usuarios en el sistema. Esta vista complementa las vistas de Despliegue, Funcional y Concurrencia, y materializa las exigencias del Marco Legal (protección de datos, facturación y trazabilidad).

Alcance y decisiones arquitectónicas

- Principios: mínimo privilegio, defensa en profundidad, separación de responsabilidades, “deny by default”.

- Modelo de control de acceso: RBAC (roles Administrador y Vendedor) con verificación en backend para cada endpoint.
- Gestión de secretos: variables sensibles en configuración segura (.env/servicio de secretos); prohibida su exposición en el cliente.

Autenticación

- Mecanismo: JWT de vida corta emitido por el backend tras validar credenciales.
- Transporte recomendado: cookie HttpOnly, Secure, SameSite=Lax/Strict para mitigar XSS/CSRF; alternativa con almacenamiento seguro + CSP estricta si se usa localStorage.
- Rotación y revocación: lista de revocación/log-out efectivo; rotación en eventos sensibles (cambio de rol, recuperación de contraseña).
- Claves: contraseñas hash con bcrypt/argon2 y factor de costo apropiado.

Autorización

- Guardas por rol (middleware) antes de ejecutar lógica de negocio.
- Segmentación de permisos:
 - Administrador: CRUD de usuarios, categorías, productos y cabeceras; consulta de facturas y clientes.
 - Vendedor: creación de facturas; consulta de catálogo y su propio historial.
- Trazabilidad: todas las operaciones sensibles generan logs (usuario, IP, timestamp, acción).

Protección de datos y comunicaciones

- En tránsito: TLS obligatorio entre cliente-servidor y con servicios externos.
- En reposo: cifrado selectivo de campos sensibles; copias de seguridad cifradas; control de acceso a la base de datos.
- CORS: lista blanca de orígenes permitidos; bloquear por defecto.

- Cabeceras de seguridad: CSP, HSTS, X-Content-Type-Options, X-Frame-Options, Referrer-Policy.
- Validación de entradas: sanitización, límites de tamaño, tipos MIME controlados y consultas parametrizadas/ODM.
- Anti-abuso: rate limiting en endpoints de autenticación/búsqueda; bloqueo progresivo por intentos fallidos.

Registro, monitoreo y respuesta

- Auditoría: intentos fallidos de login, cambios de rol, creación/eliminación de cabeceras, operaciones masivas.
- Retención: 12–24 meses para logs de seguridad, con acceso restringido; sincronización de tiempo (NTP).
- Incidentes: plan de respuesta con clasificación, contención, erradicación, recuperación y lecciones aprendidas.

Respaldo y continuidad del negocio

- Backups: copias diarias completas y diferenciales según volumen; prueba de restauración mensual.
- Objetivos: RPO \leq 24 h, RTO \leq 4 h para servicios críticos.
- Conservación documental: facturación y soportes según norma fiscal y política contable (mín. 5–10 años).

Gestión de vulnerabilidades (OWASP Top 10 – extracto)

- Control de acceso roto (A01): pruebas unitarias de guardas; revisión de rutas administrativas.
- Fallos criptográficos (A02): TLS estricto; hashes robustos; gestión de llaves.

- Inyección (A03): validación/sanitización sistemática; CSP; consultas parametrizadas/ODM.
- Mala configuración (A05): CORS por whitelist; cabeceras de seguridad; hardening del servidor.
- Identificación/autenticación (A07): expiración corta de tokens; rotación/blacklist; 2FA opcional para administradores.
- Logging/monitorización (A09): bitácora de seguridad con alertas y retención definida.

Pruebas de seguridad y trazabilidad con el MVP

- Unitarias/Integración: casos positivos y negativos (credenciales inválidas, campos obligatorios, stock negativo, etc.).
- UAT: validación de flujos por rol (login, gestión, facturación) con evidencia de capturas y resultados esperados.
- Trazabilidad: relación HU ↔ módulo ↔ prueba documentada en Resultados (cuadros de cumplimiento).

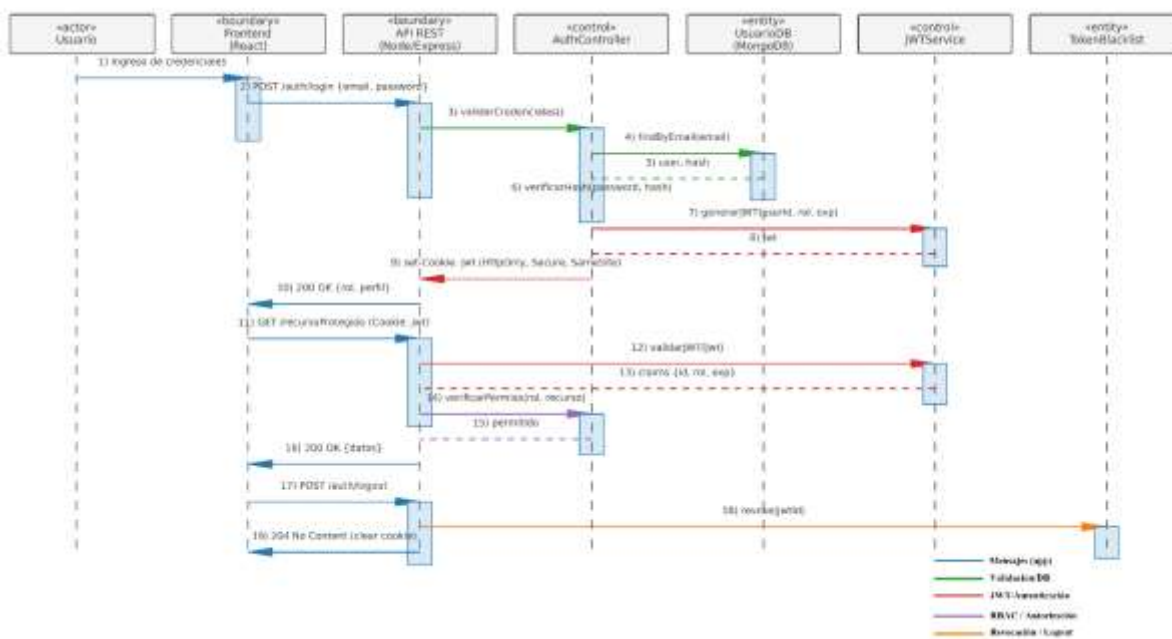
Cumplimiento y referencias cruzadas

Esta vista se alinea con el Marco Legal (Protección de Datos, Facturación Electrónica y conservación documental) y con las vistas Funcional, Despliegue y Concurrencia. La Matriz de datos personales del Marco Legal define finalidades, bases jurídicas y retenciones; aquí se indican los controles técnicos que posibilitan su cumplimiento.

Los controles adoptados (RBAC, JWT, cifrado, headers, CORS, validación, logging y backups) son coherentes con el alcance del MVP y proporcionan una base segura y escalable para futuras integraciones o ampliaciones funcionales.

Vista Arquitectónica: Vista de Seguridad y Autenticación

Diagrama:



Artefactos: Diagrama de secuencia (PNG/SVG), Diagrama de componentes de seguridad (PNG/SVG), Controlador de autenticación, Servicio JWT (firma/verificación/exp/jti), Esquemas de datos (usuario y lista de revocación), Configuración de seguridad (headers/CORS/cookies), Variables y secretos (plantilla), Colección de pruebas (Postman/Newman), Pruebas unitarias/integración (auth, rutas protegidas, expiración/blacklist)..

Nombre del diagrama:

Diagrama de Seguridad y Autenticación

Versión: 1.0

Notación: UML 2.5.1 – Diagrama de Seguridad y Autenticación

Autor(es): Juan Carlos Arrieta Bustos Richard Pardo Cardona	Fecha de creación: _10 / 01 / 2025_
--	---

Desarrollo del Prototipo Mediante el Marco Ágil Scrum

A continuación, se presenta el desarrollo de un prototipo funcional, realizado con el marco ágil Scrum, adaptado a las necesidades del proyecto y enfocado en una aplicación web para optimizar la gestión de inventarios y ventas en ADIIC Dotaciones Institucionales. En ese orden de ideas, se describe cómo se aplicó Scrum durante el proceso de desarrollo, desde la identificación del tipo de prototipo, la configuración del entorno, la definición del product backlog, hasta la ejecución de cada uno de los sprints.

Tipo de Prototipo

Desde el inicio del desarrollo se optó por una estrategia basada en un prototipo funcional de alta fidelidad, alineada con las prácticas del marco ágil Scrum. A diferencia de los métodos tradicionales, donde los prototipos suelen ser exploratorios o descartables, en este proyecto cada Sprint resultó en una versión operativa del producto que fue probada, validada y ajustada de acuerdo con la retroalimentación recibida. La implementación se llevó a cabo de manera desacoplada entre frontend y backend, lo que permitió mayor flexibilidad y escalabilidad en la evolución del sistema. Cada incremento del producto incorporó funcionalidades clave que aportaron valor tangible, asegurando entregas iterativas potencialmente desplegables desde los primeros ciclos de desarrollo.

Configuración del Entorno (Sprint 0)

Antes del primer Sprint, se realizó la configuración técnica del entorno de desarrollo para garantizar el correcto avance del proyecto. Se utilizó Visual Studio Code como IDE principal, dada su integración con tecnologías clave del stack utilizado y herramientas de control de versiones. En esta fase se instalaron las librerías y dependencias esenciales: en el frontend, herramientas como React, React Router, Axios y Tailwind CSS permitieron construir interfaces

dinámicas y seguras; en el backend, Express.js, Mongoose y jsonwebtoken facilitaron la creación de rutas API y la autenticación. Las pruebas iniciales se llevaron a cabo en equipos con Windows 10 y navegador Google Chrome, asegurando compatibilidad y rendimiento. Asimismo, se establecieron repositorios independientes en GitHub para frontend y backend bajo una estrategia de trabajo con ramas. Finalmente, se implementó una versión mínima funcional (MVP) como punto de partida para los desarrollos posteriores.

Figura 42

Árbol de carpetas en VS code del frontend y backend del proyecto



Fuente. Autoría Propia.

Product Backlog

El Product Backlog del proyecto constituye el eje central de la planificación ágil, y representa una lista priorizada de funcionalidades, requisitos técnicos y mejoras del sistema a desarrollar. Siguiendo las buenas prácticas establecidas por el marco ágil Scrum, cada ítem se ha documentado como una historia de usuario, lo cual facilita la comprensión del objetivo desde la perspectiva del usuario final y permite estimar el esfuerzo requerido de manera más precisa.

Cada entrada del backlog incluye los siguientes campos:

- ID: Código único de identificación de la historia.
- Historia de Usuario: Requerimiento redactado en lenguaje natural, centrado en el valor para el usuario.
- Descripción: Explicación detallada del objetivo funcional de la historia.
- Prioridad: Nivel de importancia para el negocio (Alta, Media, Baja).
- Estimación: Tiempo estimado para su implementación, expresado en horas.
- Criterios de Aceptación: Condiciones mínimas que deben cumplirse para considerar la historia como completa.
- Sprint: Iteración planeada en la que se desarrollará la funcionalidad.
- Estado: Situación actual de la historia (Pendiente, En desarrollo o Completada).

En total, el Product Backlog incluye 32 historias de usuario (HU001 a HU032), cubriendo desde la autenticación de usuarios y gestión de productos, hasta la creación de facturas y visualización de estadísticas. La Tabla X, presentada a continuación, muestra un extracto representativo de este backlog.

Tabla 8

Product backlog del proyecto

ID	HISTORIA DE USUARIO	DESCRIPCIÓN	PRIORIDAD	ESTIMACIÓN	CRITERIOS DE ACEPTACIÓN	SPRINT	ESTADO
HU001	Página de inicio con login integrado	Como usuario de la aplicación, Quiero contar con una página de inicio que muestre el logo, el nombre de la empresa, una breve descripción de los servicios y el formulario de inicio de sesión, Para poder obtener una experiencia clara y profesional que me permita acceder a la aplicación de manera intuitiva.	Alta	4 horas	<ul style="list-style-type: none"> • La página de inicio debe mostrar el logo y nombre de la empresa en la parte superior central. • Debe incluir un fondo representativo de un almacén o inventario. • Debe contener una breve descripción de los servicios ofrecidos. • Se debe mostrar el componente de Login que permita autenticarse. • Debe adaptarse a diferentes tamaños de pantalla (móvil, Tablet, escritorio). 	1	Completada
HU002	Formulario de login con redirección por rol.	Como usuario registrado, Quiero acceder a la aplicación ingresando mi correo electrónico y contraseña, Para poder acceder a la plataforma según mi rol asignado.	Alta	6 horas	<ul style="list-style-type: none"> • El sistema debe permitir el ingreso de un correo electrónico y una contraseña. • El correo electrónico debe tener un formato válido y pertenecer a un dominio permitido. • La contraseña debe tener al menos 5 caracteres. • Si las credenciales son correctas, el sistema debe autenticar al usuario y redirigirlo según su rol. 	1	Completada

					<ul style="list-style-type: none">• Si el usuario no está registrado, debe mostrarse un mensaje de error.• Si la contraseña es incorrecta, debe mostrarse un mensaje de error.• El sistema debe almacenar un token JWT en el almacenamiento local del navegador.• Si el usuario ya ha iniciado sesión, debe ser redirigido automáticamente a su respectiva página según su rol.		
HU003	Sidebar para acceso a funcionalidades de gestión	Como administrador, Quiero un Sidebar con accesos directos a las funcionalidades de gestión, Para poder administrar eficientemente las categorías, productos, clientes, cabeceras y facturas del sistema.	Alta	3 horas	<ul style="list-style-type: none">• El Sidebar debe estar disponible en todas las páginas donde el administrador tenga acceso.• Debe contener enlaces de navegación hacia: categorías, crear categoría, crear producto, crear cuenta, clientes, cabeceras y facturas.• Los botones deben estar estilizados con diseño visual atractivo y responsivo.• El Sidebar debe mantenerse fijo en el lado izquierdo de pantallas de PC y adaptarse a diferentes tamaños de pantalla.• La navegación debe ser intuitiva, con botones grandes y texto en mayúsculas.• Si el administrador tiene una categoría seleccionada	1	Completada

					(idCategoria \neq null), se debe habilitar el botón de "Crear Producto".		
					<ul style="list-style-type: none"> • Si no hay categoría seleccionada (idCategoria = null), el botón "Crear Producto" debe estar deshabilitado visual y funcionalmente. 		
HU004	Encabezado de la aplicación web	Como usuario autenticado, Quiero visualizar un encabezado en la aplicación web que contenga el logo de la empresa, el nombre y descripción del negocio, y un botón para cerrar sesión, Para poder identificar rápidamente la plataforma y gestionar mi sesión de usuario.	Alta	5 horas	<ul style="list-style-type: none"> • El encabezado debe estar presente en todas las páginas de la aplicación después de iniciar sesión. • Debe mostrar: Logo de la empresa (izquierda), nombre y descripción del negocio (centro), botón de "Cerrar Sesión" (derecha). • El diseño debe ser responsivo, visible correctamente en diferentes tamaños de pantalla. • Al hacer clic en "Cerrar Sesión": Se debe eliminar el token JWT del almacenamiento local, el usuario debe ser redirigido automáticamente a la página de inicio. 	1	Completada
HU005	Página de administración de categorías y productos	Como administrador del sistema, Quiero acceder a una página de administración donde pueda visualizar, actualizar y eliminar categorías, así como gestionar los	Media	8 horas	<ul style="list-style-type: none"> • Si el usuario no está autenticado, debe ser redirigido a la página de inicio de sesión. • El acceso al módulo debe validarse mediante la existencia de un token JWT. 	2	Completada

		productos asociados, Para poder mantener actualizada la organización de los artículos en la plataforma.			<ul style="list-style-type: none"> • El administrador debe poder: Ver una lista de categorías con imagen y nombre, eliminar una categoría con confirmación previa, editar una categoría desde una página específica de edición, acceder a la gestión de productos por categoría. • Las categorías deben cargarse dinámicamente desde una API. • Deben mostrarse mensajes de éxito o error según el resultado de las acciones. • Mientras se cargan los datos, debe visualizarse un indicador de carga. 		
HU006	Registro de nuevos usuarios con asignación de rol	Como administrador, Quiero registrar nuevos usuarios en la plataforma asignándoles un rol específico, Para poder gestionar el acceso y permisos según sus funciones dentro del sistema.	Baja	4 horas	<ul style="list-style-type: none"> • Solo los usuarios con rol de “admin” pueden acceder al formulario de registro. • El formulario debe incluir los siguientes campos obligatorios: Nombre, Correo electrónico, Rol (admin o regular), Contraseña, Confirmación de contraseña. • Si la contraseña y su confirmación no coinciden, debe mostrarse un mensaje de error. • Si el correo electrónico ya existe en la plataforma, debe mostrarse un mensaje de error. • Si la información es válida, el sistema debe registrar al 	2	Completada

					nuevo usuario y mostrar una confirmación.		
					<ul style="list-style-type: none"> • Al completar el registro, será redirigido al panel de administración. 		
HU007	Creación de nuevas categorías	Como administrador, Quiero crear nuevas categorías en el sistema, Para poder organizar los productos de manera eficiente.	Media	6 horas	<ul style="list-style-type: none"> • El administrador debe haber iniciado sesión para acceder a la página de creación de categorías. • El formulario debe incluir campos obligatorios: Nombre de la categoría, URL de la imagen representativa. • Debe existir un botón para registrar la categoría. • Si la categoría se crea correctamente: Mostrar mensaje de éxito, redirigir automáticamente a la vista principal de administración. • Si ocurre un error, debe mostrarse un mensaje de error. • El formulario debe ser intuitivo y responsivo, adaptado a distintos dispositivos. 	2	Completada
HU008	Actualización de categorías existentes	Como administrador, Quiero actualizar la información de una categoría existente, Para mantener la información de las categorías correctamente actualizada en el sistema.	Media	5 horas	<ul style="list-style-type: none"> • El administrador debe estar autenticado para acceder a esta funcionalidad. • El sistema debe cargar automáticamente los datos actuales de la categoría seleccionada. • El formulario debe permitir modificar el: Nombre de la 	2	Completada

					<p>categoría e imagen representativa.</p> <ul style="list-style-type: none"> • Al hacer clic en “Actualizar Categoría”, el sistema debe validar los campos antes de enviar. • Si la actualización es exitosa: Mostrar un mensaje de confirmación, redirigir al administrador a la vista de administración. • Si ocurre un error, se debe mostrar un mensaje de error claro. • Si el usuario no está autenticado, debe ser redirigido al login. 		
HU009	Actualización de información de un producto	Como administrador, Quiero actualizar la información de un producto existente, Para mantener los datos correctos y actualizados en el inventario.	Media	3 horas	<ul style="list-style-type: none"> • El sistema debe recuperar y mostrar los datos actuales del producto seleccionado basándose en su ID. • Si el producto no existe o hay error al cargarlo, debe mostrarse un mensaje de error. • El formulario debe permitir modificar los siguientes campos: Referencia, Nombre, Descripción, Talla, Color, Stock, Precio. • Todos los campos son obligatorios, excepto la imagen. • El stock debe ser un número entero positivo o cero. • El precio debe ser un número mayor a cero. 	2	Completada

					<ul style="list-style-type: none"> • Si algún dato no es válido, el sistema debe mostrar un mensaje de error. • Al hacer clic en “Actualizar Producto”: Si los datos son válidos, debe enviarse la solicitud PUT al servidor, si es exitosa, debe mostrarse un mensaje de confirmación y redirigir al listado de productos de la categoría, si falla, debe mostrarse un mensaje de error. 		
HU010	Registro de nuevos productos asociados a categorías	Como administrador, Quiero registrar nuevos productos en el sistema asignándolos a una categoría específica, Para que estos puedan ser visualizados y gestionados en la plataforma.	Media	7 horas	<ul style="list-style-type: none"> • El administrador debe estar autenticado para acceder a esta funcionalidad. • El formulario debe incluir los siguientes campos obligatorios: Referencia, Nombre, Descripción, Talla, Color, Stock, Precio, Imagen. • Los campos stock y precio deben aceptar solo valores numéricos válidos. • Al enviar el formulario: El producto debe registrarse en la base de datos, debe asociarse correctamente a la categoría seleccionada. • Si la creación es exitosa: Se debe mostrar un mensaje de confirmación, redirigir a la página de listado de productos de la categoría. • Si ocurre un error, debe mostrarse un mensaje con la causa. 	2	Completada

					<ul style="list-style-type: none"> • Debe incluirse un Sidebar para navegación dentro del panel de administración. • Debe incluirse un header con botón de cierre de sesión. 		
HU011	Listado de productos por categoría	Como administrador, Quiero ver un listado de productos organizados por categoría, Para poder gestionar y visualizar los productos de manera eficiente.	Media	4 horas	<ul style="list-style-type: none"> • Se debe obtener el ID de la categoría a partir de la URL. • Si no se detecta un ID de categoría, debe mostrarse un mensaje de error en consola. • El sistema debe hacer una solicitud a la API para obtener los productos de la categoría correspondiente. • Mientras se cargan los productos, debe mostrarse un mensaje de “Cargando productos...”. • Si hay productos disponibles: Deben mostrarse en la interfaz con sus respectivos datos. • Si no hay productos disponibles: Debe mostrarse el mensaje “No hay productos disponibles”. • La interfaz debe incluir un encabezado, un sidebar y una lista bien estructurada. • El diseño debe ser responsivo y visualmente atractivo, usando colores y estilos adecuados. 	2	Completada
HU012	Visualización y gestión de productos de una categoría específica	Como administrador, Quiero visualizar los productos de una categoría específica, Para poder	Baja	6 horas	<ul style="list-style-type: none"> • Al cargar la vista de productos, el sistema debe recuperar y mostrar la lista de 	2	Completada

		gestionar su información y disponibilidad.				<p>productos de la categoría seleccionada.</p> <ul style="list-style-type: none"> • Cada producto debe mostrar: referencia, nombre, descripción, talla, color, stock, precio e imagen. • Al hacer clic en “Editar”, el usuario debe ser redirigido a la página de actualización con los datos del producto seleccionado. • Al hacer clic en “Eliminar”, debe mostrarse una alerta de confirmación. • Si se confirma la eliminación: El producto debe eliminarse de la base de datos, la lista debe actualizarse en la interfaz. • Si se cancela la eliminación, no debe ocurrir ningún cambio. • Si ocurre un error al cargar productos o al eliminarlos, se debe mostrar un mensaje de error. • La interfaz debe ser responsiva y adaptarse a distintos dispositivos. 		
HU013	Visualización de categorías por parte del usuario regular (vendedor)	Como usuario Regular (Vendedores), Quiero visualizar las categorías de productos disponibles en el sistema, Para poder seleccionar una y gestionar sus productos.	Baja	8 horas		<ul style="list-style-type: none"> • Si el usuario no ha iniciado sesión, debe ser redirigido a la página de inicio de sesión. • El sistema debe validar la existencia del token en el almacenamiento local antes de mostrar la vista. 	3	Completada

					<ul style="list-style-type: none"> • Las categorías deben obtenerse desde la API /api/categorias al cargar la página. • Si la respuesta de la API contiene datos válidos, las categorías deben mostrarse en la interfaz. • Si ocurre un error al obtener las categorías, debe mostrarse un mensaje con opción de reintento. • La página debe tener un encabezado con el título “Panel de vendedores”. • Cada categoría debe mostrarse como tarjeta con: Imagen, nombre y botón para ver productos asociados. • El botón “Productos” debe dirigir a la página de gestión de productos de la categoría seleccionada. • La interfaz debe ser responsiva, adaptándose a distintos tamaños de pantalla. • Se debe aplicar una paleta de colores en tonos lima y verde. 		
HU014	Visualización de productos por categoría para vendedores	Como usuario Regular (Vendedores), Quiero ver los productos de una categoría específica, Para poder conocer su información y disponibilidad.	Baja	6 horas	<ul style="list-style-type: none"> • Si el usuario no está autenticado, debe ser redirigido a la página de inicio de sesión. • El sistema debe obtener la información de la categoría seleccionada desde la API. 	3	Completada

					<ul style="list-style-type: none"> • El sistema debe obtener la lista de productos asociados a la categoría desde la API. • Si ocurre un error al cargar datos, debe mostrarse un mensaje de error informativo. • El nombre de la categoría debe mostrarse en la parte superior de la interfaz. • Si hay productos en la categoría: Deben mostrarse en una cuadrícula con su imagen y detalles (nombre, precio, stock, etc.). • Si no hay productos: Debe mostrarse un mensaje indicando que no hay productos disponibles. • Debe incluirse un botón para regresar a la vista de categorías. • La interfaz debe ser responsiva y accesible desde distintos dispositivos. 		
HU015	Creación de cabeceras para facturas por sucursal	Como administrador, Quiero poder crear nuevas cabeceras para las facturas con información como nombre del local, NIT, dirección, teléfono y correo electrónico, Para poder personalizar la factura según la sucursal emisora.	Baja	5 horas	<ul style="list-style-type: none"> • El sistema debe verificar que el usuario tiene permisos de administrador. • Los campos obligatorios del formulario son: Nombre del local, dirección, NIT, teléfono y correo electrónico. • Se debe validar que ningún campo esté vacío. • El formato de la Cabecera para la factura debe ser válido según normativas fiscales locales. 	3	Completada

HU016	Edición de cabeceras	Como administrador, Quiero poder editar las cabeceras existentes, Para actualizar la información en caso de cambios en los datos de la sucursal.	Baja	6 horas	<ul style="list-style-type: none"> • Debe verificarse que no exista otra cabecera con el mismo NIT. • Si los datos son válidos: La cabecera debe registrarse correctamente en la base de datos, debe mostrarse un mensaje de confirmación. <ul style="list-style-type: none"> • El sistema debe verificar que el usuario tiene permisos de administrador. • El formulario de edición debe mostrar los datos actuales cargados automáticamente. • Se deben validar los siguientes campos obligatorios: Nombre del local, NIT, Dirección, teléfono y correo electrónico. • El formato de la cabecera para la factura debe mantenerse válido según las normas fiscales aplicables. • No debe permitirse la existencia de otra cabecera con el mismo NIT. • Si la actualización es exitosa: Se debe guardar la información modificada, debe mostrarse un mensaje de confirmación. • Si ocurre un error, se debe mostrar un mensaje de advertencia adecuado. 	3	Completada
--------------	----------------------	---	------	---------	---	---	------------

HU017	Eliminación de cabeceras innecesarias	Como administrador, Quiero poder eliminar cabeceras que ya no sean necesarias, Para mantener organizada la base de datos.	Baja	4 horas	<ul style="list-style-type: none"> • El sistema debe verificar que el usuario tenga permisos de administrador. • Se debe solicitar una doble confirmación al administrador antes de proceder con la eliminación. • Si la cabecera cumple con las condiciones y se confirma la acción: Debe eliminarse correctamente de la base de datos, debe mostrarse un mensaje de éxito. • Si no se cumplen las condiciones o se cancela la acción: La cabecera debe mantenerse intacta, se debe notificar adecuadamente al usuario. 	4	Completada
HU018	Búsqueda de cabeceras por NIT en gestión administrativa	Como administrador, Quiero buscar cabeceras existentes ingresando el NIT, Para localizarlas rápidamente y poder editarlas o eliminarlas con eficiencia.	Baja	3 horas	<ul style="list-style-type: none"> • El campo de búsqueda debe aceptar únicamente valores válidos de NIT. • Si existe una cabecera con el NIT ingresado, debe mostrarse en los resultados. • Si no existe, debe mostrarse un mensaje claro de “No se encontraron resultados”. • La búsqueda debe ofrecer retroalimentación inmediata o casi en tiempo real. • La funcionalidad debe integrarse dentro del módulo de Gestión de Cabeceras. 	4	Completada

HU019	Selección de cabecera al crear una factura	Como usuario de la aplicación, Quiero poder seleccionar una cabecera específica al momento de crear una factura, Para asegurar que la información de la sucursal sea la correcta.	Baja	6 horas	<ul style="list-style-type: none"> • El sistema debe mostrar un menú desplegable con las cabeceras disponibles activas. • El usuario debe seleccionar una cabecera antes de completar la factura. • Solo se deben mostrar en el menú las cabeceras habilitadas y con datos completos. • Si el usuario intenta generar una factura sin haber seleccionado una cabecera: El sistema debe impedirlo, debe mostrarse un mensaje de advertencia. • La cabecera seleccionada debe reflejar: Nombre del local, NIT, dirección, teléfono y correo electrónico en la factura generada. • La funcionalidad debe ser intuitiva y rápida. 	4	Completada
HU020	Creación de nueva factura	Como usuario de la aplicación, Quiero poder crear una nueva factura ingresando los datos de la cabecera, cliente y seleccionando los productos que vendo, Para registrar la venta y entregar un comprobante al cliente.	Baja	5 horas	<ul style="list-style-type: none"> • El sistema debe verificar que se ha seleccionado una cabecera válida antes de permitir la creación. • El formulario debe incluir campos obligatorios del cliente: Nombre completo, Identificación (NIT o cédula), dirección, ciudad, teléfono. • La identificación debe validarse con formato correcto según tipo. • El vendedor debe poder seleccionar uno o varios 	4	Completada

					<p>productos disponibles en el inventario.</p> <ul style="list-style-type: none"> • Se debe validar que al menos un producto haya sido agregado antes de permitir la facturación. • El sistema debe calcular dinámicamente el resumen de la venta: Subtotal, Descuento, IVA, Total. • Al hacer clic en el botón “Generar Factura”, el sistema debe: Validar todos los campos, registrar la factura en la base de datos, mostrar un mensaje de confirmación. • Si algún dato es inválido o falta información, debe mostrarse un mensaje de error claro y específico. 		
HU021	Agregar productos con cantidad, descuento e IVA a la factura	Como usuario de la aplicación, Quiero poder agregar múltiples productos a una factura indicando su cantidad, descuento e IVA, Para reflejar con precisión lo que el cliente está comprando y cumplir con la normativa fiscal.	Alta	3 horas	<ul style="list-style-type: none"> • El vendedor debe poder seleccionar productos del inventario y agregarlos uno por uno a la factura. • Para cada producto agregado, debe poder ingresarse: Cantidad (número entero positivo > 0), porcentaje de descuento (entre 0% y 100%), porcentaje de IVA (por lo general 0%, 5%, 19% u otro definido por el sistema fiscal). • El sistema debe verificar que: La cantidad no exceda el stock disponible, el descuento y el IVA estén dentro de los rangos permitidos. 	4	Completada

HU022	Cálculo automático de totales en la factura	Como usuario de la aplicación, Quiero que el sistema calcule automáticamente los subtotales, descuentos, IVA y total final mientras agrego productos, Para visualizar el monto a cobrar en tiempo real y asegurar precisión en la factura.	Alta	8 horas	<ul style="list-style-type: none"> • El sistema debe calcular automáticamente por cada producto: Subtotal sin IVA ni descuento, monto descontado, monto de IVA, total final del producto. • El resumen general de la factura debe actualizarse dinámicamente al modificar productos. • La interfaz debe permitir: Editar cantidades, descuentos e IVA antes de finalizar o eliminar productos de la factura. • Si algún dato ingresado es inválido, el sistema debe: Bloquear la acción, mostrar un mensaje de error o advertencia clara. 	4	Completada
					<ul style="list-style-type: none"> • Cada vez que se agregue, edite o elimine un producto de la factura, el sistema debe recalcular automáticamente: Subtotal sin descuentos ni IVA, monto total de descuento aplicado, monto total de IVA, total final de la factura. • Los cálculos deben actualizarse en tiempo real o con mínima latencia perceptible. • El porcentaje de IVA aplicado debe corresponder al configurado para cada producto. 		

					<ul style="list-style-type: none"> • El descuento por producto debe integrarse correctamente antes de aplicar el IVA. • Los resultados deben ser numéricamente precisos y libres de errores de redondeo. • Cualquier modificación en cantidad, descuento o eliminación de un producto debe actualizar el resumen. • La interfaz debe mostrar un resumen detallado y claro de los montos calculados. • Si los datos de algún producto son inválidos, el sistema no debe recalcular y debe mostrar un mensaje de error. 		
HU023	Verificación automática del stock al facturar	Como usuario de la aplicación, Quiero que el sistema verifique automáticamente si hay suficiente stock de los productos que estoy añadiendo a la factura, Para evitar vender más de lo disponible.	Alta	6 horas	<ul style="list-style-type: none"> • Al intentar agregar un producto a la factura, el sistema debe consultar en tiempo real el stock actual disponible en inventario. • Si la cantidad ingresada supera el stock, el sistema debe: Bloquear la acción, mostrar un mensaje claro de error (por ejemplo: “Solo hay "X" unidades disponibles”). • El sistema debe bloquear la posibilidad de aumentar la cantidad por encima del stock desde cualquier control numérico. • Si se modifica el stock en la base de datos (por otra venta), el sistema debe actualizar automáticamente el valor 	4	Completada

					<p>disponible en la interfaz antes de generar la factura.</p> <ul style="list-style-type: none"> • Solo debe permitirse facturar productos con cantidad mayor que cero y dentro del stock. • Si hay problemas de conexión o falla al consultar el stock: Debe mostrarse un mensaje de error, no debe permitirse continuar con la operación. 		
HU024	Edición de factura antes de finalizar	Como usuario de la aplicación, Quiero poder editar una factura al momento de estar creándola, Para corregir errores o realizar cambios solicitados por el cliente antes de finalizar la venta.	Alta	5 horas	<ul style="list-style-type: none"> • El sistema debe verificar que la factura aún no ha sido impresa ni finalizada para permitir su edición. • El vendedor debe poder: Modificar cantidades de productos, cambiar productos agregados, editar descuentos, actualizar los datos del cliente si es necesario. • Al editar las cantidades: El sistema debe verificar que no se exceda el stock disponible actualizado. • Cada modificación debe recalcular automáticamente los subtotales, descuentos, IVA y total final de la factura. • Todos los campos obligatorios deben mantenerse completos y válidos tras la edición. • Al generar los cambios, el sistema debe: Actualizar la información de los datos de la factura que se está generando 	4	Completada

					para ser guardados en la base de datos.		
					<ul style="list-style-type: none"> • Si ocurre un error, debe mostrarse un mensaje explicativo y no guardar cambios inconsistentes. 		
HU025	Descarga de factura finalizada en PDF	Como usuario de la aplicación, Quiero poder descargar la factura en formato PDF una vez finalizada, Para entregarle al cliente un comprobante de su compra que pueda imprimirse o enviarse digitalmente.	Alta	6 horas	<ul style="list-style-type: none"> • El sistema debe verificar que la factura esté en estado “finalizada” antes de habilitar la descarga. • Al hacer clic en el botón “Generar factura”, el sistema debe: Validar que los datos del cliente y los productos estén completos, calcular y mostrar el resumen de la venta (subtotal, descuentos, IVA, total), generar un archivo PDF de la factura con diseño adecuado. • El archivo debe incluir: Cabecera (nombre del local, NIT, dirección, teléfono y correo electrónico), datos del cliente, Fecha y hora de emisión, detalles de los productos vendidos (cantidad, precio, IVA, descuento), Total final. • El archivo PDF debe descargarse automáticamente en el dispositivo del usuario. • El sistema debe marcar la factura como “descargada” o “emitida” para evitar ediciones posteriores. 	4	Completada

					<ul style="list-style-type: none"> • Si ocurre un error al generar el archivo, debe mostrarse un mensaje claro y evitar marcar la factura como finalizada. 		
HU026	Búsqueda de facturas por número o NIT (administrador)	Como administrador, Quiero poder buscar facturas anteriores por su número o por el NIT del cliente, Para atender solicitudes de cambios, devoluciones o auditorías y facilitar la trazabilidad de las ventas.	Alta	4 horas	<ul style="list-style-type: none"> • El campo de búsqueda debe permitir ingresar: El número de factura o el NIT del cliente. • El sistema debe validar que el campo: No esté vacío. • Si existen resultados coincidentes: Deben mostrarse con información relevante (número, cliente, fecha, total). • Si no existen coincidencias: Debe mostrarse un mensaje claro como “No se encontraron resultados”. • El sistema debe realizar la búsqueda de forma rápida y eficiente. • El administrador debe tener acceso completo a todas las facturas del sistema, sin restricciones por usuario. • Desde el resultado, debe poder acceder al detalle completo de la factura seleccionada. 	4	Completada
HU027	Descuento automático del stock al generar factura	Como usuario de la aplicación, Quiero que, al generar una factura de venta, el sistema descunte automáticamente del stock la cantidad de productos	Baja	5 horas	<ul style="list-style-type: none"> • El sistema debe ejecutar el descuento del stock únicamente después de que la factura haya sido generada exitosamente. 	4	Completada

		vendidos, Para mantener el inventario actualizado y reflejar las ventas en tiempo real.				<ul style="list-style-type: none"> • La cantidad descontada debe corresponder exactamente a los valores registrados por producto en la factura. • El descuento debe reflejarse inmediatamente en la base de datos del inventario. • No debe permitirse que el stock llegue a valores negativos. • Si hay error durante la transacción de descuento, la factura no debe completarse, y debe mostrarse un mensaje de error. • Si la factura contiene múltiples productos, deben descontarse todos correctamente en la misma operación (transacción segura). 		
HU028	Consulta del historial de facturación	Como administrador, Quiero consultar el historial de facturación, Para realizar revisiones, auditorías internas o gestionar datos contables.	Alta	4 horas		<ul style="list-style-type: none"> • El sistema debe proporcionar al administrador una vista completa de todas las facturas generadas en el sistema. • Esta vista debe incluir información clave por factura: Fecha de emisión, Número de factura, Subtotal, Descuento y Total. • El sistema ordenara las facturas por fecha (de la más reciente a la más antigua). • La interfaz debe cargar los datos sin demoras significativas. 	4	Completada

HU029	Visualización de lista de clientes con compras (administrador)	Como administrador, Quiero ver la lista de todos los clientes que han realizado compras en la tienda, Para consultar su historial, verificar sus datos y gestionar correctamente la relación comercial.	Baja	3 horas	<ul style="list-style-type: none"> • Debe haber una opción para ver el detalle completo de una factura específica desde esta lista (ver factura en PDF en el navegador). • Solo los usuarios con rol de administrador deben tener acceso a esta vista. <ul style="list-style-type: none"> • Verificar que el usuario tenga permisos para ver los detalles de la factura. • El sistema debe permitir al administrador acceder a una vista de clientes registrados mediante compras realizadas. • La lista debe mostrar por cada cliente: Nombre completo, NIT o cédula, dirección, ciudad, teléfono y compras realizadas. • Deben incluirse únicamente clientes que tengan al menos una factura registrada. • La interfaz debe: Ser ordenada, permitir búsqueda por cedula o NIT. • Desde esta lista, el administrador debe poder: Acceder al detalle del cliente, consultar el número de compras realizadas por el cliente. • El acceso debe estar restringido solo a usuarios con rol de administrador. 	4	Completada
--------------	--	---	------	---------	--	---	------------

HU030	Búsqueda de clientes por NIT o identificación (administrador)	Como administrador, Quiero poder buscar un cliente específico por su NIT o número de identificación, Para localizarlo rápidamente en la base de datos sin necesidad de recorrer toda la lista.	Baja	6 horas	<ul style="list-style-type: none"> • El campo de búsqueda debe aceptar tanto: El NIT o número de identificación. • Si el campo está vacío, el sistema debe: Bloquear la búsqueda, mostrar un mensaje como “Por favor, ingrese un número de cédula o NIT para buscar”. • Si no hay coincidencias: Debe mostrarse el mensaje “No hay clientes registrados”. • Si hay resultados, deben mostrarse en una lista con: Nombre completo, cedula o NIT, dirección, ciudad, teléfono y numero de compras. • La búsqueda debe ofrecer retroalimentación rápida y correcta. • Solo los usuarios con rol de administrador deben tener acceso a esta funcionalidad. 	4	Completada
HU031	Visualización del número de compras por cliente	Como administrador, Quiero poder ver el número de compras realizadas por cada cliente, Para identificar a los clientes frecuentes y ofrecerles promociones especiales.	Baja	5 horas	<ul style="list-style-type: none"> • El sistema debe verificar que el usuario tenga rol de administrador antes de permitir el acceso a esta funcionalidad. • El contador de compras por cliente debe considerar solo facturas finalizadas. • La información debe mostrarse en una tabla o lista, incluyendo: Nombre del cliente, cedula o NIT, Dirección, Ciudad, teléfono y 	4	Completada

					<p>Cantidad total de compras realizadas.</p> <ul style="list-style-type: none"> • El sistema debe ofrecer una interfaz de carga rápida incluso con volúmenes grandes de datos. 		
HU032	Registro automático de cliente al realizar la primera compra	Como usuario de la aplicación, Quiero que el sistema registre automáticamente a un nuevo cliente cuando realiza su primera compra, Para evitar tener que ingresar sus datos en un formulario separado y agilizar el proceso de facturación.	Baja	4 horas	<ul style="list-style-type: none"> • El sistema debe verificar que el campo de identificación (cédula o NIT) tenga el formato correcto, según el tipo seleccionado. • Antes de registrar al cliente, el sistema debe consultar si ya existe una identificación igual en la base de datos. • Si el NIT o cédula ya están registrados, el sistema debe: Asociar la venta al cliente existente, evitar crear un duplicado. • Si no existe, se debe registrar automáticamente un nuevo cliente con los datos ingresados en la factura. • Los campos obligatorios para validar son: Nombre completo, cédula o NIT, dirección, ciudad y teléfono. • El cliente debe quedar registrado en la base de datos al finalizar la venta, sin pasos adicionales para el vendedor. • Esta funcionalidad debe estar disponible solo para usuarios con rol de vendedor o superior. 	4	Completada

Nota. La tabla presenta el Product Backlog del proyecto, el cual contiene la lista priorizada de requisitos, funcionalidades y mejoras necesarias para el desarrollo del sistema de gestión de inventarios del almacén.

Desarrollo Iterativo del Proyecto (Sprints)

Sprint 1: Autenticación y accesibilidad inicial

Sprint Planning

Sprint #	Duración	Sprint Goal (Meta del Sprint)	Historias seleccionadas	Tareas técnicas derivadas	Estimación
Sprint 1	7 al 21 de abril de 2025	Establecer la estructura básica de la aplicación, incluyendo el sistema de autenticación de usuarios y acceso según roles.	HU001, HU002, HU003, HU004	Configurar rutas en React Router DOM. Integrar login con token JWT. Crear componentes Header y Sidebar reutilizables.	18 horas

Sprint Execution

Durante este sprint se construyó la base visual de la interfaz de usuario y se integró un sistema de autenticación funcional usando JWT. Se aseguraron buenas prácticas de seguridad y diseño responsive.

Figura 43

Página de inicio/login y pagina de registro de usuario



Fuente. Autoría Propia.

Sprint Review

Sprint #	Historias completadas	Demostración	Feedback recibido	Decisiones
Sprint 1	HU001, HU002, HU003, HU004	Se presentó el flujo completo de autenticación en la aplicación, incluyendo la interfaz de inicio de	Mejorar la visibilidad del botón de “Cerrar sesión”. Hacer que el Sidebar sea colapsable.	Aplicar mejoras estéticas en el próximo sprint y separar los componentes en carpetas reutilizables.

sesión, la validación de credenciales mediante JWT, y la redirección dinámica según el rol del usuario (admin o regular). También se evidenció la navegación interna entre módulos a través de un layout base con menú lateral y cabecera reutilizable.

Retrospectiva

- ¿Qué funcionó bien?
 - Integración del login y control de rutas por roles.
- ¿Qué se puede mejorar?

- Modularización temprana del código.
- ¿Qué bloqueos se presentaron
- Ajustes con rutas protegidas y refresco del token.

Sprint 2: Gestión de categorías y productos

Sprint Planning

Sprint #	Duración	Sprint Goal (Meta del Sprint)	Historias seleccionadas	Tareas técnicas derivadas	Estimación
Sprint 2	21 de abril al 07 de mayo de 2025	Desarrollar los módulos de administración de categorías y productos del inventario.	HU005, HU006, HU007, HU008, HU009, HU010, HU011, HU012	Crear formularios con React Hook Form. Configurar rutas y consumo de la API para CRUD. Validar autenticación de administrador.	43 horas

Sprint Execution

Se construyó la estructura lógica para manejar categorías y productos desde el frontend y backend, incluyendo validaciones, formularios, y vistas adaptables.

Figura 44

Páginas de creación de categorías y productos



Fuente. Autoría Propia.

Sprint Review

Sprint #	Historias completadas	Demostración	Feedback recibido	Decisiones
Sprint 2	HU005, HU006, HU007, HU008, HU009, HU010, HU011, HU012	Se mostró el funcionamiento del módulo de administración de inventario, destacando	Agregar confirmación visual al eliminar productos. Validar	Mejorar feedback de la interfaz y adaptar selectores para categorías en el Sprint 3.

la creación, edición y eliminación de categorías y productos mediante formularios validados. Se evidenció el consumo de la API REST para operaciones CRUD y el control de acceso según permisos del rol administrador. La interfaz fue demostrada desde el panel de gestión con actualización en tiempo real. imágenes para evitar errores.

Retrospectiva

- ¿Qué funcionó bien?
 - Conexión efectiva con MongoDB y validación robusta.
- ¿Qué se puede mejorar?
 - Modularidad en los servicios de axios.
- ¿Qué bloqueos se presentaron?
 - Configuración inicial del Cloudinary, que fue descartada.

Sprint 3: Facturación básica y gestión de clientes

Sprint Planning

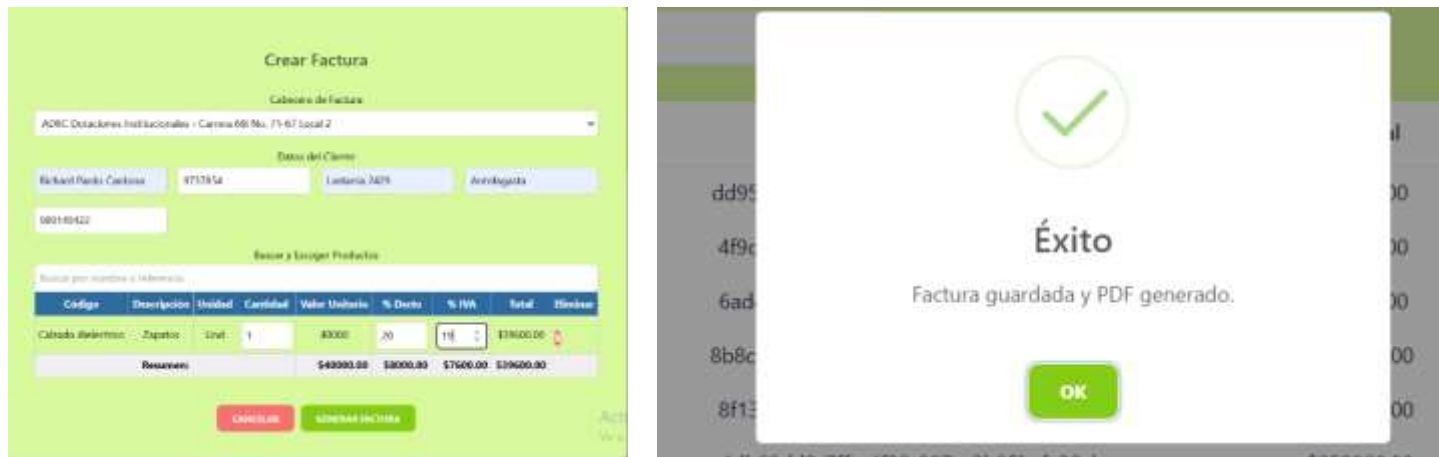
Sprint #	Duración	Sprint Goal (Meta del Sprint)	Historias seleccionadas	Tareas técnicas derivadas	Estimación
Sprint 3	07 al 19 de mayo de 2025	Implementar el módulo de facturación con selección de cabeceras y clientes.	HU013, HU014, HU015, HU016	Integrar componente ModalFactura. Autocompletado de clientes. Validación y selección dinámica de cabecera.	25 horas

Sprint Execution

Se creó el flujo completo de facturación inicial con cliente, cabecera y selección de productos. Se integró SweetAlert para confirmaciones e interacciones amigables.

Figura 45

Página de elaboración de factura y modal de factura creada



Fuente. Autoría Propia.

Sprint Review

Sprint #	Historias completadas	Demostración	Feedback recibido	Decisiones
Sprint 3	HU013, HU014, HU015, HU016	Se expuso la funcionalidad del módulo de facturación básica a través del componente FacturaModal.js, donde se permite seleccionar un cliente con autocompletado por NIT, elegir una cabecera de facturación, añadir productos dinámicamente y realizar cálculos automáticos. Se integraron alertas	Mejorar formularios y validación de campos numéricos.	Aplicar mejoras visuales al modal en el Sprint 4.

interactivas mediante
SweetAlert para validar
acciones clave durante la
creación de la factura.

Retrospectiva

- ¿Qué funcionó bien?
 - Interfaz fluida en modal y autocompletado.
- ¿Qué se puede mejorar?
 - Gestión del estado compartido entre productos.
- ¿Qué bloqueos se presentaron?
 - Retrasos en la visualización de cabeceras desde la API.

Sprint 4: Validaciones Finales, PDF y ajustes de stock***Sprint Planning***

Sprint #	Duración	Sprint Goal (Meta del Sprint)	Historias seleccionadas	Tareas técnicas derivadas	Estimación
Sprint 4	20 de mayo al 02 de julio de 2025	Completar la funcionalidad avanzada de facturación y validaciones finales.	HU017, HU018, HU019, HU020, HU021, HU022, HU023, HU024, HU025, HU026, HU027, HU028, HU029, HU030, HU031, HU032	Cálculos dinámicos. Validar stock disponible. Integrar jsPDF.	77 horas

Sprint Execution

El sistema quedó completamente funcional para facturar, generando documentos PDF con jsPDF y aplicando lógica contable completa.

Figura 46

Visualización de factura generada



Fuente. Autoría Propia.

Sprint Review

Sprint #	Historias completadas	Demostración	Feedback recibido	Decisiones
Sprint 4	HU017, HU018, HU019, HU020, HU021, HU022, HU023, HU024, HU025, HU026, HU027, HU028,	Se presentó la generación completa de una factura en formato PDF mediante la biblioteca jsPDF,	Solicitar personalización del PDF con logo de cabecera.	Se aplicó un diseño profesional y limpio al documento PDF.

HU029, HU030, HU031, HU032 incluyendo los cálculos de totales, descuentos, IVA y control de stock en tiempo real. El documento PDF incluye diseño personalizado con logotipo y datos de la empresa. Se evidenció la validación del stock disponible y el comportamiento del sistema ante múltiples productos seleccionados.

Retrospectiva

- ¿Qué funcionó bien?

- Cálculos financieros exactos y generación de PDF exitosa.
- ¿Qué se puede mejorar?
 - Manejo de errores ante desconexiones de red.
- ¿Qué bloqueos se presentaron?
 - Latencia en algunos cálculos del modal con múltiples productos.

Resultados

Los resultados obtenidos durante la ejecución del proyecto se organizaron según las fases del proceso de desarrollo del prototipo y se alinearon con los objetivos específicos planteados: análisis, diseño, desarrollo e implementación.

En primer lugar, se llevó a cabo un análisis detallado de los procesos de adquisición y venta del almacén ADIIC Dotaciones Institucionales S.A.S. mediante BPM. A través de entrevistas, observación directa y diagramas de flujo, se identificaron ineficiencias, tareas repetitivas y cuellos de botella en el manejo manual del inventario. Este diagnóstico permitió establecer los procesos actuales (AS-IS) y proponer una versión optimizada y automatizada (TO-BE), con el propósito de facilitar la trazabilidad, reducir errores y mejorar la sincronización entre áreas operativas y comerciales.

Con base en los requerimientos identificados, se diseñó una arquitectura tecnológica utilizando el modelo cliente-servidor de tres capas sustentado en la pila MERN (MERN Stack). Esta arquitectura separó adecuadamente las capas de presentación, lógica de negocio y persistencia de datos, y garantizó escalabilidad, mantenibilidad y eficiencia. Se elaboraron vistas arquitectónicas que describieron el contexto, la estructura funcional, el despliegue, la concurrencia y el desarrollo del sistema, así como diagramas que modelaron la interacción entre componentes.

Como resultado tangible, se construyó un prototipo funcional de aplicación web para la gestión de inventarios y ventas, siguiendo Scrum. El proceso inició con un Sprint 1 de configuración del entorno de desarrollo y continuó con varios sprints enfocados en la implementación progresiva de funcionalidades del Product Backlog. Se implementaron las siguientes funcionalidades principales:

- Registro y autenticación de usuarios, con roles diferenciados (administrador y vendedor).
- Gestión de categorías y productos, incluyendo creación, edición y visualización.
- Control automatizado de inventario, con verificación de stock al facturar.
- Creación y emisión de facturas, enlazadas con clientes y productos.
- Generación de documentos PDF para descarga y archivo de facturas.

El sistema fue implementado en entorno web, lo que facilitó el acceso remoto, la experiencia de uso y la eficiencia operativa.

Para validar el prototipo, se aplicaron pruebas funcionales y de usabilidad con usuarios finales de la empresa. Las pruebas funcionales confirmaron el desempeño del sistema según los requisitos establecidos, mientras que la evaluación de usabilidad mediante SUS arrojó una calificación promedio de 82 puntos, lo que indicó una alta aceptación del sistema en términos de facilidad de uso, comprensión de funcionalidades y satisfacción general.

Resumen de Cumplimiento por Módulo (MVP)

Se vinculó cada módulo con sus Historias de Usuario (HU), los objetivos específicos cubiertos y la evidencia de pruebas (unitarias, de integración y UAT). La columna “Resultado” resume el estado de cumplimiento y, cuando aplicó, se consignaron hallazgos y acciones de mejora.

Módulo / Componente	HUs incluidas (MVP)	Objetivo(s) específico(s)	Evidencia de prueba ejecutada	Resultado
Autenticación y Roles	HU001, HU002	OE3 (Implementación)	PU01 (login válido / contraseña)	Cumplido

			incorrecta); Colección Postman (auth); UAT: login por rol	
Gestión de	HU003,	OE3	PU02 (registro	Cumplido
Usuarios	HU004, HU005, HU006		válido / email duplicado); UAT: alta/edición, asignación de rol, RBAC visible	
Gestión de	HU007,	OE3	PU03 (creación	Cumplido
Categorías	HU008		válida / nombre vacío); UAT: crear/editar; validaciones front/back	
Gestión de	HU009,	OE3	PU04 (creación	Cumplido con
Productos	HU010, HU011, HU012		válida / stock negativo); UAT: CRUD + listado por categoría	hallazgos (reforzar validación $\text{stock} \geq 0$ en esquema/middleware)

Cabeceras de Factura	HU015, HU016, HU017	OE3	PU05 (creación válida / e-mail vacío); UAT: crear/editar/eliminar con confirmaciones	Cumplido
Facturación	HU020, HU021, HU022, HU023, HU024, HU025, HU026, HU027, HU029	OE3–OE4 (Implementación y Evaluación)	PU06 (factura válida / campo obligatorio vacío); Postman E2E (totales, IVA, PDF); UAT flujo completo	Cumplido (hallazgos resueltos: sincronización de totales/estado)
Gestión de Clientes	HU013, HU014, HU018, HU019, HU028, HU030, HU031, HU032	OE3	PU07 (registro válido / nombre vacío); UAT: búsqueda NIT/ID, historial, registro automático en 1. ^a compra	Cumplido

Leyenda de objetivos: OE1 Análisis; OE2 Diseño; OE3 Implementación; OE4 Evaluación/Usabilidad.

Consolidado de pruebas unitarias por módulo

Módulo	ID de Prueba(s)	Casos cubiertos	Resultado consolidado
Autenticación y Roles	PU01	Login válido; contraseña incorrecta (negativo)	2/2 OK (el negativo devuelve 401 como se espera)
Gestión de Usuarios	PU02	Registro válido; email duplicado (negativo)	2/2 OK
Gestión de Categorías	PU03	Creación válida; nombre vacío (negativo)	2/2 OK
Gestión de Productos	PU04	Creación válida; stock negativo (negativo)	2/2 OK, con hallazgo: reforzar chequeo stock ≥ 0 en esquema/middleware
Cabeceras de Factura	PU05	Creación válida; e-mail vacío (negativo)	2/2 OK

Facturación	PU06	Factura válida; campo obligatorio vacío (negativo)	2/2 OK
Gestión de Clientes	PU07	Registro válido; nombre vacío (negativo)	2/2 OK

Nota. Los casos “negativos” son satisfactorios cuando el sistema rechaza la operación con el código de estado esperado (400/401) y un mensaje claro.

Todos los módulos del MVP alcanzaron el estado “Cumplido”, con correcciones aplicadas a observaciones detectadas durante UAT y pruebas end-to-end (por ejemplo, sincronización de totales en facturación). La defensa en profundidad se reforzó con validaciones de entrada y respuestas HTTP adecuadas en casos negativos. Como acción preventiva, se recomienda formalizar el control de integridad “stock \geq 0” en el esquema de productos (además de validaciones en UI/controladores) para homogeneizar garantías en todas las capas.

Conclusiones y Recomendaciones

Conclusiones

Respecto al primer objetivo específico, consistente en analizar la gestión del inventario y el proceso de ventas en la empresa ADIIC Dotaciones Institucionales, se identificaron cuellos de botella, redundancias y tareas manuales que afectaban la eficiencia operativa. El levantamiento de información permitió modelar los procesos actuales (AS-IS) y proponer procesos optimizados (TO-BE). Una dificultad encontrada en esta etapa fue la resistencia inicial de algunos colaboradores para describir con detalle sus tareas, lo que retrasó parcialmente la caracterización de procesos. No obstante, este obstáculo se superó mediante entrevistas adicionales y triangulación de información con documentos internos.

En relación con el segundo objetivo específico, enfocado en diseñar la arquitectura de solución del sistema, se definió una estructura cliente-servidor de tres capas sustentada en la pila MERN (MERN Stack). Esta arquitectura garantizó escalabilidad, mantenibilidad y modularidad, respondiendo a los requerimientos de la empresa. Sin embargo, se presentaron dificultades técnicas al integrar librerías de autenticación y seguridad, lo que requirió investigación adicional y ajustes de diseño. Estas dificultades evidencian la necesidad de prever más tiempo para la validación técnica en fases iniciales de proyectos similares.

Respecto al tercer objetivo específico, orientado al desarrollo del sistema de gestión de inventario y ventas, se construyó un prototipo funcional bajo la metodología ágil Scrum. El trabajo iterativo facilitó la validación con usuarios finales, alcanzando un promedio de 82 puntos en la escala SUS, lo que refleja una alta aceptación. Una de las principales dificultades en esta etapa fue la limitación de recursos técnicos disponibles para pruebas en entornos de producción simulados, lo cual restringió la validación del sistema únicamente a un entorno controlado.

Además, algunos usuarios presentaron dificultades iniciales para adaptarse al uso de la aplicación web, lo que demuestra la necesidad de capacitación complementaria.

Finalmente, se concluye que, a pesar de las dificultades mencionadas, el prototipo logró cumplir con los objetivos planteados, aportando a la eficiencia operativa de ADIIC Dotaciones Institucionales y fortaleciendo las competencias del equipo desarrollador. Estas dificultades constituyen aprendizajes valiosos que enriquecen tanto la experiencia académica como la capacidad de afrontar proyectos futuros.

Con el fin de garantizar la disponibilidad y continuidad del desarrollo, el código fuente y la documentación técnica se encuentran alojados en repositorios públicos en GitHub, bajo licencia MIT, lo que permite su uso, adaptación y mejora por parte de la comunidad:

- Frontend: https://github.com/jcarrieta1974/Front_ADIIC
- Backend: https://github.com/jcarrieta1974/Back_ADIIC

Recomendaciones

En correspondencia con los resultados obtenidos y las dificultades encontradas durante el desarrollo del prototipo, se plantean las siguientes recomendaciones:

Sobre el análisis de procesos: Considerando la resistencia inicial de algunos colaboradores, se recomienda implementar talleres participativos de levantamiento de procesos en futuros proyectos, para lograr mayor compromiso y precisión desde las primeras fases.

En relación con el diseño de la arquitectura: Ante las dificultades de integración de librerías y medidas de seguridad, se sugiere planificar etapas de prueba piloto específicas para validar compatibilidades técnicas, así como documentar buenas prácticas de seguridad desde el diseño inicial.

Respecto al desarrollo del sistema: Para superar las limitaciones asociadas a la falta de entornos de prueba reales, se aconseja destinar recursos adicionales a infraestructura de pruebas y contemplar un plan de despliegue gradual en escenarios de operación controlada.

Sobre la usabilidad y adopción por parte de usuarios: Se recomienda incluir un plan de capacitación inicial y continua que acompañe la implementación del sistema, facilitando la adaptación de los usuarios y reduciendo la resistencia al cambio tecnológico.

En cuanto a la proyección futura del sistema: Se sugiere priorizar la implementación de módulos de facturación electrónica y reportes analíticos, así como la integración con proveedores. Estas mejoras reforzarán la utilidad del sistema y ampliarán su impacto organizacional.

Desde la perspectiva académica: Documentar las lecciones aprendidas y publicar los hallazgos permitirá que las dificultades enfrentadas y las soluciones implementadas sirvan como referencia para futuros proyectos de grado o investigaciones en el ámbito de tecnologías web aplicadas a pymes.

Referencias bibliográficas

- Abbas, A., Wahid Syed, A., Saqib Ahmed, M., Sagheer, A., & Kumar H, V. (2022). E-Shop Website Using MERN Stack. *International Journal for Research in Applied Science & Engineering Technology*, 10(IV), 1840–1845.
<https://doi.org/https://doi.org/10.23857/pc.v6i3.2382>
- Arenal Laza, C. (2020). *Gestion de inventarios: UF0476*. Editorial Tutor Formación.
<https://elibro-net.bibliotecavirtual.unad.edu.co/es/ereader/unad/126745?page=3>
- Bass, L., Clements, P., & Kazman, R. (2021). *Software Architecture in Practice* (4th ed.). Addison-Wesley Professional.
- Capece, C. T., Der Boghosian, R., Romero, C. E., Pablo Hazan, P., & Colque, J. (2021). *GotenJS: Framework para generar la estructura base de una aplicación web*. SEDICI Repositorio Institucional de La UNLP, 27–36. <https://sedici.unlp.edu.ar/handle/10915/140261>
- Chandrashekhar Bakale, S., Monish Naware, A., Vilas Parab, S., Anil Saxena, S., & Atul Anjaria, S. (2023). MERN Stack-Based Car Rental Website Development. *International Journal of Advanced Research in Computer Science*, 14.
<https://www.ijarcs.info/index.php/Ijarcs/article/view/6971>
- Colombia. Dirección de Impuestos y Aduanas Nacionales (DIAN). (2023, 1 de noviembre). Resolución 000165 de 2023: Por la cual se desarrolla el sistema de facturación, se adoptan la versión 1.9 del anexo técnico de factura electrónica de venta y el anexo técnico 1.0 del documento equivalente electrónico, y se dictan otras disposiciones.
<https://www.dian.gov.co/normatividad/Normatividad/Resoluci%C3%B3n%20000165%20de%2001-11-2023.pdf> Dian

Colombia. Ministerio de Trabajo y Seguridad Social & Ministerio de Salud. (1989, 31 de marzo).

Resolución 1016 de 1989: Por la cual se reglamenta la organización, funcionamiento y forma de los programas de salud ocupacional que deben desarrollar los patronos o empleadores en el país.

<https://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=5412> Alcaldía de Bogotá

Colombia. Ministerio de Trabajo y Seguridad Social. (1979, 22 de mayo). Resolución 2400 de 1979: Por la cual se establecen algunas disposiciones sobre vivienda, higiene y seguridad en los establecimientos de trabajo.

<https://www.bogotajuridica.gov.co/sisjur/normas/Norma1.jsp?i=53565>

Colombia. Ministerio del Trabajo. (2017, 27 de marzo). Resolución 1111 de 2017: Por la cual se definen los Estándares Mínimos del Sistema de Gestión de Seguridad y Salud en el Trabajo para empleadores y contratantes.

<https://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=62506>

Colombia. Unidad Administrativa Especial Dirección de Impuestos y Aduanas Nacionales (DIAN). (2020, 5 de mayo). Resolución 000042 de 2020: Por la cual se desarrollan los sistemas de facturación, los proveedores tecnológicos, el registro de la factura electrónica de venta como título valor, se expide el anexo técnico de factura electrónica de venta y se dictan otras disposiciones en materia de sistemas de facturación.

https://normograma.dian.gov.co/dian/compilacion/docs/resolucion_dian_0042_2020.html

Congreso de la República de Colombia. (1979, 24 de enero). Ley 9 de 1979: Por la cual se dictan medidas sanitarias. Diario Oficial 35.308.

https://www.secretariasenado.gov.co/senado/basedoc/ley_0009_1979.html Secretaría del

Senado

Congreso de la República de Colombia. (1990, 28 de diciembre). Ley 50 de 1990: Por la cual se introducen reformas al Código Sustantivo del Trabajo y se dictan otras disposiciones. Diario Oficial 39.618. <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=281>
Función Pública J.A. Del Río

Congreso de la República de Colombia. (2002, 27 de diciembre). Ley 789 de 2002: Por la cual se dictan normas para apoyar el empleo y ampliar la protección social y se modifican algunos artículos del Código Sustantivo del Trabajo. Diario Oficial 45.046.
<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=6778> Función Pública Alcaldía de Bogotá

Congreso de la República de Colombia. (2009, 13 de julio). Ley 1314 de 2009: Por la cual se regulan los principios y normas de contabilidad e información financiera y de aseguramiento de información aceptados en Colombia. Diario Oficial 47.409.
<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=36833> Función Pública

Congreso de la República de Colombia. (2012, 11 de julio). Ley 1562 de 2012: Por la cual se modifica el Sistema de Riesgos Laborales y se dictan otras disposiciones en materia de salud ocupacional. Diario Oficial 48.488.
https://www.secretariassenado.gov.co/senado/basedoc/ley_1562_2012.html Secretaría del Senado

Congreso de la República de Colombia. (2012, 17 de octubre). Ley 1581 de 2012: Por la cual se dictan disposiciones generales para la protección de datos personales. Diario Oficial 48.587.
<http://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=34488>

Díaz Pereira, A. (2020). Desarrollo de Software Web para la gestión de inventario, con aplicación móvil de consulta para la Empresa Cosdy S.A.S [Trabajo de grado, Universidad Distrital Francisco José de Caldas]. <https://doi.org/http://hdl.handle.net/11349/24811>

Dirección de Impuestos y Aduanas Nacionales (DIAN). (2023, 1 de noviembre). Resolución 000165 de 2023, Por la cual se desarrolla el sistema de facturación, los proveedores tecnológicos, se adopta la versión 1.9 del anexo técnico de factura electrónica de venta, se expide el anexo técnico 1.0 del documento equivalente electrónico, y se dictan otras disposiciones en materia del sistema de facturación.
<https://www.dian.gov.co/normatividad/Normatividad/Resoluci%C3%B3n%20000165%20de%2001-11-2023.pdf>

González Clavijo, C. C., Correa Rodríguez, A. de J., Cuervo Diaz, N. A., & Vásquez Bravo, D. M. (2022). Arquitectura de solución de una aplicación web para gestionar ejercicios de autoevaluación en una institución de educación superior a distancia y virtual, usando puntos de vistas. Iksad Publishing House, Cap. 18, 251–274. <https://iksadyayinevi.com/wp-content/uploads/2022/06/CURRENT-TOPICS-IN-SOCIAL-AND-ADMINISTRATIVE-SCIENCES.pdf>

González Miller, C. J. (2024). Implementación de aplicación móvil para la gestión de inventarios en el restaurante moros los prado de la ciudad de jipijapa. [Proyecto de titulación, Universidad Estatal del Sur de Manabí].
<http://repositorio.unesum.edu.ec/handle/53000/6371>

Herrero Romero, S. (2022). Diseño e implementación de una aplicación web full stack para la organización y gestión de equipos. E.T.S.I. y Sistemas de Telecomunicación (UPM).
<https://oa.upm.es/72291/>

- Izaguirre Velásquez, C. V., Sabino Blas, C. B., Villar Tiravanti, L. M., & Quiliche Castellares, R. M. (2022). Gestión de inventarios para incrementar la productividad en una empresa agrícola. *INGnosis*, 8(2), 8–18. <https://doi.org/https://doi.org/10.18050/ingnosis.v8i2.2475>
- Joyanes Aguilar, L. (2020). *Fundamentos de Programación. Algoritmos, estructura de datos y objetos* (Quinta edición). McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S. A. U.
- Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations* (1st ed.). IT Revolution Press.
- http://images.itrevolution.com/documents/DevOps_Handbook_Intro_Part1_Part2.pdf
- Ladrón de Guevara, M. Á. (2020). Gestión de inventarios. UF0476 (Miguel Ángel Ladrón Jiménez, Ed.). Tutor Formación.
- Larman, C. (2003). *Agile and Iterative Development: A Manager's Guide*. Pearson Education.
- <https://archive.org/details/agileiterativede0000larm>
- Lazo Díaz, D. L., Rodríguez García, J. I., Amador Ramos, J. E., Ramírez Echavarría, Z. G., & Lozano Montero, E. (2021). Economía digital, los efectos socioeconómicos en las MiPyMEs mexicanas. *Jóvenes En La Ciencia*, 10.
- <https://www.jovenesenlaciencia.ugto.mx/index.php/jovenesenlaciencia/article/view/3418>
- Robert M. McClure. (2001). NATO SOFTWARE ENGINEERING CONFERENCE 1968.
- Mera Rodriguez, M. A., Arauz Zambrano, J., Palacios Acosta, I., & Zambrano Cedeño, C. (2019). Aplicación web para gestionar los procesos administrativos de las Asociaciones Agropecuarias del cantón Chone de la provincia de Manabí. *Sinapsis La Revista Científica Del ITSUP*, 2 (15). <https://dialnet.unirioja.es/servlet/articulo?codigo=7471194>

- Mex Álvarez, R. M. de J., Yanez Nava, D., Garma Quen, P. M., Guillen Morales, M. M., Guillermo Chuc, G., & Novelo Pérez, M. I. (2024). Diseño y aplicación de un programa informático para la gestión de inventarios en un Laboratorio de Farmacia. *Revista Latinoamericana de Ciencias Sociales y Humanidades*, 5 (1), 540–550.
<https://doi.org/https://doi.org/10.56712/latam.v5i1.1607>
- MongoDB. (2024). MERN Stack Explained.
<https://www.mongodb.com/resources/languages/mern-stack#:~:Text=MERN%20stands%20for%20MongoDB%2C%20Express,A%20client%2Dside%20JavaScript%20framework.>
- Moragas Pardell, T. (2023). Desenvolupament d'una Web App per a gestionar objectes perduts [TREBALL FI DE GRAU, Universitat Politècnica de Catalunya].
<http://hdl.handle.net/2117/384460>
- Navarrete Lozada, E. G. (2019). Importancia de la Gestión de Inventario en las Empresas. *Revista de Investigación Formativa: Innovación y Aplicaciones Técnico - Tecnológicas*, 1 (1). <https://doi.org/https://ojs.formacion.edu.ec/index.php/rei/article/view/v1.n1.a6>
- Ortíz Fernández, J., Baldeón Tovar, M., Medina Pelaiza, L., Ortíz Huamán, C., & Godiño Poma, M. (2024). Gestión por procesos en las empresas. Una revisión sistémica. *Gestionar: Revista de Empresa y Gobierno*, 4(Artículo de revisión), 7–22.
<https://doi.org/https://doi.org/10.35622/j.rg.2024.01.001>
- Pinzón Núñez, S., Rodríguez Guerrero, R., & Vanegas, C. A. (2019). Java y el patrón Modelo-Vista-Controlador (MVC) (E. Pardo Salazar, Ed.; 1st ed.). Editorial UD.
https://books.google.com.co/books?id=5FS_EAAAQBAJ&lpg=PA1&hl=es&pg=PA12#v=onepage&q=DOI&f=false

Prajapati, A., Khedle, P., Malvi, S., & Malviya, P. (2023). Development of an E-Commerce Web App System through Implementation of the MERN Stack. *International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences*, 11(1).
<https://www.ijirms.org/research-paper.php?id=230047>

Presidencia de la República de Colombia. (1989, 30 de marzo). Decreto Ley 624 de 1989: Por el cual se expide el Estatuto Tributario de los impuestos administrados por la Dirección General de Impuestos Nacionales. *Diario Oficial* 38.756.
<https://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=6533> Alcaldía de Bogotá
 Ministerio de Comercio

Presidencia de la República de Colombia. (1993, 29 de diciembre). Decreto 2649 de 1993: Por el cual se reglamenta la contabilidad en general y se expiden los principios o normas de contabilidad generalmente aceptados en Colombia. *Diario Oficial* 41.156.
<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=9863> Función Pública

Presidencia de la República de Colombia. (2013, 27 de junio). Decreto 1377 de 2013: Por el cual se reglamenta parcialmente la Ley 1581 de 2012. *Diario Oficial* 48.834.
<http://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=49981>

Presidencia de la República de Colombia. (2015, 20 de febrero). Decreto 302 de 2015: Por el cual se reglamenta la Ley 1314 de 2009 sobre el marco técnico normativo para las normas de aseguramiento de la información. *Diario Oficial* 49.431.
<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=60905> Función PúblicaSIDN

Presidencia de la República de Colombia. (2015, 26 de mayo). Decreto 1072 de 2015: Por medio del cual se expide el Decreto Único Reglamentario del Sector Trabajo. Diario Oficial 49.523. https://www.icbf.gov.co/cargues/avance/compilacion/docs/decreto_1072_2015.htm
ICBF

Presidencia de la República de Colombia. (2015, 24 de noviembre). Decreto 2242 de 2015: Por el cual se reglamentan las condiciones de expedición e interoperabilidad de la factura electrónica con fines de masificación y control fiscal. Diario Oficial 49.706.
<https://www.suin-juriscal.gov.co/viewDocument.asp?ruta=Decretos/30033048>
SIDNJuriscal

Presidencia de la República de Colombia. (2017, 12 de enero). Decreto 52 de 2017: Por medio del cual se modifica el artículo 2.2.4.6.37 del Decreto 1072 de 2015, sobre la transición para la implementación del Sistema de Gestión de la Seguridad y Salud en el Trabajo (SG-SST). Diario Oficial 50.114.
<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=78813> Función Pública

Pressman, R. S., & Maxim, B. R. (2021). *SOFTWARE ENGINEERING: A PRACTITIONER'S APPROACH* (9th ed.). McGraw-Hill.

Richards, M., & Ford, N. (2020). *Fundamentals of Software Architecture* (C. Guzikowski, Ed.; First Edition). O'Reilly Media, Inc.

Rolón Ramírez, D. A. (2024). Transformación Tecnológica en el Modelo de Gestión de Inventarios en las MiPymes, *Revisión Bibliográfica. Ciencia Latina Revista Científica Multidisciplinar*, 8 (1), 3551–3565.
https://doi.org/https://doi.org/10.37811/cl_rcm.v8i1.9701

- Rozanski, N., & Woods, E. (2012). *Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives* (2nd ed.). Pearson Education, Inc.
- Satpathy, T. (2022). *Guía de los fundamentos de Scrum* (4th ed.). SCRUMstudyTM.
<https://scrumstudy.mx/descarga-la-4ta-edicion-2023-de-la-guia-sbok-scrumstudy-book-of-knowledge-en-espanol/#:~:text=La%20Gu%C3%ADa%20SBOK%C2%AE%20fue%20desarrollada%20como%20un%20medio,pero%20desean%20mejorar%20e%20incrementar%20madurez%20e%20Agilidad.>
- SCRUMstudy. (2022). *Guía de los fundamentos de Scrum (Guía del SBOK®)* (4 edición). SCRUMstudyTM (VMedu, Inc.). <https://www.scrumstudy.com/sbokguide/download-free-buy-sbok#new-sbok-download>
- Soledispa-Rodríguez, X. E., Álvarez-Rivera, I. Y., Baque-Suárez, A. S., & Cantos-Castro, B. A. (2021). Tecnologías de información y comunicación en la gestión empresarial de las pymes. *Polo Del Conocimiento*, 6 (3), 476–496.
- Sommerville, I. (2020). *Software Engineering* (M. Horton, M. Hirsch, M. Goldstein, C. Bell, M. Borthakur, B. Roy, & J. Holcomb, Eds.; 10th ed.). 2021.
- Storey, M.-A., Hoda, R., Paz Milani, A. M., & Baldassarre, M. T. (2025). *Guiding Principles for Mixed Methods Research in Software Engineering*. *Empirical Software Engineering*.
<https://arxiv.org/abs/2404.06011>
- Teja Labba, S., Sharfuddin, M., Sai Praveen, Z., Sujitha, B., & Reddy, D. D. (2023). COMPREHENSIVE ANALYSIS OF WEB APPLICATION DEVELOPMENT USING MERN STACK. *International Journal of Creative Research Thoughts - IJCRT* (IJCRT.ORG), Vol. 11(7), b122–b130. https://ijcrt.org/viewfull.php?&p_id=IJCRT2307133

Timarán Pereira, R., Chaves Torres, A., Checa Mora, J. C., Jiménez Toledo, J., & Ordóñez Erazo, H. (2009). *Introducción a la Programación con Scheme*.

Tovar-Cardozo, D., & Sierra-García, W. D. (2021). Propuesta de diseño de una aplicación móvil para la gestión y control de inventarios en la empresa Deluxe Business Group [Trabajo de grado, Universidad Católica de Colombia].

<https://repository.ucatolica.edu.co/entities/publication/d8527d0e-2772-4bdb-8e8e-1007359182f5>

Vásquez Ramírez, A. A., Guanuchi Orellana, L. M., Cahuana Tapia, R. D., Vera Treves, R., & Holgado Tisoc, J. (2023). *Métodos de Investigación Científica* (W. Sucari, P. Aza, & A. Flores, Eds.; 1 edición digital). Instituto Universitario de Innovación Ciencia y Tecnología Inudi Perú S.A.C. <https://doi.org/10.35622/inudi.b.094>

Anexos

Anexo 1

Caso de negocio (Business Case)

El documento completo del caso de negocio puede ser consultado en el siguiente enlace:

[Caso de negocio](#)

Anexo 2*Visión del proyecto*

El documento completo de la visión del proyecto puede ser consultado en el siguiente enlace: [Visión del proyecto](#)

Anexo 3*Acta de constitución de proyecto*

El documento completo del acta de constitución del proyecto puede ser consultado en el siguiente enlace: [Acta de constitución del proyecto](#)

Anexo 4*Formulario para entrevista de toma de requisitos*

El formulario completo utilizado para las entrevistas de toma de requisitos puede ser consultado en el siguiente enlace: [Formulario entrevistas toma de requisitos](#)

Anexo 5*Formulario pruebas unitarias proyecto ADIIC*

El formulario completo de las pruebas unitarias aplicadas al proyecto ADIIC puede ser consultado en el siguiente enlace: [Formulario pruebas unitarias proyecto ADIIC](#)

Anexo 6*Desarrollo del proyecto por épicas*

El documento completo con el desarrollo del proyecto por épicas puede ser consultado en el siguiente enlace: [Desarrollo del proyecto por épicas](#)

Anexo 7*Certificado de registro de soporte lógico – software*

El certificado de registro de soporte lógico – software puede ser consultado en el siguiente enlace: [Certificado de registro de soporte lógico – software](#)

Anexo 8*Manual de usuario versión 1.0*

El manual de usuario versión 1.0 puede ser consultado en el siguiente enlace: [Manual de usuario versión 1.0](#)