

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI



FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES

CARRERA DE COMPUTACIÓN

Tema: “Herramienta tecnológica para la gestión de inventario”

Trabajo de Integración Curricular previo a la obtención del
título de Ingeniero en Ciencias de la Computación

AUTOR: Guzmán Imbaquingo Bryan Gerardo

TUTOR: Ing. Miranda Realpe Jorge Humberto, Msc.

Tulcán, 2025.

CERTIFICADO DEL TUTOR

Certifico que el estudiante Guzmán Imbaquingo Bryan Gerardo con el número de cédula 1753155207 respectivamente ha desarrollado el Trabajo de Integración Curricular: "Herramienta tecnológica para la gestión de inventario"

Este trabajo se sujeta a las normas y metodología dispuesta en el Reglamento de la Unidad de Integración Curricular, Titulación e Incorporación de la UPEC, por lo tanto, autorizo la presentación de la sustentación para la calificación respectiva

Ing. Miranda Realpe Jorge Humberto, Msc.

TUTOR

Tulcán, agosto de 2025

AUTORÍA DE TRABAJO

El presente Trabajo de Integración Curricular constituye un requisito previo para la obtención del título de Ingeniero en la Carrera de computación de la Facultad de Industrias Agropecuarias y Ciencias Ambientales

Yo, Guzmán Imbaquingo Bryan Gerardo con cédula de identidad número 1753155207 respectivamente declaro que la investigación es absolutamente original, auténtica, personal y los resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.



Guzmán Imbaquingo Bryan Gerardo

AUTOR

Tulcán, agosto de 2025

ACTA DE CESIÓN DE DERECHOS DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Yo Guzmán Imbaquingo Bryan Gerardo declaro ser autor de los criterios emitidos en el Trabajo de Integración Curricular: "Herramienta tecnológica para la gestión de inventario" y eximo expresamente a la Universidad Politécnica Estatal del Carchi y a sus representantes de posibles reclamos o acciones legales.



Guzmán Imbaquingo Bryan Gerardo

AUTOR(A)

Tulcán, agosto de 2025

AGRADECIMIENTO

Quiero expresar mi más sincero agradecimiento a la Universidad Politécnica Estatal del Carchi por abrirme las puertas al conocimiento y por ofrecerme un entorno académico enriquecedor que ha sido esencial para mi formación profesional. Agradezco profundamente a sus docentes por su dedicación y a toda la comunidad universitaria por contribuirme una experiencia educativa integral y significativa.

En especial, deseo manifestar mi gratitud al MSc. Jorge Miranda, tutor de esta tesis, por su constante acompañamiento, orientación y apoyo durante el desarrollo de este trabajo. Su compromiso, disposición y guía experta han sido pilares fundamentales para alcanzar los objetivos propuestos y concluir exitosamente esta investigación.

DEDICATORIA

Dedico este logro con todo mi amor y gratitud a mis padres, Elena Imbaquingo y Gerardo Guzmán, por ser mi mayor inspiración en mi vida y por brindarme su apoyo incondicional en cada etapa de este camino. Su esfuerzo, sacrificio y enseñanzas han sido una guía constante que me ha impulsado a superarme y a no rendirme ante las dificultades. Gracias por creer en mí, recordarme de la importancia de luchar por mis sueños y por ser mi mayor ejemplo de responsabilidad, perseverancia y amor.

También agradezco de corazón a mis hermanos, Andy y Gael Guzmán, por su compañía, alegría y motivación inquebrantable. Su presencia ha sido una fuente de fuerza y energía en los momentos más exigentes, y su cariño me ha recordado siempre la importancia de seguir adelante con determinación y esperanza.

ÍNDICE

RESUMEN	12
ABSTRACT	13
INTRODUCCIÓN	14
I. EL PROBLEMA	16
1.1. PLANTEAMIENTO DEL PROBLEMA	16
1.2. FORMULACIÓN DEL PROBLEMA	17
1.3. JUSTIFICACIÓN	17
1.4. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN	18
1.4.1. Objetivo General	18
1.4.2. Objetivos Específicos	19
1.4.3. Preguntas de Investigación.....	19
II. FUNDAMENTACIÓN TEÓRICA	20
2.1. ANTECEDENTES DE LA INVESTIGACIÓN	20
2.2. MARCO TEÓRICO	21
2.2.1 Gestión de inventario	21
2.2.1.1. Sistema de gestión de inventario	22
2.2.1.2. Facturación automatizada	22
2.2.1.3. Herramienta tecnológica	22
2.2.1.4. Ingeniería de Software	22
2.2.2. Metodologías ágiles	22
2.2.2.1. Metodología XP	23
2.2.2.2. Metodología Scrum	23
2.2.2.3. Metodologías tradicionales.....	23
III. METODOLOGÍA	29
3.1. ENFOQUE METODOLÓGICO	29

3.1.1. Enfoque Cualitativo	29
3.1.2. Enfoque cuantitativo.....	29
3.1.2. Tipo de Investigación	30
3.1.2.1. Investigación descriptiva	30
3.1.2.2. Investigación explicativa	30
3.1.2.3. Investigación documental	30
3.2. IDEA A DEFENDER	31
3.3. DEFINICIÓN Y OPERACIONALIZACIÓN DE LAS VARIABLES	31
3.3.1.1. Definición variable Independiente	31
3.4. MÉTODOS UTILIZADOS	35
3.4.1. Método Inductivo	35
3.4.2. Técnicas	35
IV. RESULTADOS Y DISCUSIÓN	36
4.1. RESULTADOS.....	36
4.1.1. Resultados de la entrevista.....	36
4.2. PROPUESTA	39
4.2.1 Estudio de factibilidad	39
4.2.1.1 Factibilidad organizacional.....	39
4.2.1.1.1. Factibilidad técnica	40
4.2.1.1.3. Factibilidad Económica.....	41
4.2.1.1.4. Factibilidad Operativa	41
4.2.2. Metodología Scrum.....	43
4.2.2.1. Fase 1 Inicio (Preparación del Proyecto)	43
4.2.2.2. Fase 2 Planificación y Estimación	45
4.2.2.3 Fase 3 Implementación	50
V. CONCLUSIONES Y RECOMENDACIONES.....	82
5.1. CONCLUSIONES.....	82
5.2. RECOMENDACIONES.....	83

VI. REFERENCIAS BIBLIOGRÁFICAS	84
VII. ANEXOS.....	87

ÍNDICE DE FIGURAS

Figura 1. Diseño de interfaz.....	50
Crear los datos en la interfaz de productos para el registro de instrumentos.	50
Figura 2. Interfaz registro de instrumentos	50
Figura 3. Instrumentos registrados.....	51
Figura 4. Registro de instrumentos en la base de datos.....	51
4.2.2.3.2. Desarrollo Sprint 2.....	51
Figura 5. Registro de entradas de instrumentos (cuando llegan al inventario).	52
Figura 6. Registro de salidas (cuando un instrumento se vende o se retira del inventario).	52
Figura 7. Implementación del modelo de datos para registrar movimientos de inventario.....	53
Figura 8. lógica para actualizar el stock total en tiempo real.	53
Figura 9. lógica para actualizar el stock total en tiempo real.	54
Figura 10. Validación de movimientos (por ejemplo, no permitir salidas si no hay suficiente stock).....	54
Figura 11. Conexión con la base de datos para guardar los registros de entradas/salidas.....	54
Figura 12. Reporte de inventario.....	55
Figura 13. Reportes de entradas y salidas de inventario.	55
Figura 14. Diseño de la interfaz para visualizar y generar reportes.	56
Figura 15. Generación de reportes visuales (tablas, PDF.).	56
Figura 16. Generación de reportes visuales (tablas, PDF.).	57
Figura 17. Posibilidad de configurar el umbral mínimo de stock por producto.	57

Figura 18. Visualización de alertas en el panel administrativo cuando el inventario de un instrumento alcance un nivel mínimo preestablecido.	58
4.2.2.3.5. Desarrollo Sprint 5.....	58
Figura 19. Generación automática de comprobantes electrónicos al registrar ventas	58
Figura 20. Sincronización entre ventas, inventario y facturación	59
Figura 21. Base de datos.....	59
Codificación	60
Figura 22. Función para mostrar un resumen de todo lo que realiza el sistema de inventario.....	60
Figura 23. Función para crear un usuario	61
Figura 24. Función para crear una categoría	61
ProductoResource.php	61
Figura 25. Función para crear un producto	62
Figura 26. Función para crear un producto	62
Figura 27. Función para crear un stock de cualquier instrumento que este registrado en productos	63
MovimientoInventarioResource.php.....	64
Figura 28. Función para crear un nuevo movimiento de inventario.....	64
Figura 30. Función para crear un nuevo movimiento de inventario.....	65
Figura 31. Función para crear un registro de un nuevo proveedor con todos los datos de la empresa	66
Figura 32. Función para crear un registro de un nuevo proveedor con todos los datos de la empresa	67
IngresoBodegaResource.php	68
Figura 33. Función para registrar un ingreso a bodega con todos los datos de la persona que recibió el pedido	68
Figura 34. Función para crear un pedido de reposición con todos los datos del proveedor cuando el stock de un instrumento este bajo	69

Figura 35. Detalles del instrumento que desea reponer	69
TipoMovimientoResource.php	70
Figura 36. Función para crear un tipo de movimiento.....	70
EstadoPedidoResource.php.....	70
Figura 37. Función para crear el estado del pedido.....	70
Figura 38. Función para crear y registrar los datos del cliente, detalle de la venta, detalles del producto, para ver si hay stock disponible y para calcular el subtotal, IVA y precio total.....	71
Figura 39. Función para crear y registrar los datos del cliente, detalle de la venta, detalles del producto, para ver si hay stock disponible y para calcular el subtotal, IVA y precio total.....	71
Figura 40. Función para crear y registrar los datos del cliente, detalle de la venta, detalles del producto, para ver si hay stock disponible y para calcular el subtotal, IVA y precio total.....	72
Figura 41. Función para crear y registrar los datos del cliente, detalle de la venta, detalles del producto, para ver si hay stock disponible y para calcular el subtotal, IVA y precio total.....	72
Figura 42. Función para crear y registrar los datos del cliente, detalle de la venta, detalles del producto, para ver si hay stock disponible y para calcular el subtotal, IVA y precio total.....	73

ÍNDICE DE ANEXOS

Anexo 1. Acta de la sustentación de Predefensa del TIC	87
Anexo 2. Certificado del abstract por parte de idiomas.....	88

RESUMEN

El objetivo principal de esta investigación fue implementar una herramienta tecnológica para optimizar la gestión de inventario en el almacén DK'PO, ubicado en Cayambe. Se identificó como problema central el uso de hojas físicas para el control de inventario, lo que generaba dificultades como desconocimiento de existencias, errores en pedidos, retrasos en reabastecimiento, pérdidas económicas y falta de control sobre robos o descuentos mal aplicados. Estas fallas afectaban directamente la rentabilidad del negocio, como lo evidencian estudios previos sobre la importancia de una gestión adecuada del inventario. Para abordar esta problemática se utilizó un enfoque metodológico mixto. En el enfoque cualitativo se realizaron entrevistas al propietario del almacén para comprender sus necesidades, mientras que en el cuantitativo se establecieron indicadores como la cantidad de equipos registrados correctamente, tasa de errores y nivel de inactividad. La metodología de desarrollo empleada fue Scrum, organizada en cuatro sprints. El sistema fue construido con PHP, Laravel, Filament, JavaScript y MySQL, y desarrollado en Visual Studio Code. Entre los beneficios alcanzados destacan la reducción de errores en los registros, control exacto del stock, generación automática de reportes, alertas de bajo inventario y mejora en la atención al cliente. La interfaz del sistema fue diseñada para ser intuitiva y funcional, facilitando el registro de instrumentos, control de entradas y salidas, así como la generación de reportes estratégicos. Las pruebas de aceptación confirmaron que el sistema cumplió con los requisitos establecidos, demostrando su efectividad y viabilidad. Se concluye que la herramienta tecnológica desarrollada optimizó la gestión de inventario en el almacén DK'PO, y que posee potencial para futuras ampliaciones, como la inclusión de módulos de ventas o facturación electrónica, fortaleciendo aún más la toma de decisiones del negocio.

Palabras Claves: gestión de inventario, Laravel Filament, metodología Scrum.

ABSTRACT

The main objective of this research was to implement a technological tool to optimize inventory management at DK'PO warehouse, located in Cayambe. The central problem identified was the use of physical sheets for inventory control, which generated difficulties such as lack of knowledge of stock levels, order errors, restocking delays, economic losses, and lack of control over theft or improperly applied discounts. These issues directly affected the business's profitability, as evidenced by previous studies on the importance of proper inventory management. To address this problem, a mixed-method approach was applied. In the qualitative approach, interviews were conducted with the warehouse owner to understand their needs, while in the quantitative approach, indicators such as the number of items correctly registered, error rates, and downtime levels were established. The development methodology used was Scrum, organized into four sprints. The system was built using PHP, Laravel, Filament, JavaScript, and MySQL, and developed in Visual Studio Code. The benefits achieved include a reduction in registration errors, accurate stock control, automatic report generation, low inventory alerts, and improved customer service. The system interface was designed to be intuitive and functional, facilitating the registration of items, control of incoming and outgoing stock, as well as the generation of strategic reports. Acceptance testing confirmed that the system met the established requirements, demonstrating its effectiveness and feasibility. It is concluded that the technological tool developed optimized inventory management at DK'PO warehouse and has potential for future enhancements, such as the inclusion of sales or electronic invoicing modules, further strengthening business decision-making.

Keywords: inventory management, Laravel Filament, Scrum methodology.

INTRODUCCIÓN

En la actualidad, las herramientas tecnológicas de inventario representan una parte principal en la estructura operativa de las empresas, evolucionando desde simples registros manuales hacia plataformas digitales integradas que facilitan el control, seguimiento y análisis en tiempo real de los productos almacenados.

Sin embargo, muchos negocios, especialmente pequeñas y medianas empresas como el almacén musical DK'PO en Cayambe, continúan realizando métodos manuales para gestionar sus inventarios enfrentando problemas significativos que afectan su funcionamiento. La ausencia de un sistema automatizado genera varios inconvenientes: falta de precisión en el registro de existencias, desconocimiento de la cantidad exacta de productos disponibles, errores frecuentes en pedidos y reabastecimiento, dificultad para identificar pérdidas o robos, y pedidos tardíos o excesivos que comprometen la liquidez del negocio.

El propósito principal de esta investigación es implementar una herramienta tecnológica para la gestión de inventario en el almacén DK'PO que permita superar estas limitaciones, automatizando los procesos de registro, control y análisis de existencias. Este propósito responde directamente a la necesidad identificada de mejorar la eficiencia operativa y reducir las pérdidas económicas asociadas a una gestión manual deficiente.

La relevancia de este análisis se basa en su posibilidad para transformar significativamente la operatividad del almacén, proporcionando una solución adaptada a sus necesidades específicas que permita optimizar la disponibilidad de productos, reducir pérdidas por desabastecimiento o sobre almacenamiento y aumentar la rentabilidad a largo plazo. Además, representa una contribución práctica al campo de la gestión empresarial, demostrando cómo la aplicación de herramientas tecnológicas apropiadas puede beneficiar a pequeños negocios del sector comercial.

Para abordar esta problemática, se adopta un enfoque metodológico mixto que combina técnicas cualitativas y cuantitativas, permitiendo una comprensión integral del fenómeno estudiado. Desde la perspectiva cualitativa, se realizaron entrevistas al propietario del almacén para identificar necesidades específicas y expectativas respecto al sistema, mientras que el componente cuantitativo permitió establecer

indicadores medibles para evaluar objetivamente el funcionamiento de la solución implementada.

La construcción de la propuesta se fundamenta en la metodología ágil Scrum, estructurando el desarrollo en cuatro sprints: diseño de interfaz y registro de instrumentos, control de stock (entradas/salidas), reportes de inventario y alertas automáticas de stock bajo. Este enfoque iterativo e incremental facilitó la adaptación continua del sistema a los requerimientos identificados, asegurando que la solución final responda eficazmente a las necesidades específicas del almacén musical DK'PO.

I. EL PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

Según un estudio de González-Torres y Álvarez-Rodríguez (2020) que se realizó a microempresas comerciales de México para evaluar el impacto de la gestión de inventarios en la rentabilidad. Los resultados mostraron que la falta de un sistema de inventario eficiente y las prácticas deficientes están significativamente asociadas con menor rentabilidad. Específicamente, la ausencia de un sistema de inventario lleva a pérdida de ventas por falta de stock, aumento de costos de almacenamiento, dificultad para tomar decisiones informadas sobre compras, problemas en el control de inventario y menor capacidad para negociar con proveedores.

Según Paredes-Rodríguez y Pérez-Terán (2023) al realizar un estudio cuantitativo en empresas comercializadoras de la provincia de Pichincha, Ecuador, con el objetivo de analizar la relación entre la gestión de inventarios y la rentabilidad. Los resultados indicaron que la falta de un sistema de inventario adecuado y la implementación de prácticas deficientes tienen un impacto negativo en la rentabilidad. En particular, se encontró que las empresas con sistemas de inventario deficientes enfrentan mayores costos de almacenamiento y obsolescencia, lo que reduce su eficiencia operativa. Además, la ausencia de información precisa sobre los niveles de inventario dificulta la optimización del stock y la planificación de la demanda, afectando la capacidad de respuesta ante fluctuaciones del mercado. En general, una gestión ineficiente de inventarios limita el crecimiento y la competitividad de las empresas en la provincia.

Por otro lado, en el cantón Cayambe en el almacén DK'PO el proceso de gestión del inventario se realiza de manera manual y a menudo se producen errores de registro, lo que puede afectar de manera negativa la experiencia del cliente. Los procesos de gestión de inventario en el almacén se realizan actualmente en hojas físicas. Debido a la ausencia de un registro minucioso de ventas, no se lleva un control adecuado de los productos vendidos, lo que ocasiona la caída en las ventas por la ausencia de productos disponibles en el inventario.

Algunos de los problemas que se genera al no contar con alguna herramienta que ayude a la eficiencia y reducción de errores para la gestión de inventario en el almacén son:

Escasez de disponibilidad de productos populares.

- No se conoce la cantidad exacta de cada producto en existencia.
- Errores en pedidos y reabastecimiento debido a información inexacta.
- Dificultad para identificar robos o pérdidas de inventario.
- Pedidos tardíos o excesivos por falta de información precisa.
- Pérdidas económicas por precios incorrectos o descuentos mal aplicados.

1.2. FORMULACIÓN DEL PROBLEMA

El escaso uso de una herramienta tecnológica para la gestión de inventario en un almacén musical causa pérdida de dinero lo que genera ineficiencia en la gestión de inventario en el año 2024.

1.3. JUSTIFICACIÓN

El diseño e implementación de una herramienta web para la gestión de inventarios en un almacén de venta de instrumentos musicales ofrece múltiples beneficios que optimizan la eficiencia operativa y mejoran el control de stock. Diversos estudios y proyectos han demostrado la eficacia de estas soluciones tecnológicas en la administración de inventarios, permitiendo una mejor planificación, reducción de pérdidas y mayor disponibilidad de productos para los clientes.

Según Martínez-Pérez y Gómez-López (2021) el impacto de la digitalización en la gestión de inventarios en pequeñas y medianas empresas del sector musical, encontraron que la implementación de un sistema de inventario redujo en un 40% las pérdidas por falta de stock o sobre almacenamiento, mejoró en un 30% la velocidad de reposición de productos y optimizó la precisión en la gestión del inventario.

La efectividad en la gestión del inventario será un punto clave del sistema, ya que los módulos implementados para el control de stock, registro de entradas y salidas de productos, y generación de reportes permitirán al personal administrar el inventario de manera más eficiente.

Un buen diseño de una herramienta tecnológica de gestión de inventario permite a las empresas organizar todo el proceso, desde el ingreso de productos al almacén

hasta su venta o distribución. Esto contribuye a optimizar la disponibilidad de productos, reducir pérdidas por desabastecimiento o sobre almacenamiento y mejorar la eficiencia operativa.

En otro estudio Según Martínez y Silva (2023) analizaron el impacto del control de inventario en la rentabilidad de las tiendas de instrumentos musicales en América Latina. Su estudio reveló que aquellas tiendas que utilizaban sistemas manuales o basados en hojas de cálculo enfrentaban mayores costos de inventario, menores márgenes de beneficio y un mayor riesgo de obsolescencia. En contraste, las tiendas que implementaron sistemas de gestión de inventario automatizados redujeron costos, mejoraron su rentabilidad y aumentaron su habilidad de reaccionar ante variaciones en la demanda del mercado.

Además, Gupta y Kim (2022) afirman que “la satisfacción del cliente en aquellas empresas con sistemas de gestión eficientes ofreció una mayor disponibilidad de instrumentos, accesorios y partituras. Además, una gestión de inventario efectiva redujo los tiempos de espera y mejoro el asesoramiento al cliente, lo que contribuyó a un servicio de mayor calidad y a una ventaja competitiva en el sector”.

Estos estudios evidencian que la creación de una herramienta tecnológica para la gestión de inventarios es beneficiosa en los almacenes de venta de instrumentos musicales, ya que permite automatizar el control de stock, reducir errores en el registro de productos, mejorar la disponibilidad de inventario y optimizar la eficiencia operativa. La adopción de estas tecnologías se traduce en una ventaja competitiva en el sector comercial, permitiendo una mejor planificación y adaptación a las exigencias del mercado y las normativas vigentes.

Por lo tanto, el desarrollo de una herramienta tecnológica para la gestión de inventarios podría mejorar la eficiencia operativa, reducir pérdidas por desabastecimiento o sobre almacenamiento y aumentar la rentabilidad a largo plazo.

1.4. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN

1.4.1. Objetivo General

Desarrollar una herramienta tecnológica para la gestión de inventario en el almacén DK'PO.

1.4.2. Objetivos Específicos

1. Fundamentar teóricamente las herramientas tecnológicas y de gestión de inventario que sirven como sustento para la investigación
2. Diagnosticar como se está realizando el proceso de gestión de inventario en el almacén musical.
3. Diseñar una solución tecnológica que ayude en el proceso de gestión de inventario dentro del almacén musical.

1.4.3. Preguntas de Investigación

- ¿Cuáles son las herramientas tecnológicas y de gestión de inventario que permiten sustentar la investigación?
- ¿Como se están realizando los procesos de gestión de inventario en el almacén musical?
- ¿Con que solución tecnológica se mejorara el proceso de inventario dentro del almacén musical?

II. FUNDAMENTACIÓN TEÓRICA

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

En un estudio realizado por Ojo y Ayeni (2021) analizaron la influencia directa de los sistemas de gestión de inventario (SGI) en la eficiencia operativa y la rentabilidad dentro del contexto específico de tiendas minoristas de instrumentos musicales. A través de su investigación, identificaron que la implementación de un SGI permitió a estos establecimientos reducir de manera significativa los costos asociados al mantenimiento del inventario, mejorar sustancialmente la precisión y control de los niveles de stock, y optimizar la satisfacción del cliente gracias a la disponibilidad oportuna de los productos. Además, destacaron que la automatización de los procesos de inventario facilitó una toma de decisiones más informada y contribuyó a incrementar los márgenes de rentabilidad en un sector caracterizado por una amplia variedad de productos y fluctuaciones en la demanda

En otro estudio realizado por Kim y Lee (2022) elaboraron un estudio de caso centrado en la aplicación de una herramienta para la gestión de inventario (SGI) dentro de una tienda de instrumentos musicales, donde analizaron tanto las dificultades como las ventajas derivadas de este proceso. Los autores señalaron que uno de los principales retos enfrentados fue la dificultad del personal para adaptarse a nuevas condiciones, especialmente por empleados acostumbrados a los métodos manuales de gestión, lo cual generó la necesidad de realizar una capacitación extensa y continua para asegurar la correcta adopción del sistema. A pesar de estos desafíos iniciales, el estudio concluyó que la implementación del SGI permitió mejorar significativamente la visibilidad de los niveles de inventario, optimizar la planificación de la demanda según patrones de venta más precisos y reducir las pérdidas asociadas al robo y a la obsolescencia de productos, contribuyendo así a una gestión más eficiente de la tienda.

Un estudio llevado a cabo por Chen y Wang (2023) realizaron un análisis comparativo entre distintos sistemas de gestión de inventario (SGI) utilizados en tiendas de música, con el objetivo de determinar cuáles optimizan mejor el control del inventario y contribuyen a la eficiencia operativa. Los autores destacaron que los SGI basados en la nube, que incorporan análisis de datos en tiempo real, demostraron ser significativamente más efectivos que los sistemas tradicionales, al permitir una reducción notable de los valores asociados a la gestión de inventarios y una mayor capacidad de respuesta ante la demanda del cliente. Además, el estudio evidenció que estas plataformas modernas proporcionan mayor visibilidad y control sobre el inventario, lo que repercute positivamente en la toma de decisiones estratégicas y en la mejora de la experiencia del cliente dentro del entorno competitivo de las tiendas de música.

En un curso impartido por Cursos de Desarrollo Web (2023), se presentó la elaboración de una herramienta de gestión de inventario utilizando Laravel Filament V3, Livewire 3 y MySQL. Este curso se enfocó en la creación de un panel administrativo eficiente que permite gestionar categorías, productos, pedidos y líneas de pedidos. Los autores resaltaron la rapidez y flexibilidad de Laravel Filament para construir interfaces administrativas escalables mediante formularios, listados, menús y widgets personalizados. Además, se demostró la aplicabilidad práctica de estas tecnologías en entornos educativos y profesionales, ofreciendo ejemplos claros que facilitan su implementación. Este recurso evidencia cómo herramientas modernas pueden optimizar los procesos de gestión de inventario, apoyando tanto el aprendizaje como el desarrollo de soluciones efectivas.

2.2. MARCO TEÓRICO

2.2.1 Gestión de inventario

Bhatnagar y Teo (2019) definen la gestión de inventario como un conjunto de decisiones y procesos diseñados para controlar los niveles de stock con el fin de equilibrar los costos y mejorar el servicio al cliente. Este proceso requiere tener en cuenta factores como la demanda, los plazos de entrega y la capacidad de almacenamiento para optimizar la disponibilidad de productos y la eficiencia operativa.

2.2.1.1. Sistema de gestión de inventario

Chaffey y Ellis-Chadwick (2019) destacan que un sistema de gestión de inventario (SGI) es una herramienta esencial en el marketing digital, ya que permite administrar eficientemente el stock de productos disponibles para la venta en línea. Su implementación ayuda a las empresas a garantizar la disponibilidad de productos, evitar desabastecimientos y optimizar la fijación de precios, contribuyendo así a una mejor experiencia del cliente y a una gestión más eficiente del comercio electrónico.

2.2.1.2. Facturación automatizada

La facturación automatizada consiste en la utilización de sistemas tecnológicos que permiten generar, distribuir y almacenar facturas de forma automática. Estos sistemas suelen integrarse con plataformas contables, facilitando así el seguimiento en tiempo real de las transacciones financieras (Laudon & Laudon, 2017).

2.2.1.3. Herramienta tecnológica

Mokyr (2021) define una herramienta tecnológica como un dispositivo o proceso que permite a los seres humanos manipular y controlar su entorno para satisfacer sus necesidades. Desde esta perspectiva, la implementación de tecnologías en la gestión de inventarios facilita la optimización de los recursos y mejora la eficiencia operativa en diversos sectores.

2.2.1.4. Ingeniería de Software

Jalote (2023) define la ingeniería de software como un conjunto de prácticas y principios que permiten diseñar, desarrollar, probar y mantener software de alta calidad, asegurando que este cumpla con las necesidades de los usuarios. En el contexto de los sistemas de gestión de inventarios, la aplicación de estos principios contribuye a la creación de soluciones eficientes, escalables y adaptadas a las exigencias del mercado.

2.2.2. Metodologías ágiles

Ambler (2023) define las metodologías ágiles como un conjunto de enfoques iterativos e incrementales para el desarrollo de software, enfocados en la colaboración con el cliente, la adaptabilidad al cambio y la entrega frecuente de software funcional. En el contexto de un sistema de gestión de inventarios, la aplicación de metodologías ágiles permite desarrollar soluciones flexibles y

adaptativas, que mejora la eficiencia y la capacidad de respuesta a las necesidades del negocio.

2.2.2.1. Metodología XP

Wake (2021) presenta una visión actualizada de Extreme Programming (XP), destacando sus valores fundamentales de comunicación, simplicidad, retroalimentación, coraje y respeto, y su aplicación en el desarrollo de software moderno. En el contexto de una herramienta de gestión de inventarios, la adopción de XP puede mejorar la calidad del software, facilitar la adaptabilidad a los cambios en los requisitos y optimizar la colaboración entre desarrolladores y usuarios.

2.2.2.2. Metodología Scrum

Cohn (2019) describe Scrum como una metodología para la gestión de proyectos, donde el trabajo se organiza en sprints o iteraciones cortas. El equipo opera de forma autoorganizada, con un enfoque centrado en la presentación progresiva al cliente. Esta metodología puede ser especialmente útil en el desarrollo de sistemas de gestión de inventarios, ya que permite adaptarse rápidamente a los cambios y mejorar el producto de manera incremental.

2.2.2.3. Metodologías tradicionales

Pfleeger y Atlee (2020) definen las metodologías tradicionales como un proceso que va en secuencia para el desarrollo de software, donde cada fase se completa antes de pasar a la siguiente. Este enfoque, aunque estructurado, puede ser menos flexible frente a los cambios rápidos que requieren adaptaciones inmediatas en sistemas como los de gestión de inventarios.

Comparación entre metodología tradicional y ágil.

Misra et al. (2019) comparan las metodologías ágiles y tradicionales, destacando que las primeras se centran en la colaboración con el cliente y la entrega frecuente de software funcional, mientras que las metodologías tradicionales se enfocan en la planificación detallada y la documentación exhaustiva. Esta diferencia es clave en el desarrollo de sistemas de gestión de inventarios, donde las metodologías ágiles pueden ofrecer mayor flexibilidad y adaptabilidad frente a cambios rápidos en el mercado.

Tabla 1. Comparación entre metodologías tradicionales y ágiles

Tradicionales	Ágiles
El tamaño de los proyectos no importa	Se utiliza en proyectos pequeños
Equipos más avanzados	Duración corta
La duración es media larga	El equipo está conformado por el cliente
El cliente forma parte de las reuniones	La documentación no es muy detallada
Documentación precisa	Colaborativa
Los roles se mantienen fijos	Bajos costos en prototipado
Altos costos en prototipado	

Luego de evaluar la comparación entre metodologías ágiles y tradicionales, se observa una ventaja significativa al optar por enfoques ágiles. Estas metodologías destacan una interacción continua y una colaboración estrecha con el cliente durante todas las etapas del proyecto. Gracias a esto, el resultado final logra adaptarse mejor a las expectativas y requerimientos del cliente, a diferencia de los métodos tradicionales, donde la comunicación suele ser más estructurada y el desarrollo sigue un proceso lineal.

2.2.3. Metodología de desarrollo de software

Según García (2021), las metodologías de desarrollo de software son esenciales para garantizar la calidad y eficiencia en los procesos de creación de software. Estas metodologías fundamentaron un enfoque estructurado para el desarrollo de software, que incluye la planificación, el diseño, la codificación, la prueba y el mantenimiento de sistemas de software. En la actualidad, existen varias metodologías de desarrollo de software, entre las que se incluyen la metodología Scrum, la metodología Kanban y la metodología DevOps. Cada metodología tiene sus propias fortalezas y debilidades, y la selección de la metodología varía según diferentes aspectos, como el tamaño y la complejidad del proyecto, los requisitos de los interesados y los recursos disponibles. La metodología Scrum se centra en la colaboración, la flexibilidad y la entrega continua de software funcional en ciclos cortos. La metodología Kanban se centra en la optimización del flujo de trabajo y la reducción de residuos mediante el uso de tableros Kanban. La metodología DevOps se centra en la integración continua, la entrega continua y la automatización de la infraestructura de TI.

2.2.4. Leguajes de programación

Según López (2021), los lenguajes de programación son fundamentales para la creación de software y sistemas informáticos. Los lenguajes de programación son utilizados por los programadores para escribir código que pueda ser entendido tanto

por seres humanos como por máquinas. En la actualidad, pueden hallarse diferentes lenguajes de programación disponibles, cada uno con sus propias características y aplicaciones. Algunos de los lenguajes de programación más populares incluyen Java, Python, C++, JavaScript y PHP. 8 Java es un lenguaje de programación orientado a objetos que se utiliza principalmente para el desarrollo de aplicaciones empresariales y servidores web. Python es un lenguaje de programación de alto nivel que se utiliza para una amplia gama de aplicaciones, incluyendo ciencia de datos, inteligencia artificial y desarrollo web. C++ es un lenguaje de programación de bajo nivel que se utiliza para el desarrollo de aplicaciones de sistemas, como sistemas operativos y programas de control de dispositivos. JavaScript es un lenguaje de programación que se utiliza principalmente para el desarrollo de aplicaciones web. PHP es un lenguaje de programación que se utiliza principalmente para el desarrollo de aplicaciones web dinámicas.

2.2.5. Fases de desarrollo de software

Sommerville (2021) describe el desarrollo de software como un proceso estructurado que abarca varias fases fundamentales, incluyendo la especificación, el diseño, la implementación, las pruebas, el despliegue y el mantenimiento. Según el autor, seguir un enfoque bien definido en cada una de estas etapas permite garantizar la calidad del software y su alineación con los requisitos del usuario. Además, enfatiza la importancia de aplicar buenas prácticas de ingeniería de software para optimizar el desarrollo y la sostenibilidad del sistema a largo plazo.

2.2.6. Análisis de software

Hull, Jackson y Dick (2023) definen el análisis de software como un proceso fundamental dentro de la ingeniería de requisitos, el cual consiste en comprender y documentar las necesidades del usuario para un sistema de software. Este análisis es clave para garantizar que el sistema desarrollado cumpla con los objetivos y expectativas del cliente, minimizando posibles errores y optimizando su funcionalidad.

La correcta aplicación de este proceso contribuye significativamente al éxito del desarrollo de software.

2.2.7. Diseño de software

Gorton (2020) define el diseño de software como el proceso de crear una representación abstracta de un sistema de software, la cual sirve como guía para su

implementación. Este diseño permite establecer la estructura, los componentes y las interacciones del sistema, asegurando que cumpla con los requisitos funcionales y no funcionales. Una arquitectura de software bien definida es esencial para el desarrollo eficiente de sistemas robustos y escalables.

2.2.8. Servidor

Kurose y Ross (2021) definen un servidor como un programa que proporciona un servicio a otros programas denominados clientes. En el contexto de una herramienta de gestión de inventario para un almacén de venta de instrumentos musicales, un servidor desempeña un papel fundamental al almacenar, procesar y gestionar la información del inventario en tiempo real. La implementación de servidores permite una mejor administración de los datos, asegurando disponibilidad, seguridad y acceso eficiente a la información desde diferentes dispositivos.

2.2.9. Frameworks

Según Johnson (2018), un framework es un conjunto de herramientas, bibliotecas y convenciones que garantizan una estructura para el desarrollo de aplicaciones de software. Los frameworks se utilizan ampliamente en la creación de aplicaciones web, móviles y de escritorio para mejorar la eficiencia, la escalabilidad y la mantenibilidad del código.

2.2.10. Desarrollo del software

Martin (2020) enfatiza que escribir código limpio es fundamental para el desarrollo de software de calidad. Según el autor, el código debe ser legible, mantenible y eficiente, lo que implica aplicar principios sólidos de diseño, evitar la complejidad innecesaria y refactorizar continuamente para mejorar la estructura del software. La creación de soluciones efectivas en el desarrollo de software requiere un enfoque iterativo y colaborativo, en el cual los equipos aplican buenas prácticas como la simplicidad, el modularidad y la reutilización del código. Además, Martin destaca que un código bien estructurado reduce el riesgo de errores y facilita su evolución a largo plazo.

2.2.11. Pruebas de software

Dustin (2023) define las pruebas de software como un conjunto de actividades diseñadas para evaluar la calidad del software y garantizar que cumple con los requisitos del usuario. Estas pruebas incluyen la identificación de defectos, la

verificación de la funcionalidad esperada y la validación del rendimiento en distintos escenarios. Según el autor, un enfoque estructurado en las pruebas permite minimizar errores, potenciar la confiabilidad del software y optimizar la experiencia del usuario final. Además, destaca que las pruebas deben integrarse en todas las etapas del desarrollo para detectar problemas desde fases tempranas y reducir costos de corrección en el futuro.

2.2.12. JavaScript

Simpson (2020) explica que JavaScript es un lenguaje con múltiples características avanzadas que pueden resultar complejas de entender sin un estudio profundo. Su obra enfatiza la importancia de comprender los conceptos fundamentales y avanzados del lenguaje, como el ámbito léxico, el cierre de funciones (*closures*), la manipulación del prototipo y la asincronía. Según el autor, dominar estos aspectos permite a los desarrolladores escribir código más eficiente y optimizado, evitando errores comunes y mejorando la calidad del software.

2.2.13. Visual Studio Code

Dayley (2023) proporciona una guía completa sobre el uso de Visual Studio Code como entorno de desarrollo integrado, abordando tanto sus funcionalidades básicas como sus características avanzadas. El autor explica cómo configurar el editor, instalar extensiones, depurar código y optimizar el flujo de trabajo para mejorar la productividad en el desarrollo de software. Además, destaca la versatilidad de Visual Studio Code, ya que admite múltiples lenguajes de programación y permite personalizar el entorno según las necesidades del usuario.

2.2.14. PHP

Fernández y Castro (2021) señalan que PHP es esencial en la arquitectura del backend de las aplicaciones web, ya que se encarga de gestionar la lógica de negocio, establecer la conexión con las bases de datos y generar las respuestas adecuadas para el frontend. Esto ha consolidado a PHP como una herramienta robusta y versátil para el desarrollo de aplicaciones web dinámicas y eficientes.

2.2.15. Laravel

Laravel es un framework de desarrollo web basado en PHP, conocido por su sintaxis elegante y su enfoque en la simplicidad y productividad del desarrollador. Ofrece herramientas como Eloquent ORM, Blade Templating y un sistema de rutas intuitivo, lo

que lo convierte en una opción popular para aplicaciones web escalables (Stauffer, 2021)

2.2.16. Filament

Filament es un paquete de código abierto para Laravel que permite crear paneles de administración modernos y personalizables de manera rápida. Utiliza componentes de Livewire y Alpine.js para ofrecer interfaces dinámicas sin necesidad de escribir mucho código JavaScript (Filament, 2023)

2.2.16.1. Integración de Laravel y Filament para desarrollo ágil
La combinación de Laravel y Filament permite a los desarrolladores construir aplicaciones web con backend y paneles administrativos eficientes en poco tiempo. Filament se integra perfectamente con Laravel, aprovechando su arquitectura para gestionar datos y usuarios de forma segura y visualmente atractiva (Jones, 2022)

2.2.17. MySQL

Soto y Vargas (2024) señalan que MySQL es un sistema de gestión de bases de datos relacional (SGBDR) de código abierto, empleado ampliamente en entornos de desarrollo web y de escritorio para almacenar y gestionar datos estructurados de manera eficiente. Además, su compatibilidad con diversos lenguajes de programación y su capacidad para manejar grandes volúmenes de información lo posicionan como una herramienta clave en el desarrollo de aplicaciones modernas.

2.2.18. Calidad del software

Evans (2018) considera que un software tiene calidad cuando cumple adecuadamente con los requisitos especificados y satisface las necesidades del usuario. Según el autor, la calidad del software no solo se basa en la ausencia de defectos, sino también en su capacidad de adaptación a cambios, su rendimiento eficiente y su facilidad de mantenimiento.

III. METODOLOGÍA

3.1. ENFOQUE METODOLÓGICO

Para la siguiente investigación se utilizó un enfoque metodológico mixto el cual utiliza una estrategia que mezcla elementos tanto cuantitativos como cualitativos para un estudio logrando así una comprensión profunda de la investigación

3.1.1. Enfoque Cualitativo

Denzin y Lincoln (2018) definen el enfoque cualitativo como un método de investigación centrado en la comprensión profunda de los fenómenos sociales. Este enfoque se basa en la recolección y análisis de datos no numéricos, como entrevistas, observaciones y documentos, permitiendo explorar significados, percepciones y experiencias desde la perspectiva de los participantes.

Para esta investigación, el enfoque cualitativo aportó para conocer la realidad en la que se encuentra el proceso de gestión e inventario del almacén de venta de instrumentos musicales, considerando tanto el punto de vista de los trabajadores como del propietario.

3.1.2. Enfoque cuantitativo

Creswell y Creswell (2018) definen el enfoque cuantitativo como un método de investigación basado en la recopilación y análisis de datos numéricos para medir variables y probar hipótesis. Este enfoque se caracteriza por su énfasis en la objetividad, la generalización y el establecimiento de relaciones causales, lo que lo hace ideal para estudios donde se busca precisión y replicabilidad.

Teniendo en cuenta esta definición, en el contexto de una herramienta de gestión de inventario, se aplicó también el enfoque cuantitativo, el cual proporcionó precisión y confiabilidad en la recolección y análisis de los datos numéricos relacionados con el control de stock, movimientos de productos y niveles de inventario.

3.1.2. Tipo de Investigación

3.1.2.1. Investigación descriptiva

Bhattacharjee (2012) plantea que la investigación en ciencias sociales tiene como objetivo analizar fenómenos sociales a través de métodos científicos, combinando enfoques cualitativos y cuantitativos. Este proceso permite a los investigadores recopilar y analizar datos de manera estructurada para generar conocimiento válido y confiable. Su obra se centra en la aplicación de metodologías rigurosas que aseguren la objetividad en la recolección e interpretación de la información, facilitando así la toma de decisiones basadas en evidencia.

Se utilizó la investigación descriptiva, ya que permitió identificar las necesidades y requisitos del usuario de la herramienta tecnológica para la gestión de inventario. Además, facilitó una comprensión precisa y detallada de los procesos y características asociadas a la gestión de inventario, así como la identificación y descripción de los elementos clave de la herramienta tecnológica y los pasos involucrados en dicho proceso.

3.1.2.2. Investigación explicativa

Babbie (2022) explica que la investigación explicativa tiene como objetivo principal descubrir relaciones causales entre variables, permitiendo comprender las razones detrás de los fenómenos sociales. Este tipo de investigación es fundamental en estudios donde se busca determinar el impacto de ciertos factores en un resultado específico, utilizando métodos cuantitativos y cualitativos para validar sus hallazgos.

La investigación explicativa permitió identificar y analizar las causas de los problemas o desafíos en el proceso de gestión de inventario, lo que fue fundamental para desarrollar soluciones efectivas. Además, contribuyó a detectar áreas de mejora y optimización, facilitando la toma de decisiones informadas sobre la implementación de características, la selección de tecnologías y el diseño de la arquitectura de la herramienta tecnológica.

3.1.2.3. Investigación documental

Bowen (2009) define la investigación documental como un método de investigación cualitativa que se basa en el estudio sistemático de documentos existentes, los cuales pueden incluir informes institucionales, registros oficiales, artículos académicos,

manuales, políticas, entre otros. Este método permite extraer información relevante para comprender un fenómeno, identificar patrones, contrastar hallazgos previos y complementar otros métodos de recolección de datos. Además, Bowen enfatiza que el análisis documental es una herramienta útil cuando el acceso a fuentes primarias es limitado o cuando se busca obtener una perspectiva histórica o estructural del tema en estudio.

La investigación documental fue fundamental para obtener una comprensión profunda del contexto específico de la gestión de inventarios en un almacén de instrumentos musicales. Permitted explorar las mejores prácticas utilizadas en otras industrias afines, mantenerse actualizado sobre los avances tecnológicos en gestión de inventarios y garantizar el cumplimiento de normativas y estándares relevantes en el sector musical.

Al analizar la documentación existente sobre herramientas de gestión de inventarios, se identificaron estrategias efectivas utilizadas en el pasado y se pudieron anticipar posibles desafíos. Además, proporcionó referencias sólidas para respaldar decisiones de diseño, permitiendo realizar un análisis comparativo con otras soluciones del mercado. Esto facilitó la identificación de características innovadoras y áreas de mejora. En definitiva, la investigación documental sirvió como una base de conocimiento clave para el desarrollo de una herramienta eficaz, optimizada y adaptada a las necesidades específicas del almacén de instrumentos musicales.

3.2. IDEA A DEFENDER

EL desarrollo de una herramienta tecnológica permitirá mejorar el proceso de gestión de inventario en el almacén musical DK'PO en el año 2024.

3.3. DEFINICIÓN Y OPERACIONALIZACIÓN DE LAS VARIABLES

3.3.1. Definición de las variables

3.3.1.1. Definición variable Independiente

Herramienta tecnológica

Una herramienta tecnológica es un dispositivo o aplicación diseñado para facilitar, mejorar o realizar tareas específicas mediante la aplicación de principios y 13 conocimientos tecnológicos. Estas herramientas suelen incorporar tecnología digital o electrónica para proporcionar funcionalidades avanzadas que simplifican procesos, aumentan la eficiencia o permiten la realización de actividades que de

otra manera serían más complejas o incluso imposibles. Las herramientas tecnológicas pueden incluir hardware, software, dispositivos móviles, aplicaciones en línea y otros recursos digitales que contribuyen al logro de objetivos específicos en diversos campos, como la producción, la comunicación, la educación, la salud, entre otros.

3.3.2. Definición variable Dependiente

Gestión de inventario

La gestión de inventario se refiere al conjunto de prácticas y procesos utilizados para supervisar, controlar y organizar los bienes y productos almacenados por una organización. El objetivo principal de la gestión de inventario es asegurar que una empresa tenga la cantidad adecuada de productos en el lugar correcto y en el momento adecuado, minimizando al mismo tiempo los costos asociados con el almacenamiento y la gestión de existencias.

Tabla 2. Operacionalización de la variable dependiente

Variable Dependiente	Definición	Dimensión	Indicador	Técnicas	Instrumentos
Gestión de inventario en un estudio musical	La gestión de inventario se refiere al conjunto de prácticas y procesos utilizados para supervisar, controlar y organizar los bienes y productos almacenados por una organización	Gestión	<ul style="list-style-type: none">• Número de equipos correctamente registrados• Porcentaje de equipos en uso• Tasa de inactividad de equipos• Tasa de errores en los registros	Entrevista	Cuestionario

Tabla 3. Operacionalización de la variable independiente

Variable Independiente	Definición	Dimensión	Indicador	Técnica	Instrumentos
Herramienta tecnológica	Una herramienta tecnológica es un dispositivo o aplicación diseñado para facilitar, mejorar o realizar tareas específicas mediante la aplicación de principios y 13 conocimientos tecnológicos. Estas herramientas suelen incorporar tecnología digital o electrónica para proporcionar funcionalidades avanzadas que simplifican procesos, aumentan la eficiencia o permiten la realización de actividades que de otra manera serían más complejas o incluso imposibles.	Funcionalidad	<ul style="list-style-type: none"> Cantidad de funciones implementadas 	Entrevista	Cuestionario
		Usabilidad	<ul style="list-style-type: none"> Cantidad de informes sobre el inventario Facilidad con que el usuario puede interactuar 	Entrevista	

3.4. MÉTODOS UTILIZADOS

3.4.1. Método Inductivo

El método inductivo es un proceso de razonamiento que parte de observaciones específicas y particulares para llegar a conclusiones generales o teorías. Se basa en la identificación de patrones, regularidades o tendencias en los datos observados para formular hipótesis y, eventualmente, construir un principio o ley general. (Martínez-Rodríguez, 2021).

Este método ayudó a analizar los datos y a identificar los problemas más recurrentes para luego llegar a conclusiones generales. En resumen, el método inductivo es un proceso riguroso y ordenado que permite llegar a conclusiones generales a partir de la observación empírica.

3.4.2. Técnicas

3.4.2.1. Entrevista

Según Sánchez-Planells (2022, p. 45), "La entrevista se define como una conversación estructurada con un propósito específico, que busca obtener información relevante sobre el entrevistado a través de preguntas dirigidas y la observación de su comportamiento verbal y no verbal."

Así mismo, Fernández-Villavicencio y Méndez-Ibáñez (2021, p. 112) definen la entrevista como "una técnica de recolección de datos que involucra una interacción cara a cara (o virtual) entre un entrevistador y un entrevistado, con el objetivo de explorar a profundidad las experiencias, opiniones y perspectivas del entrevistado sobre un tema específico."

IV. RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS

4.1.1. Resultados de la entrevista

Para comprender la situación actual del almacén de venta de instrumentos musicales en relación con la gestión de inventario, se realizó una entrevista con el dueño del negocio. El objetivo de esta entrevista fue recopilar información clave sobre el método de administración empleado, identificar dificultades en la gestión del stock y conocer sus necesidades y expectativas respecto a un sistema automatizado.

Tras la entrevista, se identificó que el propietario del almacén no cuenta con un sistema de inventario formal ni utiliza un software especializado para la gestión de su stock. Actualmente, el control se lleva a cabo de manera manual mediante anotaciones en un cuaderno y, en algunas ocasiones, con el apoyo de hojas de cálculo básicas. Sin embargo, este método ha generado diversas dificultades, como errores en el registro de entradas y salidas, falta de precisión en la cantidad de productos disponibles y problemas para prever la reposición de instrumentos.

Además, el dueño del almacén mencionó que en varias ocasiones ha experimentado pérdidas o extravíos de instrumentos y accesorios debido a la falta de un control riguroso del stock. Estas situaciones afectan directamente la operatividad del negocio y dificultan la toma de decisiones informadas sobre compras y ventas.

Respecto a las mejoras esperadas con un sistema automatizado, el propietario expresó su interés en contar con una herramienta que le permita llevar un control y administración del inventario con mayor eficiencia, minimizando errores y optimizando la administración del stock. Entre las funcionalidades que considera esenciales, destacó la posibilidad de registrar información detallada sobre cada instrumento, como nombre, categoría, marca, modelo, número de serie, cantidad en stock, precio de compra y venta, y fecha de ingreso.

Asimismo, resaltó la importancia de implementar alertas y notificaciones automáticas para avisar cuando el stock de un instrumento sea bajo, así como recibir recordatorios sobre pedidos pendientes o vencimientos de pagos a proveedores. En cuanto a la usabilidad del sistema, el dueño enfatizó que, si bien él es quien gestiona el inventario, en ocasiones sus empleados también colaboran con el registro de productos. Por ello, es fundamental que la herramienta tecnológica sea intuitiva y fácil de usar, pero sin sacrificar funcionalidades avanzadas que permitan obtener reportes detallados sobre las existencias y movimientos del inventario.

- **Resultado de la entrevista al dueño del almacén**

1. ¿Cómo administra actualmente el inventario de su almacén?

Actualmente, llevo el control del inventario de forma manual. Anoto los ingresos y salidas de instrumentos en un cuaderno, y en ocasiones utilizo una hoja de cálculo básica para tener un registro más organizado.

2. ¿Utiliza algún software, hoja de cálculo o lo gestiona manualmente?

No utilizo ningún software especializado. Principalmente, hago el seguimiento del inventario de manera manual y, en algunos casos, apoyo el registro con hojas de cálculo, pero sin una estructura automatizada.

3. ¿Qué dificultades ha enfrentado al gestionar su inventario con su método actual?

Una de las principales dificultades es la falta de precisión en los registros. A veces se cometen errores al anotar las entradas y salidas de instrumentos, lo que provoca discrepancias en el inventario real. También resulta complicado hacer un seguimiento del stock disponible y prever cuándo es necesario reabastecer ciertos productos.

4. ¿Ha tenido pérdidas o extravíos de equipos por falta de control?

Sí, en algunas ocasiones he notado que faltan instrumentos o accesorios sin un registro claro de su venta o traslado. Esto puede deberse a errores en la anotación o a la falta de un control más riguroso sobre el stock.

5. ¿Qué aspectos del inventario le gustaría mejorar con un sistema automatizado?

Me gustaría tener un control más preciso del stock disponible, recibir alertas cuando los niveles de ciertos instrumentos sean bajos y generar reportes detallados sobre ventas, ingresos y salidas, también facilitaría mucho la búsqueda de productos.

6. ¿Qué información le gustaría registrar en el sistema sobre cada equipo o instrumento?

Sería útil registrar el nombre del instrumento, su categoría, marca, modelo, número de serie, cantidad en stock, precio de compra y venta y fecha de ingreso

7. ¿Necesita que el sistema tenga alertas o notificaciones? Por ejemplo, por el bajo stock

Sí, sería fundamental contar con alertas que avisen cuando el stock de un instrumento esté por debajo de un nivel mínimo. También sería útil recibir notificaciones sobre pedidos pendientes o vencimientos de pagos a proveedores.

8. ¿Quién se encarga de gestionar el inventario en el estudio? ¿Solo usted o hay otras personas involucradas?

Principalmente, yo me encargo de la gestión del inventario, aunque en ocasiones mis empleados también ayudan a registrar nuevas entradas o ventas. Un sistema *automatizado facilitaría mucho esta tarea y reduciría errores.

9. ¿Ha utilizado algún sistema de inventario o software similar antes?

No, nunca he utilizado un sistema de inventario automatizado. Hasta ahora, todo se ha gestionado de manera manual.

10. ¿Prefiere un sistema con una interfaz sencilla y fácil de usar o con más funcionalidades avanzadas?

Prefiero un sistema con una interfaz sencilla y fácil de usar, pero que cuente con las funciones necesarias para mejorar la gestión del inventario. No necesito algo demasiado complejo, pero sí que sea eficiente y práctico.

.

4.2. PROPUESTA

La propuesta se desarrolló a partir de una entrevista realizada al propietario del negocio, quien expresó la necesidad de implementar un sistema de gestión de inventario. Como resultado, se definieron los módulos clave que deberá incluir el sistema: gestión de instrumentos, control de stock, reportes y alertas y generación de inventario.

Como un valor agregado a la investigación y considerando las necesidades actuales del mercado, se decidió incorporar un módulo adicional de facturación electrónica. Esta funcionalidad permitirá al almacén DK'PO generar las ventas registradas en el sistema de inventario con la emisión automática de facturas, lo que optimizará los procesos administrativos del almacén.

Para garantizar un desarrollo eficiente y adaptable a las necesidades del negocio, se utilizará la metodología Scrum, lo que permitirá una implementación ágil y una mejora continua del sistema a través de iteraciones y retroalimentación constante del propietario.

4.2.1 Estudio de factibilidad

4.2.1.1 Factibilidad organizacional

Aspectos generales de la organización.

- **Institución:** Almacén DK'PO ESTUDIO.
- **Ubicación geográfica:** Cayambe
- **Área:** Almacén
- **Objeto social:** Servicio público
- **Misión:**

Proveer a músicos de todos los niveles con una amplia variedad de instrumentos musicales y accesorios de alta calidad, brindando un servicio personalizado, asesoramiento experto y precios competitivos. Nuestro objetivo es contribuir al desarrollo musical de nuestros clientes, satisfaciendo sus necesidades de manera eficiente y fomentando la pasión por la música en nuestra comunidad.

- **Visión:**

Ser el almacén líder en ventas de instrumentos musicales, reconocido por su excelencia en el servicio al cliente, la calidad de sus productos y su capacidad para

actualizarse a las tendencias del mercado musical. Queremos ser la primera opción de músicos y entusiastas de la música en nuestra región, promoviendo la cultura musical y fomentando un entorno de aprendizaje y creatividad.

4.2.1.1. Factibilidad técnica

Para la elaboración del sistema se identificaron y seleccionaron los recursos necesarios, incluyendo tanto software como hardware. La aplicación web será implementada utilizando tecnologías como Laravel, Filament, PHP, JavaScript y MySQL, las cuales fueron escogidas debido a su flexibilidad y capacidad de adaptación, lo que las convierte en una opción ideal para llevar a cabo este proyecto.

Tecnologías Software Utilizadas

Se emplearán tecnologías de desarrollo accesibles y ampliamente utilizadas, tales como:

- **Lenguaje de programación:** PHP
- **Framework Backend y Frontend:** Laravel (PHP)
- **Panel de Administración / Interfaz:** Filament (Laravel-based Admin Panel)
- **Sistema gestor de Base de Datos:** MySQL
- **Entorno de desarrollo integrado:** Visual Studio Code

El almacén de venta de instrumentos musicales cuenta con la infraestructura tecnológica necesaria para implementar el proyecto de un sistema de gestión de inventario. Además, dispone de conexión a internet que ayudara a la comunicación y el control de las operaciones internas. Aunque el negocio no cuenta con un servidor propio, la tecnología disponible es suficiente para garantizar que los recursos sean adecuados, lo que permite concluir que existe una factibilidad técnica para la realización del sistema propuesto.

Tecnologías Hardware

Equipo de computación: Computadora de escritorio Core i7

Impresora: Epson

4.2.1.3. Factibilidad Económica

El presupuesto del proyecto incluye la estimación de los recursos necesarios, abarcando hardware, software, personal y materiales de oficina.

Tabla 4. Factibilidad económica

Descripción	Cantidad	Costo Real	Costo referencial
Costos de Hardware			
Equipos de computación	1	00,00	500,00
Impresora	1	00,00	250,00
Total, de hardware		00,00	750,00
Costos de software			
PHP		00,00	00,00
Visual Studio Code		00,00	00,00
Laravel		00,00	00,00
Filament		00,00	00,00
MySQL		00,00	00,00
XAMPP server		00,00	00,00
Total de Software		\$ 00,00	\$ 00,00
Talento humano			
Programadores	1	00,00	2500,00
Total de talento humano		\$ 00,00	\$2500,00
Materiales de oficina			
Internet		17	17
Total de materiales de oficina		\$ 17	\$ 17
Subtotal		\$228,00	\$3.467,00
10% de imprevistos		\$ 22,8	\$ 346,7
Total		\$250,8	\$ 3.613,7

4.2.1.4. Factibilidad Operativa

Situación Actual

En la actualidad, el almacén de venta de instrumentos musicales gestiona su inventario de manera manual y sin un sistema automatizado. Las actividades relacionadas al control de productos se realizan de la siguiente forma:

- **Registro de productos:** El registro de nuevos instrumentos al inventario se hace manualmente, anotando la información básica en cuadernos o en hojas de cálculo simples.
- **Control de stock:** Las entradas y salidas de instrumentos se registran de forma manual, lo que genera imprecisiones y dificultad para conocer el stock real de cada producto en tiempo real.
- **Reportes:** Los reportes de inventario y ventas no se generan de manera regular ni automática, lo que restringe la habilidad para analizar y tomar decisiones.

- **Alertas:** No existe un mecanismo que notifique cuando el stock de un instrumento está por debajo del nivel mínimo, por lo que la reposición de productos suele hacerse de manera reactiva y no planificada.
- **Trazabilidad:** No se cuenta con un historial detallado de los movimientos de los productos, lo que dificulta el seguimiento adecuado de entradas y salidas, así como la detección de posibles pérdidas o extravíos.

Situación Ideal

La situación ideal para el almacén sería la implementación de un sistema web de gestión de inventario, que automatice y optimice los procesos actuales. Este sistema integraría las siguientes funcionalidades:

- Agregar, editar y eliminar productos.
- Control de stock (entradas/salidas) de instrumentos musicales.
- Reportes automáticos de inventario y ventas.
- Alertas de stock bajo para productos con niveles mínimos.
- Historial de movimientos de productos para un mejor control y trazabilidad.
- Emisión automática de facturas, cumpliendo con las normativas.
- Integración directa entre ventas e inventario, donde cada venta genere automáticamente la factura correspondiente y actualice el stock en tiempo real.

Con la implementación de este sistema web, el almacén podría:

- Reducir errores en la gestión de inventario.
- Optimizar los tiempos de atención al cliente.
- Tener un control preciso y actualizado de su stock.
- Facilitar la toma de decisiones estratégicas basadas en datos.
- Mejorar la eficiencia operativa y la satisfacción de sus clientes.
- Cumplir automáticamente con las obligaciones sin procesos manuales adicionales.
- Mejorar la imagen profesional del negocio al entregar comprobantes electrónicos modernos y confiables.
- Facilitar la conciliación contable al tener todos los movimientos integrados en un solo sistema.

4.2.2. Metodología Scrum

4.2.2.1. Fase 1 Inicio (Preparación del Proyecto)

Esta fase es crucial para establecer las bases para la elaboración de la herramienta tecnológica de inventario. Aquí se definen los objetivos, se identifican los requerimientos y se organiza el trabajo para garantizar una implementación estructurada y eficiente.

Objetivo del Sistema

Apoyar la gestión y control del stock en el almacén de venta de instrumentos musicales, integrando un sistema de facturación electrónica, reduciendo pérdidas y optimizando los procesos administrativos.

Objetivos específicos:

- Automatizar el registro de entrada y salida de instrumentos.
- Mantener un control preciso del stock disponible.
- Generar reportes sobre movimientos y estado del inventario.
- Implementar alertas y notificaciones sobre bajo stock.
- Facilitar la búsqueda de productos con un sistema organizado.
- Integrar un módulo de facturación electrónica que genere comprobantes automáticamente al registrar ventas.
- Sincronizar las ventas con la actualización automática del inventario para mantener datos consistentes.

Requerimientos Iniciales

Los requerimientos iniciales se dividen en funcionales y no funcionales.

Requerimientos Funcionales:

Tabla 5. Requerimientos funcionales

ID	Nombre	Descripción
RF01		El sistema debe permitir agregar nuevos productos (instrumentos musicales) al inventario.
RF02		El sistema debe permitir editar la información de los productos existentes (nombre, categoría, marca, precio).
RF03		El sistema debe permitir eliminar productos del inventario.
RF04		El sistema debe registrar las entradas y salidas de stock de cada producto.
RF05		El sistema debe generar reportes automáticos de inventario (productos disponibles, productos más vendidos).
RF06		El sistema debe generar reportes de ventas por período (diario, semanal, mensual).
RF07		El sistema debe enviar alertas cuando el stock de un producto esté por debajo del nivel mínimo establecido.
RF08		El sistema debe mostrar un historial detallado de los movimientos de cada producto (entradas y salidas).
RF09		El sistema debe permitir la búsqueda y filtrado de productos por nombre, categoría o marca.
RF10		El sistema debe gestionar usuarios con diferentes roles (administrador y empleados) y restringir el acceso a ciertas funciones según el rol.
RF11		El sistema debe generar facturas electrónicas automáticamente al registrar una venta.
RF12		El sistema debe actualizar automáticamente el inventario al emitir una factura de venta.

Requerimientos No Funcionales:

Tabla 6. Requerimientos No funcionales

ID	Nombre	Descripción
RNF01	Usabilidad	El sistema debe tener una interfaz amigable e intuitiva para facilitar su uso por el personal del almacén.
RNF02	Seguridad	El sistema debe asegurar la protección de los datos mediante autenticación de usuarios y contraseñas.
RNF03	Disponibilidad	El sistema debe estar disponible al menos el 99% del tiempo
RNF04	Rendimiento	El sistema debe permitir tiempos de carga menores a 3 segundos por página.
RNF05	Escalabilidad	El sistema debe ser escalable para soportar un aumento en la cantidad de productos y usuarios sin afectar el rendimiento.
RNF06	Accesibilidad	El sistema debe ser accesible para todo el personal del almacén
RNF07	Eficiencia	La aplicación debe proporcionar tiempos de respuesta rápidos
RNF08	Integridad de datos	El sistema debe garantizar la integridad y trazabilidad de todos los comprobantes electrónicos emitidos.
RNF09	Respaldo	El sistema debe mantener respaldos seguros de todos los comprobantes electrónicos

Product Backlog

El Product Backlog es una lista priorizada que nos sirve para ver todas las funcionalidades y tareas necesarias para el sistema.

Product Backlog

Tabla 7. Funcionalidades y tareas del sistema

ID	Funcionalidad	Prioridad	Descripción
1	Registro de instrumentos	Alta	Permitir agregar y editar información de instrumentos
2	Control de stock	Alta	Registrar entradas y salidas de inventario
3	Reportes de inventario	Media	Generar informes sobre los productos en stock
4	Alertas de bajo stock	Alta	Notificar cuando un producto está por agotarse
5	Interfaz intuitiva	Media	Diseñar una interfaz fácil de usar
6	Facturación electrónica	Alta	Generar comprobantes electrónicos automáticamente al registrar ventas
7	Sincronización inventario-facturación	Alta	Actualizar automáticamente el stock al emitir facturas

4.2.2.2. Fase 2 Planificación y Estimación

El objetivo de esta fase es estructurar y organizar el desarrollo del sistema de inventario de manera eficiente y progresiva. A través de la planificación de Sprints, la priorización de tareas y la definición de criterios de aceptación, se busca garantizar un desarrollo ordenado, minimizar riesgos y asegurar que cada funcionalidad cumpla con los requisitos antes de su entrega.

Dividir el desarrollo en Sprints

Para garantizar un desarrollo estructurado y eficiente del sistema de gestión de inventario, se ha dividido el trabajo en Sprints, siguiendo la metodología ágil Scrum. Cada Sprint tiene una duración de tres semanas y se enfoca en desarrollar funcionalidades específicas del sistema, priorizadas según su impacto y dependencia con otros módulos.

Lista de funcionalidades seleccionadas para cada Sprint

Se ha definido un conjunto de funcionalidades esenciales para el sistema de inventario, organizadas en el Product Backlog. Las funcionalidades han sido distribuidas en los Sprints según su prioridad y dependencia, asegurando un flujo lógico en el desarrollo del sistema.

A continuación, se presentan las funcionalidades seleccionadas para cada Sprint:

- **Sprint 1:**
 - Diseño de la interfaz gráfica del sistema.
 - Implementación del módulo de registro de instrumentos (CRUD: creación, edición, eliminación y visualización).

- **Sprint 2:**
 - Desarrollo del módulo de control de stock (registro de entradas y salidas de inventario).
 - Implementación de validaciones para asegurar consistencia en los movimientos de stock.
- **Sprint 3:**
 - Generación de reportes de inventario en formatos descargables (PDF, Excel).
 - Implementación de filtros de búsqueda en los reportes.
- **Sprint 4:**
 - Configuración del sistema de alertas automáticas de bajo stock.
 - Implementación de notificaciones visuales cuando el stock de un producto sea crítico.
- **Sprint 5:**
 - Generación automática de comprobantes electrónicos al registrar ventas
 - Sincronización entre ventas, inventario y facturación

Tabla 8. Planificación de Sprints y sus respectivas funcionalidades

Sprint	Duración	Funcionalidades desarrolladas
Sprint 1	3 semanas	Diseño de interfaz + Registro de instrumentos
Sprint 2	3 semanas	Control de stock (entradas/salidas)
Sprint 3	2 semanas	Reportes de inventario
Sprint 4	2 semanas	Alertas automáticas de stock bajo
Sprint 5	3 semanas	Facturación electrónica

Priorización de tareas

Para establecer la prioridad de cada tarea en el desarrollo del sistema, se han considerado los siguientes criterios:

- **Impacto en el sistema:** Funcionalidades esenciales como el registro de instrumentos y el control de stock tienen mayor prioridad, ya que son la base del sistema.
- **Dependencias:** Algunas tareas requieren que otras sean completadas previamente (por ejemplo, el control de stock depende del registro de instrumentos).

Identificación de tareas críticas y su relación con otras funcionalidades

Las funcionalidades críticas son aquellas que impactan directamente la gestión del inventario y que son la base del sistema. Estas incluyen:

1. **Registro de instrumentos:** Permite gestionar los productos en el almacén. Es esencial antes de implementar el control de stock.
2. **Control de stock:** Necesita que los productos estén registrados para gestionar entradas y salidas de inventario.
3. **Alertas de bajo stock:** Depende del control de stock para monitorear niveles de inventario.
4. **Reportes de inventario:** Requiere datos de los registros de productos y del control de stock para generar informes útiles.
5. **Facturación electrónica;** Generación automática de comprobantes electrónicos al registrar ventas y la sincronización entre ventas, inventario y facturación.

Desglose de cada funcionalidad en tareas más pequeñas

Para una mejor organización, cada funcionalidad se ha desglosado en tareas específicas que deben completarse en cada Sprint.

Tabla 9. Tareas organizadas por prioridad

ID	Funcionalidad	Prioridad	Tareas
1	Registro de instrumentos	Alta	Crear formulario de registro, implementar validaciones, almacenar en base de datos
2	Control de stock	Alta	Diseñar interfaz, programar funciones de entradas/salidas, actualizar inventario en BD
3	Alertas de bajo stock	Alta	Configurar sistema de notificaciones, definir umbrales de stock
4	Reportes de inventario	Media	Generar reportes en PDF y Excel, implementar filtros de búsqueda
5	Interfaz intuitiva	Media	Diseñar y optimizar la UI/UX para facilitar la navegación del usuario
6	Facturación electrónica	Alta	Generar facturas electrónicas y sincronizar con ventas, inventario y facturación

Definir criterios de aceptación

Para que una funcionalidad sea considerada finalizada y lista para su entrega, debe cumplir con los siguientes criterios generales:

1. **Correctitud y funcionalidad:** La funcionalidad debe ejecutarse sin errores y cumplir con los requisitos especificados.

2. **Almacenamiento y procesamiento de datos:** La información ingresada debe registrarse correctamente en la base de datos.
3. **Interfaz de usuario:** La funcionalidad debe ser accesible y fácil de usar sin generar confusión.
4. **Pruebas y validación:** Debe haber pasado pruebas unitarias y de integración.
5. **Documentación:** Se debe registrar su funcionamiento y cualquier configuración necesaria.

Tabla 10. Criterios de aceptación para cada funcionalidad

Funcionalidad	Criterios de Aceptación
Registro de instrumentos	El usuario puede agregar y editar un instrumento sin errores. Los datos se almacenan correctamente en la base de datos.
Control de stock	Al registrar una entrada/salida, el stock se actualiza automáticamente en el sistema. Se muestran mensajes de confirmación.
Reportes de inventario	Los reportes muestran información precisa y pueden descargarse en PDF y Excel.
Alertas de bajo stock	Se envía una notificación automática cuando el stock de un producto baja del umbral establecido.
Interfaz intuitiva	Los elementos de la UI son comprensibles y funcionales, con navegación fluida.
Facturación electrónica	Una vez realizada una venta se debe generar automáticamente una factura electrónica y se debe actualizar el stock.

Verificación de requisitos antes de la entrega

Para garantizar que cada funcionalidad cumple con los criterios establecidos antes de ser entregada, se realizará:

- **Pruebas unitarias:** Para validar el correcto funcionamiento de cada módulo.
- **Pruebas de integración:** Para verificar la interacción entre diferentes funcionalidades.
- **Revisión del código:** Para optimizar la eficiencia y detectar errores.
- **Validación con usuarios clave:** Para asegurar que cumple con las necesidades del almacén.

Historias de Usuario por Sprint

Tabla 11. Sprint 1: Diseño de interfaz + Registro de instrumentos

ID	Rol	Funcionalidad	Beneficio
HU-001	Administrador del almacén	Contar con una interfaz de usuario intuitiva y atractiva	Gestionar el inventario de forma eficiente y sin confusiones
HU-002	Administrador del almacén	Poder agregar nuevos instrumentos al sistema	Mantener actualizado el catálogo de productos disponibles
HU-003	Administrador del almacén	Poder modificar la información de los instrumentos existentes	Mantener los datos actualizados y corregir errores

Tabla 12. Sprint 2: Control de stock (entradas/salidas)

ID	Rol	Funcionalidad	Beneficio
HU-004	Administrador del almacén	Registrar la entrada de nuevos instrumentos al inventario	Aumentar el stock disponible y mantener un control de las adquisiciones
HU-005	Administrador del almacén	Registrar la salida de instrumentos del inventario	Disminuir el stock y mantener un control de las ventas o transferencias
HU-006	Administrador del almacén	Asignar y modificar ubicaciones físicas de los instrumentos	Facilitar su localización y optimizar el espacio de almacenamiento

Tabla 13. Sprint 3: Reportes de inventario

ID	Rol	Funcionalidad	Beneficio
HU-007	Administrador del almacén	Generar reportes del estado actual del inventario	Para tener una visión clara de los instrumentos disponibles y su valorización
HU-008	Administrador del almacén	Generar reportes de los movimientos de entrada y salida	Para analizar el flujo de instrumentos en un período determinado

Tabla 14. Sprint 4: Alertas automáticas de stock bajo

ID	Rol	Funcionalidad	Beneficio
HU-009	Administrador del almacén	Establecer umbrales mínimos para cada instrumento	Para definir cuándo se considera que un producto tiene stock bajo
HU-010	Administrador del almacén	Recibir notificaciones cuando el stock baje del umbral establecido	Para poder tomar acciones oportunas de reposición
HU-011	Administrador del almacén	Que el sistema genere sugerencias de pedidos de reposición	Para agilizar el proceso de reabastecimiento de instrumentos con stock bajo

Tabla 15. Sprint 5: Facturación electrónica

ID	Rol	Funcionalidad	Beneficio
HU-012	Administrador del almacén	Generar facturas electrónicas automáticamente al registrar una venta	Para cumplir con obligaciones tributarias y agilizar el proceso de venta
HU-013	Administrador del almacén	Que el sistema actualice automáticamente el inventario al emitir una factura	Para mantener consistencia entre ventas e inventario

4.2.2.3 Fase 3 Implementación

Desarrollar las funcionalidades de cada sprint para que el sistema funcione correctamente

4.2.2.3.1. Desarrollo Sprint 1

Funcionalidades por desarrollar

Diseño de interfaz

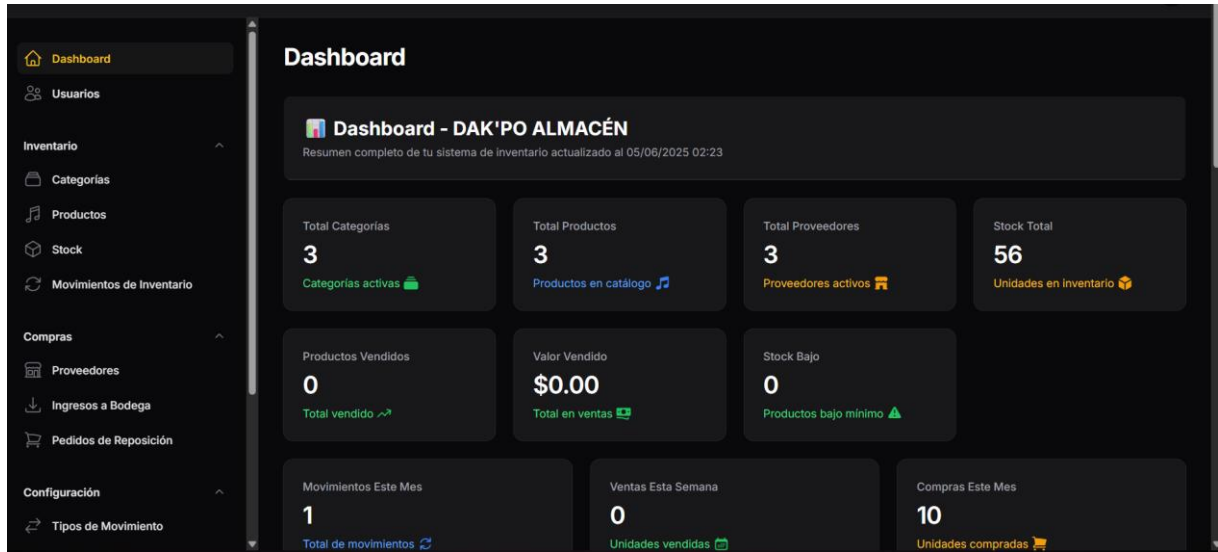


Figura 1. Diseño de interfaz

Crear los datos en la interfaz de productos para el registro de instrumentos.

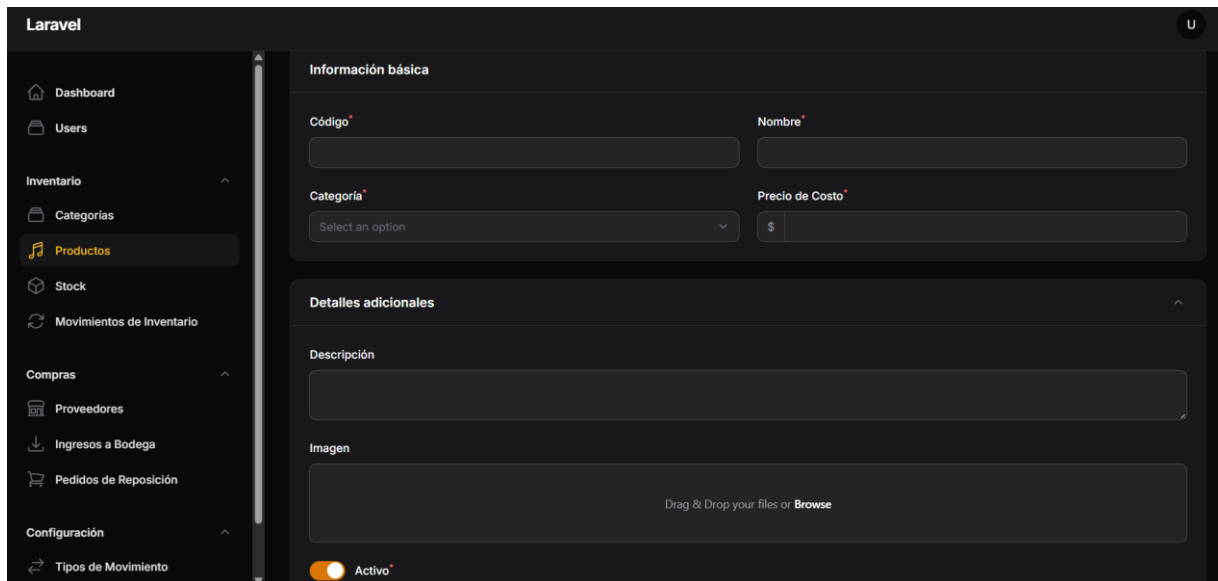


Figura 2. Interfaz registro de instrumentos

Implementar la funcionalidad para capturar los datos de un instrumento (nombre, tipo, marca, precio, etc.).

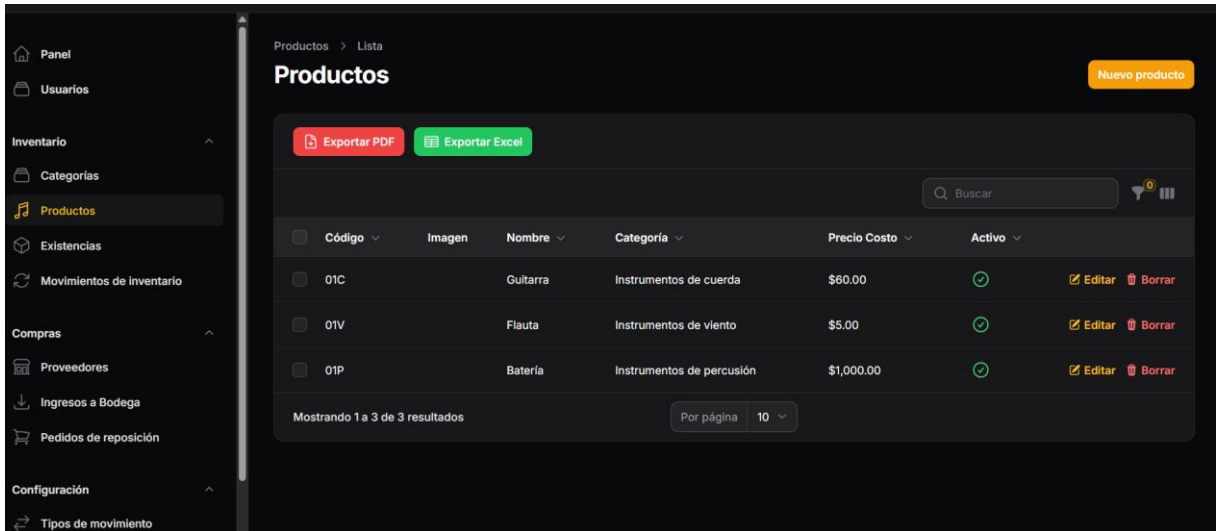


Figura 3. Instrumentos registrados

Conectar el formulario con la base de datos para guardar la información de los instrumentos.

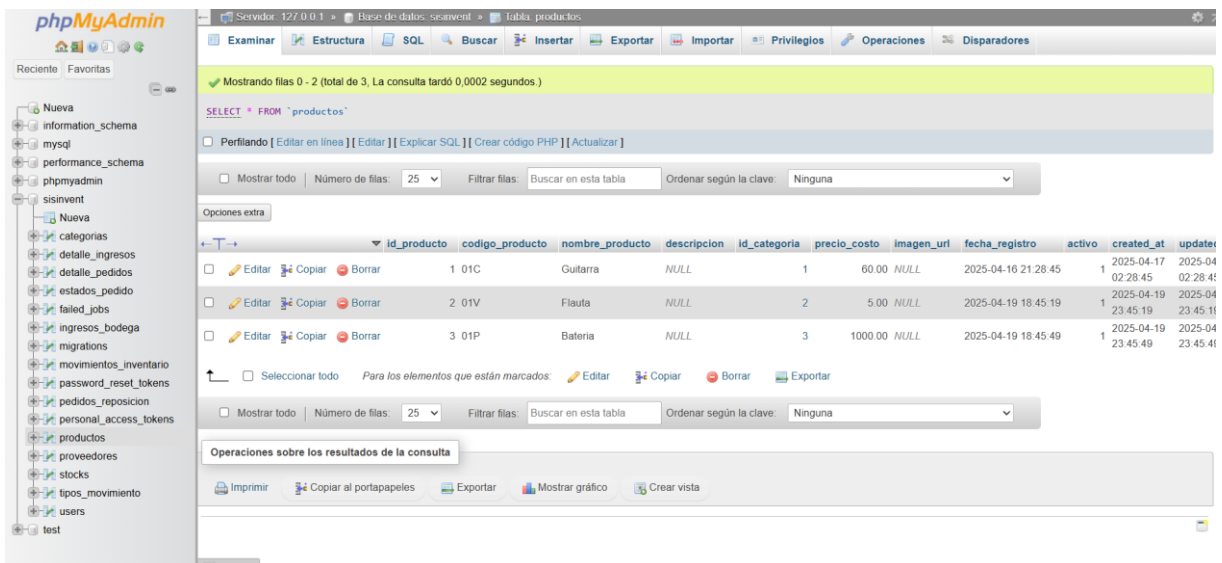


Figura 4. Registro de instrumentos en la base de datos

4.2.2.3.2. Desarrollo Sprint 2

Control de stock (entradas/salidas)

Funcionalidades por desarrollar

The screenshot shows a web application interface for inventory management. On the left is a sidebar menu with options like 'Panel', 'Usuarios', 'Inventario', 'Compras', and 'Configuración'. The main content area is titled 'Ver movimiento inventario' and contains a form with two sections: 'Información del Movimiento' and 'Detalles del Movimiento'. The 'Información del Movimiento' section includes fields for 'Producto' (Guitarra), 'Tipo de movimiento' (Ingreso por compra), 'Cantidad' (40), and 'Fecha del Movimiento' (19/04/2025 23:57:40). The 'Detalles del Movimiento' section includes fields for 'Origen de la ubicación', 'Ubicación Destino', 'Motivo', 'Responsable' (Bryan), and 'Notas'. On the right side of the form, there are buttons for 'Nuevo movimiento inventario', 'Vista', 'Editar', and 'Borra'.

Figura 5. Registro de entradas de instrumentos (cuando llegan al inventario).

The screenshot shows the same web application interface as Figure 5, but with different data entered in the form. The 'Información del Movimiento' section now shows 'Producto' as 'Flauta', 'Tipo de movimiento' as 'Venta', 'Cantidad' as '5', and 'Fecha del Movimiento' as '28/04/2025 21:01:12'. The 'Detalles del Movimiento' section shows 'Responsable' as 'Bryan'. The rest of the interface, including the sidebar and right-side buttons, remains the same.

Figura 6. Registro de salidas (cuando un instrumento se vende o se retira del inventario).

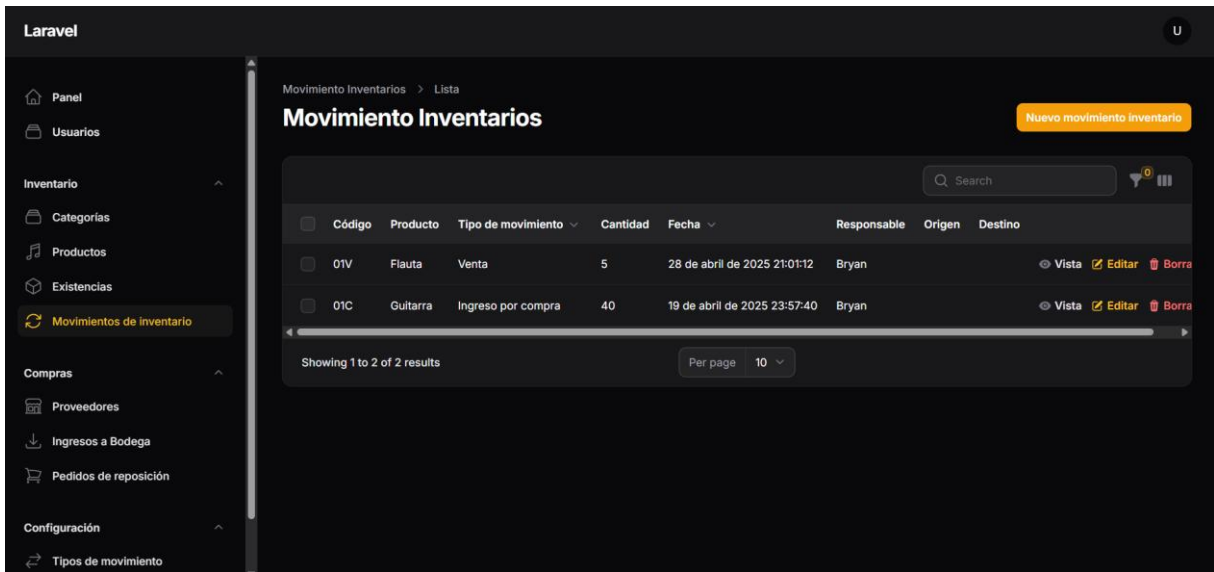


Figura 7. Implementación del modelo de datos para registrar movimientos de inventario.

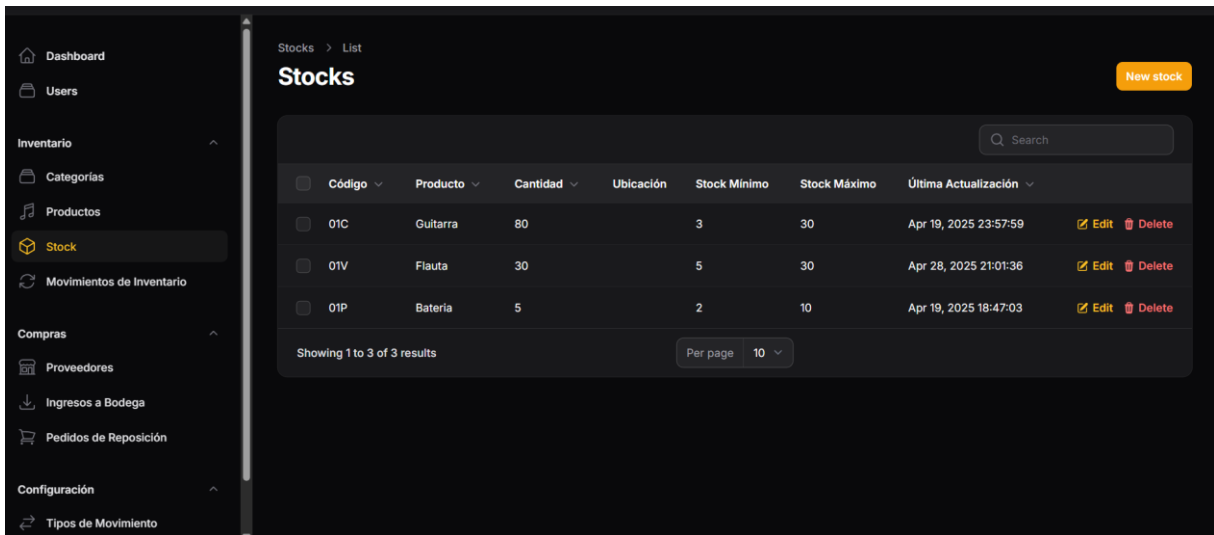


Figura 8. lógica para actualizar el stock total en tiempo real.

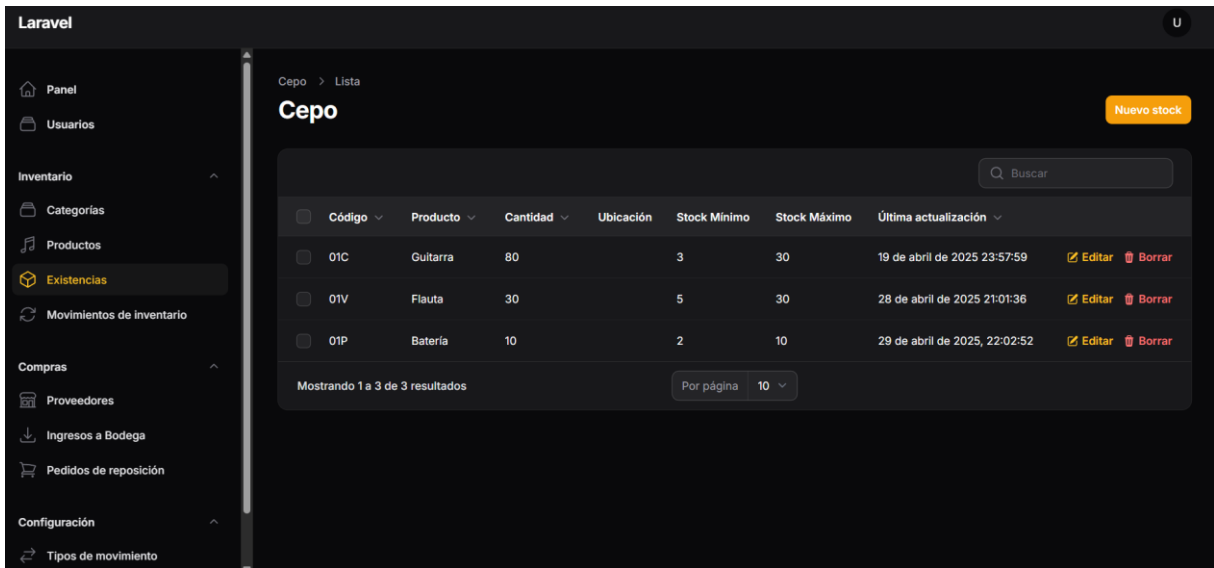


Figura 9. Lógica para actualizar el stock total en tiempo real.



Figura 10. Validación de movimientos (por ejemplo, no permitir salidas si no hay suficiente stock).

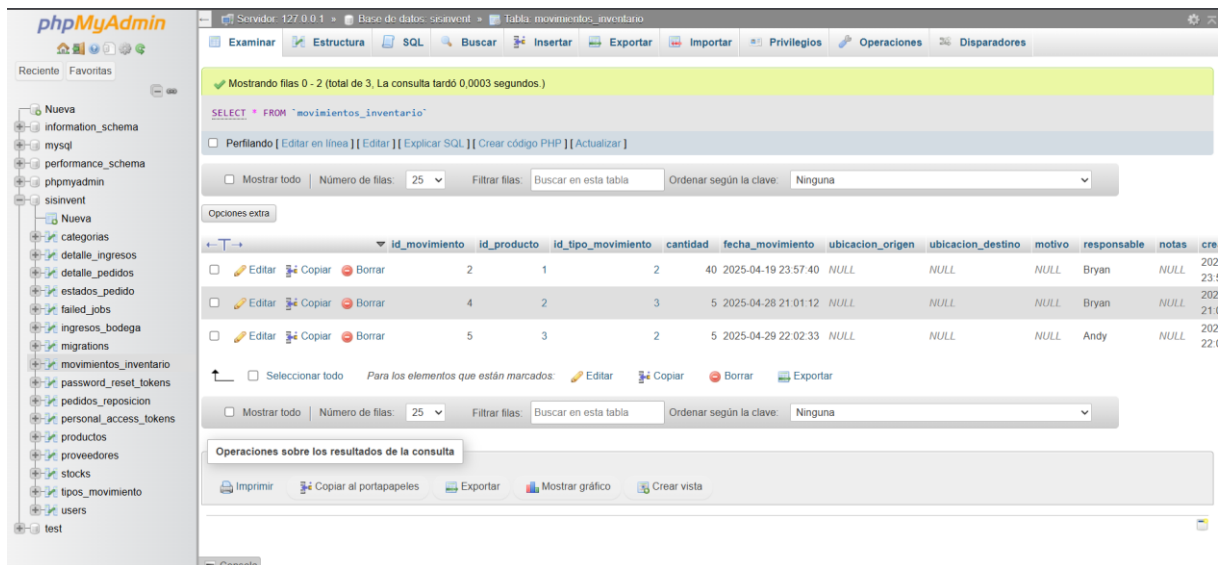


Figura 11. Conexión con la base de datos para guardar los registros de entradas/salidas.

4.2.2.3.3. Desarrollo Sprint 3

Reportes de inventario y ventas

Reportes de stock actual.

Reporte de Inventario

Fecha de generación: 30/04/2025 02:34

Código	Producto	Categoría	Stock Actual	Stock Mínimo	Stock Máximo	Ubicación	Precio Costo	Valor Total
01C	Guitarra	Instrumentos de cuerda	80	3	30		\$60.00	\$4,800.00
01V	Flauta	Instrumentos de viento	30	5	30		\$5.00	\$150.00
01P	Bateria	Instrumentos de percusión	10	2	10		\$1,000.00	\$10,000.00
Total Valor Inventario:								\$14,950.00

Figura 12. Reporte de inventario

Reporte de Movimientos

Todos los tiempos

Fecha de generación: 30/04/2025 02:35

Fecha	Código	Producto	Tipo	Cantidad	Responsable	Origen	Destino
29/04/2025 22:02	01P	Bateria	Ingreso por Compra	5	Andy	-	-
28/04/2025 21:01	01V	Flauta	Venta	5	Bryan	-	-
19/04/2025 23:57	01C	Guitarra	Ingreso por Compra	40	Bryan	-	-
Total:				50			

Figura 13. Reportes de entradas y salidas de inventario.

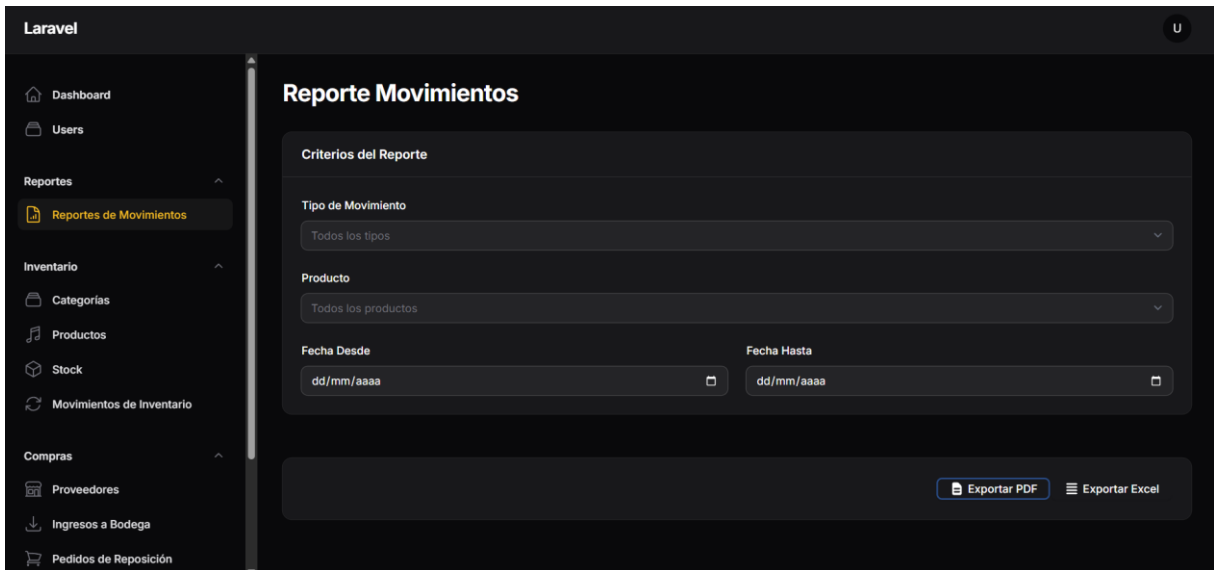


Figura 14. Diseño de la interfaz para visualizar y generar reportes.

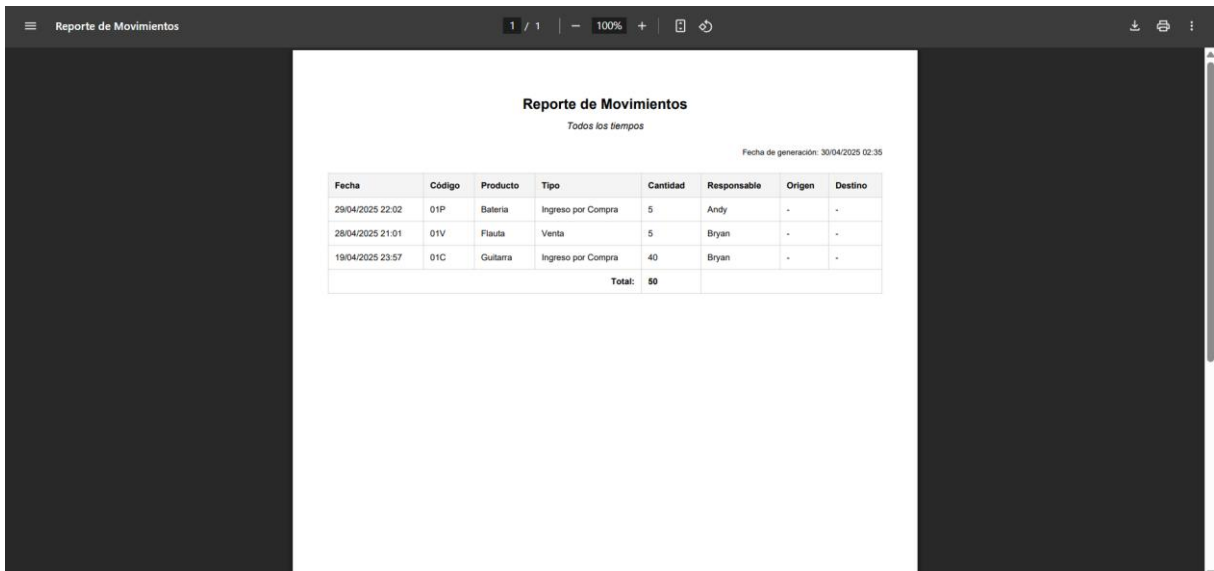


Figura 15. Generación de reportes visuales (tablas, PDF.).

Fecha	Código	Producto	Tipo Movimiento	Cantidad	Responsable	Origen	Destino
29/04/2025 22:02	01P	Bateria	Ingreso por Compra	5	Andy		
28/04/2025 21:01	01V	Flauta	Venta	5	Bryan		
19/04/2025 23:57	01C	Guitarra	Ingreso por Compra	40	Bryan		
Total:				50			

Figura 16. Generación de reportes visuales (tablas, PDF.).

4.2.2.3.4. Desarrollo Sprint 4

Alertas automáticas de stock bajo

Edit Stock Delete

Producto* Cantidad*

Ubicación en Bodega Stock Mínimo*

Stock Máximo*

Save changes Cancel

Figura 17. Posibilidad de configurar el umbral mínimo de stock por producto.

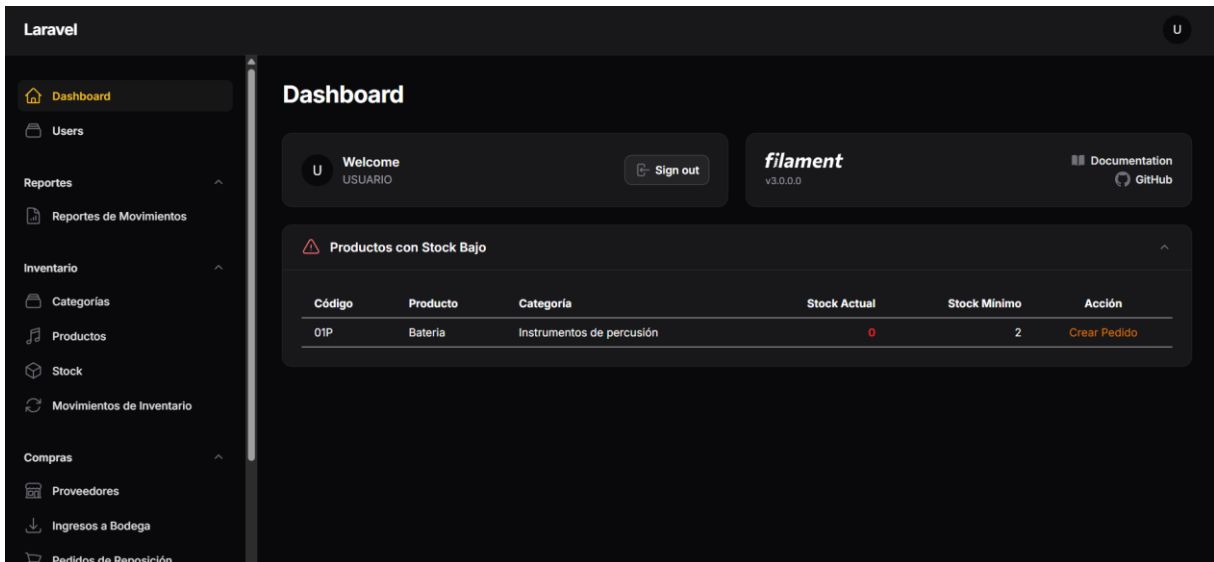


Figura 18. Visualización de alertas en el panel administrativo cuando el inventario de un instrumento alcance un nivel mínimo preestablecido.

4.2.2.3.5. Desarrollo Sprint 5

Funcionalidades por desarrollar

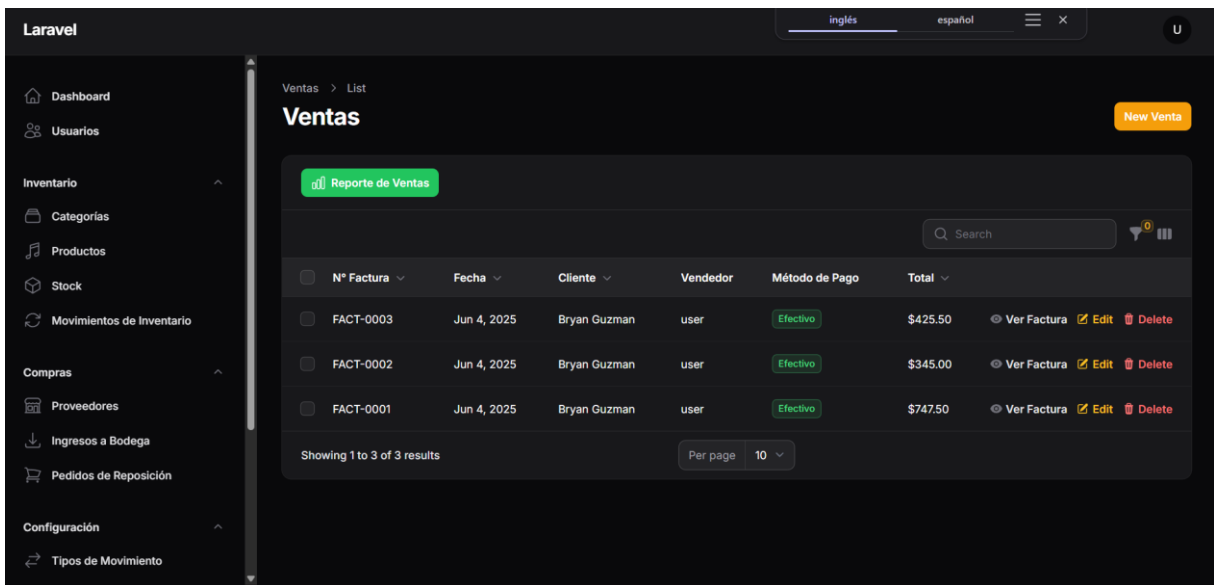


Figura 19. Generación automática de comprobantes electrónicos al registrar ventas

Codificación

Dashboard.php

```
app > Filament > Pages > Dashboard.php > Dashboard
1  <?php
2
3  namespace App\Filament\Pages;
4
5  use App\Filament\Widgets\InventarioStatsWidget;
6  use App\Filament\Widgets\CategoriasWidget;
7  use App\Filament\Widgets\VentasWidget;
8  use App\Filament\Widgets\EstadisticasAdicionalesWidget;
9  use Filament\Pages\Dashboard as BaseDashboard;
10
11 class Dashboard extends BaseDashboard
12 {
13     protected static ?string $navigationIcon = 'heroicon-o-home';
14
15     protected static string $view = 'filament.pages.dashboard';
16
17     public function getWidgets(): array
18     {
19         return [
20             InventarioStatsWidget::class,
21             EstadisticasAdicionalesWidget::class,
22             CategoriasWidget::class,
23             VentasWidget::class,
24         ];
25     }
26
27     public function getColumns(): int | array
28     {
29         return [
30             'sm' => 1,
31             'md' => 2,
32             'xl' => 4,
33         ];
34     }
35 }
```

Figura 22. Función para mostrar un resumen de todo lo que realiza el sistema de inventario

UserResource.php

```
22     public static function form(Form $form): Form
23     {
24         return $form
25             ->schema([
26             Forms\Components\TextInput::make('name')
27                 ->required()
28                 ->maxLength(255)
29                 ->label('Nombre'),
30
31             Forms\Components\TextInput::make('email')
32                 ->email()
33                 ->required()
34                 ->unique(ignoreRecord: true)
35                 ->maxLength(255)
36                 ->label('Correo Electrónico'),
37
38             Forms\Components\TextInput::make('password')
39                 ->password()
40                 ->required(fn (string $context): bool => $context === 'create')
41                 ->minLength(8)
42                 ->dehydrated(fn ($state) => filled($state))
43                 ->dehydrateStateUsing(fn ($state) => Hash::make($state))
44                 ->label('Contraseña'),
45
46             Forms\Components\TextInput::make('password_confirmation')
47                 ->password()
48                 ->same('password')
49                 ->required(fn (string $context): bool => $context === 'create')
50                 ->dehydrated(false)
51                 ->label('Confirmar Contraseña'),
52         ]);
53     }
```

Figura 23. Función para crear un usuario

CategoriaResource.php

```
public static function form(Form $form): Form
{
    return $form
        ->schema([
            Forms\Components\TextInput::make('nombre_categoria')
                ->required()
                ->maxLength(100)
                ->unique(ignoreRecord: true)
                ->label('Nombre de la Categoría'),

            Forms\Components\Textarea::make('descripcion')
                ->maxLength(65535)
                ->columnSpanFull()
                ->label('Descripción'),

            Forms\Components\Toggle::make('activo')
                ->required()
                ->default(true)
                ->label('Activo'),
        ]);
}
```

Figura 24. Función para crear una categoría

ProductoResource.php

```

public static function form(Form $form): Form
{
    return $form
        ->schema([
            Forms\Components\Section::make('Información básica')
                ->schema([
                    Forms\Components\TextInput::make('codigo_producto')
                        ->required()
                        ->maxLength(50)
                        ->unique(ignoreRecord: true)
                        ->label('Código'),

                    Forms\Components\TextInput::make('nombre_producto')
                        ->required()
                        ->maxLength(200)
                        ->label('Nombre'),

                    Forms\Components\Select::make('id_categoria')
                        ->required()
                        ->relationship('categoria', 'nombre_categoria')
                        ->preload()
                        ->searchable()
                        ->label('Categoría'),

                    Forms\Components\TextInput::make('precio_costo')
                        ->required()
                        ->numeric()
                        ->prefix('$')
                        ->label('Precio de Costo'),
                ])
            ->columns(2),
        ])
}

```

Figura 25. Función para crear un producto

```

Forms\Components\Section::make('Detalles adicionales')
    ->schema([
        Forms\Components\Textarea::make('descripcion')
            ->maxLength(65535)
            ->columnSpanFull()
            ->label('Descripción'),

        Forms\Components\FileUpload::make('imagen_url')
            ->directory('productos')
            ->image()
            ->imageEditor()
            ->label('Imagen'),

        Forms\Components\Toggle::make('activo')
            ->required()
            ->default(true)
            ->label('Activo'),
    ])
    ->collapsible(),
]);
}

```

Figura 26. Función para crear un producto

StockResource.php

```
public static function form(Form $form): Form
{
    return $form
        ->schema([
            Forms\Components\Select::make('id_producto')
                ->relationship('producto', 'nombre_producto')
                ->required()
                ->searchable()
                ->preload()
                ->label('Producto'),

            Forms\Components\TextInput::make('cantidad')
                ->required()
                ->numeric()
                ->minValue(0)
                ->default(0)
                ->label('Cantidad'),

            Forms\Components\TextInput::make('ubicacion_bodega')
                ->maxLength(100)
                ->label('Ubicación en Bodega'),

            Forms\Components\TextInput::make('stock_minimo')
                ->required()
                ->numeric()
                ->minValue(0)
                ->default(0)
                ->label('Stock Mínimo'),

            Forms\Components\TextInput::make('stock_maximo')
                ->required()
                ->numeric()
                ->minValue(0)
                ->default(0)
                ->label('Stock Máximo'),

        ]);
}
```

Figura 27. Función para crear un stock de cualquier instrumento que este registrado en productos

MovimientoInventarioResource.php

```
public static function form(Form $form): Form
{
    return $form
        ->schema([
            Forms\Components\Section::make('Información del Movimiento')
                ->schema([
                    Forms\Components\Select::make('id_producto')
                        ->relationship('producto', 'nombre_producto')
                        ->required()
                        ->searchable()
                        ->preload()
                        ->label('Producto')
                        ->reactive()
                        ->afterStateUpdated(function ($state, callable $set) {
                            if ($state) {
                                $stock = \App\Models\Stock::where('id_producto', $state)->first();
                                $set('stock_actual', $stock ? $stock->cantidad : 0);
                            }
                        }),
                    Forms\Components\Select::make('id_tipo_movimiento')
                        ->relationship('tipoMovimiento', 'nombre_tipo')
                        ->required()
                        ->label('Tipo de Movimiento')
                        ->reactive(),
                    Forms\Components\TextInput::make('stock_actual')
                        ->label('Stock Actual')
                        ->disabled()
                        ->dehydrated(false)
                        ->visible(function (callable $get) {
                            $tipoMovimientoId = $get('id_tipo_movimiento');
                            if (!$tipoMovimientoId) return false;
                        })
                ])
        ]);
}
```

Figura 28. Función para crear un nuevo movimiento de inventario

```
        $tipoMovimiento = \App\Models\TipoMovimiento::find($tipoMovimientoId);
        return $tipoMovimiento && strtolower($tipoMovimiento->nombre_tipo) === 'venta';
    }
});

Forms\Components\TextInput::make('cantidad')
    ->required()
    ->numeric()
    ->minValue(1)
    ->label('Cantidad')
    ->reactive()
    ->afterStateUpdated(function ($state, callable $get, callable $set) {
        $productoId = $get('id_producto');
        $tipoMovimientoId = $get('id_tipo_movimiento');

        if (!$productoId || !$tipoMovimientoId) return;

        $tipoMovimiento = \App\Models\TipoMovimiento::find($tipoMovimientoId);
        if (!$tipoMovimiento || strtolower($tipoMovimiento->nombre_tipo) !== 'venta') return;

        $stock = \App\Models\Stock::where('id_producto', $productoId)->first();
        $stockActual = $stock ? $stock->cantidad : 0;

        if ($state > $stockActual) {
            // Muestra un mensaje de error
            $set('error_stock', "¡Atención! Stock insuficiente. Disponible: {$stockActual}");
        } else {
            $set('error_stock', null);
        }
    })
});
```

Figura 29. Función para crear un nuevo movimiento de inventario

```

Forms\Components\TextInput::make('error_stock')
    ->label('')
    ->disabled()
    ->dehydrated(false)
    ->extraAttributes(['class' => 'text-danger-500'])
    ->visible(fn (callable $get) => $get('error_stock')),

Forms\Components\DateTimePicker::make('fecha_movimiento')
    ->required()
    ->default(now())
    ->label('Fecha del Movimiento'),
])
->columns(2),

Forms\Components\Section::make('Detalles del Movimiento')
    ->schema([
        Forms\Components\TextInput::make('ubicacion_origen')
            ->maxLength(100)
            ->label('Ubicación Origen'),

        Forms\Components\TextInput::make('ubicacion_destino')
            ->maxLength(100)
            ->label('Ubicación Destino'),

        Forms\Components\TextInput::make('motivo')
            ->maxLength(200)
            ->label('Motivo'),

        Forms\Components\TextInput::make('responsable')
            ->required()
            ->maxLength(100)
            ->label('Responsable'),
    ])

```

Figura 30. Función para crear un nuevo movimiento de inventario

ProveedorResource.php

```
public static function form(Form $form): Form
{
    return $form
        ->schema([
            Forms\Components\Section::make('Información de la empresa')
                ->schema([
                    Forms\Components\TextInput::make('nombre_empresa')
                        ->required()
                        ->maxLength(200)
                        ->label('Nombre de la Empresa'),

                    Forms\Components\TextInput::make('telefono')
                        ->tel()
                        ->maxLength(50)
                        ->label('Teléfono'),

                    Forms\Components\TextInput::make('email')
                        ->email()
                        ->maxLength(100)
                        ->label('Email'),
                ])
                ->columns(2),

            Forms\Components\Section::make('Dirección')
                ->schema([
                    Forms\Components\Textarea::make('direccion')
                        ->maxLength(65535)
                        ->label('Dirección'),

                    Forms\Components\Grid::make()
                        ->schema([
                            Forms\Components\TextInput::make('ciudad')
                                ->maxLength(100)
                                ->label('Ciudad'),
                        ])
                ])
        ])
}
```

Figura 31. Función para crear un registro de un nuevo proveedor con todos los datos de la empresa

```

        Forms\Components\TextInput::make('pais')
            ->maxLength(100)
            ->label('País'),
    ],
    ->collapsible(),

Forms\Components\Section::make('Información de contacto')
    ->schema([
        Forms\Components\TextInput::make('contacto_nombre')
            ->maxLength(100)
            ->label('Nombre del Contacto'),

        Forms\Components\TextInput::make('contacto_cargo')
            ->maxLength(100)
            ->label('Cargo del Contacto'),
    ])
    ->columns(2)
    ->collapsible(),

Forms\Components\Section::make('Información adicional')
    ->schema([
        Forms\Components\Textarea::make('notas')
            ->maxLength(65535)
            ->label('Notas'),

        Forms\Components\Toggle::make('activo')
            ->required()
            ->default(true)
            ->label('Activo'),
    ])
    ->collapsible(),
    ]);
}

```

Figura 32. Función para crear un registro de un nuevo proveedor con todos los datos de la empresa

IngresoBodegaResource.php

```
public static function form(Form $form): Form
{
    return $form
        ->schema([
            Forms\Components\Section::make('Información del Ingreso')
                ->schema([
                    Forms\Components\Select::make('id_proveedor')
                        ->relationship('proveedor', 'nombre_empresa')
                        ->required()
                        ->searchable()
                        ->preload()
                        ->label('Proveedor'),

                    Forms\Components\TextInput::make('numero_referencia')
                        ->maxLength(100)
                        ->label('Número de Referencia'),

                    Forms\Components\DatePicker::make('fecha_ingreso')
                        ->required()
                        ->default(now())
                        ->label('Fecha de Ingreso'),

                    Forms\Components\TextInput::make('responsable_recepcion')
                        ->required()
                        ->maxLength(100)
                        ->label('Responsable de Recepción'),
                ])
                ->columns(2),

            Forms\Components\Section::make('Notas')
                ->schema([
                    Forms\Components\Textarea::make('notas')
                        ->maxLength(65535)
                        ->columnSpanFull()
                        ->label('Notas'),
                ])
        ])
}
```

Figura 33. Función para registrar un ingreso a bodega con todos los datos de la persona que recibió el pedido

PedidoReposicionResource.php

```
public static function form(Form $form): Form
{
    return $form
        ->schema([
            Forms\Components\Section::make('Información del Pedido')
                ->schema([
                    Forms\Components\DatePicker::make('fecha_solicitud')
                        ->required()
                        ->default(now())
                        ->label('Fecha de Solicitud'),

                    Forms\Components\Select::make('id_proveedor')
                        ->relationship('proveedor', 'nombre_empresa')
                        ->required()
                        ->searchable()
                        ->preload()
                        ->label('Proveedor'),

                    Forms\Components\Select::make('id_estado')
                        ->relationship('estado', 'nombre_estado')
                        ->required()
                        ->label('Estado'),

                    Forms\Components\DatePicker::make('fecha_estimada_entrega')
                        ->label('Fecha Estimada de Entrega'),

                    Forms\Components\TextInput::make('responsable_pedido')
                        ->required()
                        ->maxLength(100)
                        ->label('Responsable del Pedido'),
                ])
            ->columns(2),
        ];
```

Figura 34. Función para crear un pedido de reposición con todos los datos del proveedor cuando el stock de un instrumento este bajo

DetallesRelationManager.php

```
public function form(Form $form): Form
{
    return $form
        ->schema([
            Forms\Components\Select::make('id_producto')
                ->relationship('producto', 'nombre_producto')
                ->required()
                ->searchable()
                ->preload()
                ->label('Producto'),

            Forms\Components\TextInput::make('cantidad_solicitada')
                ->required()
                ->numeric()
                ->minValue(1)
                ->label('Cantidad Solicitada'),

            Forms\Components\TextInput::make('cantidad_recibida')
                ->numeric()
                ->minValue(0)
                ->default(0)
                ->dehydrated(fn ($state, $record) => $record !== null)
                ->label('Cantidad Recibida'),

            Forms\Components\TextInput::make('precio_unitario')
                ->numeric()
                ->prefix('$')
                ->minValue(0)
                ->step(0.01)
                ->label('Precio Unitario'),
        ])
    ];
}
```

Figura 35. Detalles del instrumento que desea reponer

TipoMovimientoResource.php

```
public static function form(Form $form): Form
{
    return $form
        ->schema([
            Forms\Components\TextInput::make('nombre_tipo')
                ->required()
                ->maxLength(50)
                ->unique(ignoreRecord: true)
                ->label('Nombre del Tipo'),

            Forms\Components\Textarea::make('descripcion')
                ->maxLength(65535)
                ->columnSpanFull()
                ->label('Descripción'),
        ]);
}
```

Figura 36. Función para crear un tipo de movimiento

EstadoPedidoResource.php

```
public static function form(Form $form): Form
{
    return $form
        ->schema([
            Forms\Components\TextInput::make('nombre_estado')
                ->required()
                ->maxLength(50)
                ->unique(ignoreRecord: true)
                ->label('Nombre del Estado'),

            Forms\Components\Textarea::make('descripcion')
                ->maxLength(65535)
                ->columnSpanFull()
                ->label('Descripción'),
        ]);
}
```

Figura 37. Función para crear el estado del pedido

VentaResource.php

```
public static function form(Form $Form): Form
{
    return $form
        ->schema([
            wizard::make([
                wizard\Step::make('Datos del Cliente')
                    ->schema([
                        Forms\Components\Select::make('id_cliente')
                            ->label('Cliente')
                            ->relationship('cliente', 'nombre')
                            ->searchable()
                            ->preload()
                            ->createOptionForm([
                                Forms\Components\TextInput::make('nombre')
                                    ->required()
                                    ->maxLength(200)
                                    ->label('Nombre Completo'),

                                Forms\Components\TextInput::make('identificacion')
                                    ->required()
                                    ->maxLength(20)
                                    ->label('Cédula/RUC'),

                                Forms\Components\TextInput::make('direccion')
                                    ->maxLength(255)
                                    ->label('Dirección'),

                                Forms\Components\TextInput::make('telefono')
                                    ->maxLength(20)
                                    ->tel()
                                    ->label('Teléfono'),

                                Forms\Components\TextInput::make('email')
                                    ->email()
                                    ->maxLength(100)
                                    ->label('Email'),
```

Figura 38. Función para crear y registrar los datos del cliente, detalle de la venta, detalles del producto, para ver si hay stock disponible y para calcular el subtotal, IVA y precio total

```
Forms\Components\DatePicker::make('fecha_venta')
    ->required()
    ->default(now())
    ->label('Fecha de Venta'),

Forms\Components\TextInput::make('responsable_venta')
    ->required()
    ->maxLength(100)
    ->label('Vendedor')
    ->default(fn () => auth()->user()->name),

Forms\Components\Select::make('metodo_pago')
    ->options([
        'efectivo' => 'Efectivo',
        'tarjeta' => 'Tarjeta de Crédito/Débito',
        'transferencia' => 'Transferencia Bancaria',
        'cheque' => 'Cheque',
    ])
    ->required()
    ->default('efectivo')
    ->label('Método de Pago'),
),
```

Figura 39. Función para crear y registrar los datos del cliente, detalle de la venta, detalles del producto, para ver si hay stock disponible y para calcular el subtotal, IVA y precio total

```

Wizard\Step::make('Productos')
->schema([
  Forms\Components\Repeater::make('detalles')
  ->relationship()
  ->schema([
    Forms\Components>Select::make('id_producto')
    ->label('Producto')
    ->relationship('producto', 'nombre_producto')
    ->searchable()
    ->preload()
    ->required()
    ->reactive()
    ->afterStateUpdated(function ($state, callable $set, callable $get) {
      if ($state) {
        $producto = Producto::find($state);
        $stock = Stock::where('id_producto', $state)->first();

        if ($producto) {
          $set('precio_unitario', $producto->precio_costo * 1.25); // 25% de margen
        }

        if ($stock) {
          $set('stock_disponible', $stock->cantidad);

          // Mostrar alerta si no hay stock
          if ($stock->cantidad <= 0) {
            $set('error_stock', "¡Atención! No hay stock disponible para este producto.");

            Notification::make()
              ->title('Sin Stock')
              ->body("No hay stock disponible para el producto seleccionado.")
              ->danger()
              ->send();
          }
        }
      }
    });
  ]);
] );

```

Figura 40. Función para crear y registrar los datos del cliente, detalle de la venta, detalles del producto, para ver si hay stock disponible y para calcular el subtotal, IVA y precio total

```

Forms\Components\TextInput::make('stock_disponible')
->label('Stock Disponible')
->disabled()
->numeric()
->dehydrated(false)
->helperText('Cantidad disponible en inventario'),

Forms\Components\TextInput::make('cantidad')
->required()
->numeric()
->minValue(1)
->default(1)
->label('Cantidad')
->reactive()
->afterStateUpdated(function ($state, callable $set, callable $get) {
  $stockDisponible = floatval($get('stock_disponible') ?? 0);
  $state = floatval($state ?? 1);
  $idProducto = $get('id_producto');
  $producto = $idProducto ? Producto::find($idProducto) : null;
  $nombreProducto = $producto ? $producto->nombre_producto : 'Producto seleccionado';

  if ($state > $stockDisponible) {
    $set('error_stock', "¡Atención! Stock insuficiente. Disponible: {$stockDisponible}");

    Notification::make()
      ->title('Stock Insuficiente')
      ->body("No hay suficiente stock para {$nombreProducto}. Disponible: {$stockDisponible}")
      ->danger()
      ->persistent()
      ->send();
  } else {
    $set('error_stock', null);
  }
});

```

Figura 41. Función para crear y registrar los datos del cliente, detalle de la venta, detalles del producto, para ver si hay stock disponible y para calcular el subtotal, IVA y precio total

```

Wizard\Step::make('Finalizar')
->schema([
    Forms\Components\Placeholder::make('resumen_venta')
        ->label('Resumen de la Venta')
        ->content(function (callable $get) {
            $detalles = $get('detalles');
            if (!$detalles || empty($detalles)) {
                return 'No hay productos seleccionados.';
            }

            $subtotal = 0;
            $descuentoTotal = 0;

            foreach ($detalles as $detalle) {

                $cantidad = floatval($detalle['cantidad'] ?? 0);
                $precio = floatval($detalle['precio_unitario'] ?? 0);
                $descuento = floatval($detalle['descuento'] ?? 0);

                $subtotal += $cantidad * $precio;
                $descuentoTotal += $descuento;
            }

            $iva = ($subtotal - $descuentoTotal) * 0.15;
            $total = $subtotal - $descuentoTotal + $iva;

            return "Subtotal: $" . number_format($subtotal, 2) . "\n" .
                "Descuento: $" . number_format($descuentoTotal, 2) . "\n" .
                "IVA (15%): $" . number_format($iva, 2) . "\n" .
                "Total: $" . number_format($total, 2);
        }),
]);

```

Figura 42. Función para crear y registrar los datos del cliente, detalle de la venta, detalles del producto, para ver si hay stock disponible y para calcular el subtotal, IVA y precio total

4.2.2.4 Fase 4 Revisión y Pruebas

En esta fase se analizó y evaluó el trabajo realizado en el desarrollo de los sprints

Revisión de Sprints

Resultados Sprint 1

Como resultado del desarrollo del sprint 1, se logró implementar una interfaz intuitiva y visualmente atractiva que permite al administrador navegar fácilmente por todas las secciones del inventario. Además, se desarrolló con éxito la función de registro de instrumentos la cual permite ingresar información detallada de cada producto, como nombre, categoría, marca, cantidad y precio, esta función garantiza un control preciso del inventario y facilita la gestión de los instrumentos musicales en el almacén, lo que permite mantener actualizada la base de datos en tiempo real.

Tablas de Pruebas de Aceptación

Sprint 1: Diseño de interfaz + Registro de instrumentos

Tabla 16. Pruebas de Aceptación: HU-001 - Diseño de interfaz intuitiva

ID	Criterio	Contexto	Evento	Resultado Esperado	Cumplido
PA-001-01	Consistencia visual	Se accede a diferentes secciones del sistema	Cuando navego entre pantallas	Todos los elementos mantienen un diseño coherente (colores, tipografía, iconos)	[Sí]
PA-001-02	Navegación intuitiva	Dado que estoy en cualquier sección del sistema	Cuando necesito ir a otra sección	Se puede encontrar fácilmente los elementos de navegación y llegar a mi destino sin confusiones	[Sí]
PA-001-03	Tiempo de carga	Dado que uso el sistema	Cuando carga una página o sección	El tiempo de carga es menor a 3 segundos	[Sí]
PA-001-04	Mensajes de feedback	Se realizo cualquier acción en el sistema	Cuando la acción se completa o falla	Entonces recibo un mensaje claro que me informa sobre el resultado	[Sí]

Tabla 17. Pruebas de Aceptación: HU-002 - Registro de instrumentos musicales

ID	Criterio	Contexto	Evento	Resultado Esperado	Cumplido
PA-002-01	Creación de instrumento	Estoy en la sección de registro de instrumentos	Cuando estoy el formulario y presiono "Crear"	El instrumento se crea y aparece en el listado	[Sí]
PA-002-02	Validación de campos	Estoy registrando un instrumento	Cuando dejo campos obligatorios en blanco	El sistema me muestra un mensaje de error indicando los campos requeridos	[Sí]
PA-002-03	Persistencia en DB	Se ha registrado un instrumento	Cuando cierro sesión y vuelvo a ingresar	El instrumento sigue disponible en el sistema	[Sí]
PA-002-04	Carga de imágenes	Estoy registrando un instrumento	Cuando subo una imagen para el producto	Entonces la imagen se asocia correctamente y se visualiza en la ficha del producto	[Sí]
PA-002-05	Códigos únicos	Estoy registrando un instrumento	Cuando intento usar un código que ya existe	Entonces el sistema me alerta que el código debe ser único	[Sí]

Tabla 18. Pruebas de Aceptación: HU-003 - Edición de instrumentos

ID	Criterio	Contexto	Evento	Resultado Esperado	Cumplido
PA-003-01	Modificación de datos	Selecciono un instrumento ya existente	Cuando edito sus datos y presiono "Actualizar"	Los cambios se guardan correctamente	[Sí]
PA-003-02	Registro de cambios	He modificado un instrumento	Cuando reviso el historial de cambios	Entonces puedo ver los cambios realizados	[Sí]
PA-003-03	Actualización en tiempo real	Edito un instrumento	Cuando guardo los cambios	Los cambios se reflejan inmediatamente en todas las secciones del sistema	[Sí]

Resultados Sprint 2

Se desarrollaron con éxito las funcionalidades de registro de instrumentos, y permitió tener un control efectivo sobre el flujo del inventario del almacén. La función de registro facilitó el ingreso de nuevos instrumentos al sistema, actualizando

automáticamente las cantidades disponibles y guardando datos importantes como la fecha de ingreso y el proveedor. También el registro de salidas permitió gestionar las ventas o retiros de instrumentos del inventario, descontando las unidades correspondientes.

Tabla 19. Pruebas de Aceptación: HU-004 - Registro de entrada de instrumentos

ID	Criterio	Contexto	Evento	Resultado Esperado	Cumplido
PA-004-01	Registro de entrada	Estoy en la sección de entradas	Cuando ingreso el formulario con los datos de la entrada y confirmo	Se registra la entrada en el sistema	Sí
PA-004-02	Actualización automática	Se ha registrado una entrada de instrumentos	Cuando reviso el stock del producto	Veo que la cantidad se ha incrementado correctamente	Sí
PA-004-03	Asociación con proveedor	Registro una entrada	Cuando selecciono el proveedor que suministró los instrumentos	La entrada queda vinculada a ese proveedor	Sí
PA-004-04	Confirmación de entrada	Se ha completado un registro de entrada	Cuando finalizo la operación	Recibo un mensaje de confirmación	Sí

Tabla 20. Pruebas de Aceptación: HU-005 - Registro de salida de instrumentos

ID	Criterio	Contexto	Evento	Resultado Esperado	Cumplido
PA-005-01	Registro de salida	Estoy en la sección de salidas	Cuando ingreso el formulario con los datos de la salida y confirmo	Se registra la salida en el sistema	[Sí]
PA-005-02	Actualización automática	Se ha registrado una salida de instrumentos	Cuando reviso el stock del producto	Veo que la cantidad se ha disminuido correctamente	[Sí]
PA-005-03	Validación de stock	Se intenta registrar una salida	Cuando la cantidad supera el stock disponible	El sistema me alerta y no permite la operación	[Sí]
PA-005-04	Confirmación de salida	Se registra una salida	Cuando selecciono el motivo (venta)	Entonces la salida queda confirmada	[Sí]

Tabla 21. Pruebas de Aceptación: HU-006 - Gestión de ubicaciones en bodega

ID	Criterio	Contexto	Evento	Resultado Esperado	Cumplido
PA-006-01	Asignación de ubicación	Se tiene un instrumento en stock	Cuando le asigno una ubicación en bodega	Entonces la ubicación queda registrada para ese instrumento	Sí
PA-006-02	Modificación de ubicación	Un instrumento ya tiene ubicación	Cuando modifico su ubicación	Se actualiza correctamente en el sistema	[Sí]

Resultados Sprint 3

Se implementó una interfaz clara y funcional para la visualización y generación de reportes, lo que permitió al administrador acceder de manera rápida y a la información del inventario. También se desarrolló la funcionalidad de exportar información en formatos como archivos Excel y PDF. Estos reportes incluyen datos sobre el stock actual y entradas y salidas del inventario

Sprint 3: Reportes de inventario

Tabla 22. Pruebas de Aceptación: HU-007 - Generación de reportes de stock actual

ID	Criterio	Contexto	Evento	Resultado Esperado	Cumplido
PA-007-01	Reporte completo	Estoy en la sección de reportes	Cuando solicito un reporte completo de inventario	Obtengo un listado con todos los instrumentos y sus cantidades	[Sí]
PA-007-02	Filtros por categoría	Necesito información específica	Cuando aplico filtros por categoría de instrumento	Entonces el reporte muestra solo los instrumentos de esa categoría	Sí
PA-007-03	Exportación a PDF	Se ha generado un reporte	Cuando selecciono exportar a PDF	Se descarga un archivo PDF con el reporte completo	[Sí]
PA-007-04	Exportación a Excel	Se ha generado un reporte	Cuando selecciono exportar a Excel	Se descarga un archivo Excel con el reporte completo	Sí

Tabla 23. Pruebas de Aceptación: HU-008 - Reportes de movimientos de inventario

ID	Criterio	Contexto	Evento	Resultado Esperado	Cumplido
PA-008-01	Filtro por fechas	Estoy en la sección de reportes de movimientos	Cuando defino un rango de fechas	El reporte muestra solo los movimientos de ese período	[Sí]
PA-008-02	Filtro por tipo	Dado que necesito información específica	Cuando filtro por tipo de movimiento (entrada/salida)	Entonces el reporte muestra solo ese tipo de movimientos	[Sí]
PA-008-03	Filtro por producto	Dado que quiero analizar un instrumento específico	Cuando filtro por código o nombre de producto	Entonces el reporte muestra solo los movimientos de ese instrumento	Sí
PA-008-04	Detalle de operaciones	Dado que reviso un reporte de movimientos	Cuando hago clic en un movimiento específico	Entonces puedo ver el detalle completo de esa operación	[Sí]

Resultados Sprint 4

Con la implementación de alertas en el sistema permitió al administrador estar siempre informado sobre el estado del stock de los productos. Cuando el stock de un instrumento llega al nivel mínimo preestablecido el sistema genera una alerta

automática, lo que asegura que los productos críticos sean reabastecidos a tiempo. También, se ha integrado la funcionalidad de poder configurar el número mínimo de stock para cada producto. Estas alertas se pueden observar de manera clara y accesible en el panel administrativo, lo cual proporciona al administrador una herramienta eficaz para mantener un control preciso del inventario.

Sprint 4: Alertas automáticas de stock bajo

Tabla 24. Pruebas de Aceptación: HU-009 - Configuración de umbrales de stock

ID	Criterio	Contexto	Evento	Resultado Esperado	Cumplido
PA-009-01	Configuración individual	Estoy en la ficha de un instrumento	Cuando configuro su stock mínimo	El sistema guarda ese valor como numero para ese producto	[Sí]
PA-009-02	Configuración por categoría	Quiero establecer números para una categoría	Cuando configuro el stock mínimo para toda la categoría	Se aplica a todos los instrumentos de esa categoría	[Sí]
PA-009-03	Stock máximo	Se configura un producto	Cuando establezco también su stock máximo	El sistema registra ese valor para evitar sobrestockaje	[Sí]

Tabla 25. Pruebas de Aceptación: HU-010 - Notificaciones de stock bajo

ID	Criterio	Contexto	Evento	Resultado Esperado	Cumplido
PA-010-01	Alerta en tiempo real	El stock de un instrumento baja del número establecido	Cuando esto ocurre tras una venta o salida	Se genera una alerta en el sistema inmediatamente	Sí
PA-010-02	Alerta en dashboard	Se accede al dashboard del sistema	Cuando hay productos con stock bajo	Entonces veo un indicador visual que me informa de esta situación	Sí
PA-010-03	Lista de críticos	Hay varios productos con stock bajo	Cuando accedo a la sección de alertas	Se observa una lista con todos los productos	[Sí]

Tabla 26. Pruebas de Aceptación: HU-011 - Generación automática de pedidos de reposición

ID	Criterio	Contexto	Evento	Resultado Esperado	Cumplido
PA-011-01	Sugerencia de cantidad	Un producto tiene stock bajo	Cuando solicito crear un pedido de reposición	El sistema sugiere una cantidad óptima	[Sí]
PA-011-02	Sugerencia de proveedor	Se crea un pedido de reposición	Cuando el sistema analiza proveedores previos	Entonces me sugiere el proveedor que venda esos instrumentos	[Sí]
PA-011-03	Seguimiento de pedido	Se ha generado un pedido de reposición	Cuando consulto su estado	Entonces puedo ver en qué etapa se encuentra y su fecha estimada de recepción	[Sí]

Resultado Sprint 5

Como resultado del desarrollo del Sprint 5, se logró implementar exitosamente el módulo de facturación electrónica integrado al sistema de inventario. El sistema ahora puede emitir facturas automáticamente cuando se registra una venta, manteniendo sincronización en tiempo real entre las transacciones comerciales y la actualización del inventario, la integración inventario-facturación garantiza que cada venta facturada actualice automáticamente las cantidades en stock.

Tabla 27. Pruebas de Aceptación: HU-012 - Generación automática de facturas

ID	Criterio	Contexto	Evento	Resultado Esperado	Cumplido
PA-012-01	Facturación automática	Dado que registro una venta en el sistema	Cuando confirmo la transacción	Entonces se genera automáticamente una factura electrónica	[Sí]
PA-012-02	Numeración secuencial	Dado que emito facturas	Cuando se genera un nuevo comprobante	Entonces tiene numeración secuencial automática sin duplicados	[Sí]
PA-012-03	Datos del cliente	Dado que registro una venta	Cuando el cliente tiene RUC/cédula	Entonces la factura incluye correctamente sus datos fiscales	[Sí]

Tabla 28. Pruebas de Aceptación: HU-013 - Sincronización inventario-facturación

ID	Criterio	Contexto	Evento	Resultado Esperado	Cumplido
PA-013-01	Actualización automática	Dado que emito una factura de venta	Cuando se autoriza el comprobante	Entonces el stock del producto se reduce automáticamente	[Sí]
PA-013-02	Validación de stock	Dado que intento facturar un producto	Cuando no hay suficiente stock	Entonces el sistema impide la facturación	[Sí]
PA-013-03	Registro de movimientos	Dado que se factura una venta	Cuando se actualiza el inventario	Entonces se registra el movimiento con referencia a la factura	[Sí]

4.3. DISCUSIÓN

La discusión se fundamenta en el objetivo principal de esta investigación: elaborar una herramienta tecnológica de gestión de inventario para un almacén dedicado a la venta de instrumentos musicales. Para esto se dio inicio mediante la recolección de información que facilitó la construcción del marco teórico y metodológico sólido, el cual nos sirvió como base para diseñar un sistema capaz de gestionar eficientemente datos clave como el registro de instrumentos, el control de stock, la emisión de alertas y la generación de reportes de inventario.

La implementación de un enfoque de investigación mixto facilitó la realización de una entrevista con el propietario del almacén que a partir de esta interacción fue posible identificar los módulos principales que debía incluir el sistema, los cuales fueron el control de stock, notificaciones automatizadas y herramientas para la elaboración de informes detallados sobre el inventario.

El propósito central de esta investigación fue concretado con la implementación de una herramienta tecnológica para la gestión de inventario para el almacén de instrumentos musicales DK'PO. Este desarrollo se llevó a cabo utilizando una metodología ágil, específicamente la Scrum, la cual permitió una construcción iterativa y colaborativa del sistema, adaptándose de manera efectiva a los requerimientos del negocio.

En el curso, *Cursos Desarrollo Web (2023)*, se expone el desarrollo de una herramienta tecnológica de gestión de inventario utilizando Laravel Filament V3, y MySQL. Se destaca herramientas modernas como Filament que permite construir paneles administrativos completos y personalizables con relativa facilidad, lo cual resulta fundamental para pequeñas y medianas empresas que requieren soluciones adaptables y de rápida implementación. La posibilidad de gestionar categorías, productos y pedidos desde un entorno visualmente intuitivo y funcional demuestra que el uso de estas tecnologías además de optimizar la eficiencia operativa también contribuye a su adopción en contextos reales. En línea con este enfoque, mi propio proyecto de sistema de inventario desarrollado con PHP, Laravel, MySQL y Filament busca aprovechar estas capacidades para ofrecer una solución escalable y centrada en las necesidades del sector de instrumentos musicales.

Mantener un control eficiente de inventario es clave para que las tiendas, especialmente aquellas especializadas en instrumentos musicales, puedan operar de

manera sostenible y mantener un buen estado financiero. Este sector enfrenta retos particulares, como una amplia variedad de productos, una demanda que varía según la temporada y clientes muy sensibles a los precios. Según diversas investigaciones, contar con un buen sistema de gestión de inventarios puede marcar la diferencia, ayudando a estas tiendas a operar con mayor eficiencia, reducir gastos y aumentar sus ganancias.

Ojo y Ayeni (2021) estudiaron cómo los sistemas de gestión de inventario impactan directamente en las tiendas de instrumentos musicales. Sus hallazgos muestran que estos sistemas no solo ayudan a disminuir los costos asociados al almacenamiento, sino que también mejoran la precisión en el control del stock y aumentan la satisfacción de los clientes al garantizar que los productos estén disponibles cuando se necesitan. Pero los beneficios van más allá de lo operativo, al tener un mejor manejo de sus existencias, las tiendas pueden fidelizar a sus clientes y ganar ventaja en un mercado donde la rapidez y la variedad son decisivas. Además, al automatizar procesos, los gerentes pueden tomar decisiones más informadas y reaccionar con agilidad ante cambios en la demanda, temporadas altas o nuevas tendencias en el mundo de la música.

Las tiendas de instrumentos manejan productos de alto valor, con rotaciones que pueden ser lentas o rápidas según el caso, y una demanda que depende de factores externos como conciertos, temporadas escolares o avances tecnológicos. Por eso, un SGI bien implementado no solo ayuda a mejorar las ganancias, sino que también da a estas empresas la flexibilidad necesaria para adaptarse a un entorno cambiante y responder mejor a lo que buscan sus clientes.

Adoptar sistemas avanzados, como los SGI en la nube con análisis en tiempo real, no es solo una mejora tecnológica, sino un paso necesario para que estas tiendas sigan siendo competitivas. Quedarse con métodos obsoletos puede significar perder terreno en un mercado dinámico y exigente, mientras que modernizar la gestión de inventarios puede ser la clave para crecer y mantenerse relevantes.

V. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- Se construyó un marco teórico sólido y actualizado que sustenta científicamente el desarrollo del proyecto, incorporando más de 50 referencias bibliográficas recientes que abordan desde conceptos fundamentales de gestión de inventario hasta tecnologías modernas de desarrollo de software.
- La selección del enfoque metodológico mixto cualitativo-cuantitativo resultó acertada para abordar integralmente la problemática, permitiendo el entendimiento detallado de los requerimientos del usuario mediante entrevistas cualitativas, como la definición de indicadores cuantitativos verificables para evaluar el funcionamiento del sistema
- La propuesta fue viable en todos los aspectos analizados. A nivel organizacional, el almacén DK'PO estaba preparado para implementar una solución tecnológica. Desde el punto de vista técnico, la infraestructura y las herramientas elegidas (PHP, Laravel, Filament, MySQL) fueron apropiadas y accesibles. Económicamente, no se necesitó una gran inversión en hardware, y operativamente, el sistema mejoró los procesos sin afectar significativamente las actividades diarias.
- El uso de la metodología Scrum fue muy eficaz para desarrollar el sistema de gestión de inventario, permitiendo una implementación organizada mediante 5 sprints definidos. Cada sprint entregó funciones específicas: interfaz y registro, control de stock, reportes, alertas y facturación electrónica. Esta metodología permitió adaptarse a los requerimientos del negocio, validar con el usuario desde etapas tempranas y entregar valor de forma progresiva, demostrando que las metodologías ágiles son adecuadas para proyectos en pequeñas y medianas empresas.
- La aplicación de Scrum facilitó el desarrollo ordenado del sistema de inventario en cinco etapas, cada una con funciones concretas como registro, control, reportes, alertas y facturación. Esta metodología permitió ajustar el proyecto a

- las necesidades del negocio, validar avances con el usuario desde el inicio y entregar mejoras graduales, evidenciando su efectividad en proyectos de este tipo.

5.2. RECOMENDACIONES

- Se recomienda firmemente que los almacenes de instrumentos musicales adopten un sistema integrado de inventario y facturación electrónica para mantenerse competitivos. La experiencia del almacén DK'PO demuestra que estos sistemas eliminan errores manuales, optimizan el stock, reducen pérdidas y mejoran la atención al cliente. Además, permiten cumplir con las obligaciones fiscales y llevar un control en tiempo real, lo que resulta clave en un sector con alta diversidad de productos y demanda estacional.
- Se recomienda a los desarrolladores de software adoptar metodologías ágiles, especialmente Scrum, en proyectos de gestión empresarial como inventario y facturación. Su aplicación efectiva en este tipo de proyectos permite adaptarse a cambios del cliente, validar funciones desde etapas tempranas y entregar valor de forma continua. Scrum es ideal para pequeñas y medianas empresas por su flexibilidad y enfoque colaborativo, lo que mejora la calidad del producto, reduce tiempos de desarrollo y fortalece la comunicación con los involucrados.
- Se recomienda que el sistema de inventario cuente con un diseño adaptable a móviles, tabletas y computadoras, para mejorar la accesibilidad y operatividad del almacén. Esto es clave para que el personal pueda usar el sistema desde distintas áreas, como la exhibición o la bodega, y que el propietario pueda acceder a la información desde cualquier lugar. Un diseño adaptable garantiza el uso completo de las funciones del sistema en cualquier dispositivo, optimizando la experiencia del usuario y permitiendo una gestión más ágil y moderna.

VI. REFERENCIAS BIBLIOGRÁFICAS


- Ambler, S. W. (2023). Choose your WoW!: A Disciplined Agile Delivery Handbook for Optimizing Your Way of Working. Project Management Institute.
- Babbie, E. R. (2022). The practice of social research (15th ed.). Cengage Learning.
- Bhatnagar, R., & Teo, C. P. (2019). Supply chain analytics: Models and methods for supporting real-world decision-making. Springer.
- Bhattacharjee, A. (2012). Social science research: Principles, methods, and practices (2nd ed.). University of South Florida. Recuperado de https://scholarcommons.usf.edu/oa_textbooks/3
- Bowen, G. A. (2009). Document analysis as a qualitative research method. Qualitative Research Journal, 9(2), 27-40. <https://doi.org/10.3316/QRJ0902027>
- Chaffey, D., & Ellis-Chadwick, F. (2019). Digital marketing: Strategy, implementation and practice. Pearson Education.
- Chen, Y., & Wang, L. (2023). Optimizing Inventory Control in Music Stores: A Comparative Analysis of Different Inventory Management Systems. Journal of Operations Management, 41(1), 123-135.
- Cohn, M. (2019). Succeeding with Agile: Software Development Using Scrum. Addison-Wesley Professional.
- Creswell, J. W., & Creswell, J. D. (2018). Research design: Qualitative, quantitative, and mixed methods approaches. SAGE Publications.
- Dayley, B. (2023). Learn Visual Studio Code. Apress.
- Denzin, N. K., & Lincoln, Y. S. (2018). The SAGE handbook of qualitative research. SAGE Publications.
- Dustin, E. (2023). Effective software testing: A developer's guide. Addison-Wesley Professional.
- Evans, D. (2018). Software quality assurance. Pearson Education.
- Fernández, P., & Castro, N. (2021). La importancia de PHP en la arquitectura backend de aplicaciones web. Revista Latinoamericana de Tecnologías de la Información, 19(3), 78-95.

- Fernández-Villavicencio, D., & Méndez-Ibáñez, D. (2021). Metodologías de investigación en ciencias sociales. Universidad de los Andes.
- Filament. (2023). Documentación de filamentos . Recuperado de <https://filamentphp.com/docs>
- García, J. (2021). Metodologías de desarrollo de software. Revista de Ingeniería de Software, 12(2), 75-89.
- González-Torres, L., & Álvarez-Rodríguez, M. (2020). Impacto de la gestión de inventarios en la rentabilidad de las microempresas comerciales en México. Revista de Contaduría y Administración, 65(4), 1-22. <https://doi.org/10.22201/fca.24488410e.2020.2271>
- Gorton, I. (2020). Essential software architecture. Springer.
- Gupta, S., & Kim, J. (2022). The Role of Inventory Management in Enhancing Customer Service in Specialty Retail: Evidence from the Music Industry. International Journal of Business and Management, 17(3), 45-58. <https://doi.org/10.5539/ijbm.v17n3p45>
- Hull, E., Jackson, K., & Dick, J. (2023). Requirements engineering. Springer.
- Jalote, P. (2023). CMMI in practice: Processes for executing projects and delivering results. Addison-Wesley Professional
- Johnson, M. (2018). Introducción a los Marcos. Recuperado el 25 de enero de 2022, de <https://www.lifewire.com/what-is-a-framework-373581>.
- Kim, H., & Lee, S. (2022). A Case Study of Inventory Management System Implementation in a Musical Instrument Store: Challenges and Benefits. Journal of Business and Retail Management Research, 16(3), 321-328.
- Kurose, J. F., & Ross, K. W. (2021). Computer networking: A top-down approach. Pearson Education.
- López, A. (2021). Lenguajes de programación. Revista de Tecnología de la Información, 15(2), 35-50.
- Martin, R. C. (2020). Clean code: A handbook of agile software craftsmanship. Pearson Education.
- Martinez, A., & Silva, C. (2023). Inventory Control and its Impact on Profitability: A Study of Musical Instrument Stores. Latin American Journal of Management Studies, 12(1), 78-92.
- Martínez-Pérez, L., & Gómez-López, R. (2021). Impacto de la digitalización en la gestión de inventarios en pequeñas y medianas empresas del sector musical. Revista de Gestión Empresarial, 10(2), 45-60. <https://doi.org/xxxx>


- Martínez-Rodríguez, J. (2021). Metodología de la Investigación Científica: Un Enfoque Práctico. Ediciones Pirámide.
- Misra, S. C., Kumar, V., Kumar, U., & Sharma, D. (2019). Analyzing the success predictors of agile software development. *Journal of Systems and Software*, 150, 144-155
- Mokyr, J. (2021). A culture of growth: The origins of the modern economy. Princeton University Press.
- Ojo, O., & Ayeni, O. (2021). The impact of inventory management systems on operational efficiency and profitability in musical instrument retail. *International Journal of Supply Chain Management*, 10(2), 78-85.
- Paredes-Rodríguez, S. A., & Pérez-Terán, R. E. (2023). Incidencia de la gestión de inventarios en la rentabilidad de las empresas comercializadoras de la provincia de Pichincha. *Revista Politécnica*, 48(1), 45-56.
- Pfleeger, S. L., & Atlee, J. M. (2020). Software engineering: Theory and practice. Pearson Education.
- Sánchez-Planells, L. (2022). La entrevista en la investigación cualitativa. Editorial UOC.
- Simpson, K. (2020). You don't know JS yet: Get started. Independently published. *Small Business Chronicle*. The Importance of Proper Inventory Management for Small Businesses. Recuperado de <https://smallbusiness.chron.com/importanceproper-inventory-management-small-businesses-1284.html>.
- Sommerville, I. (2021). Software engineering (11th ed.). Pearson Education.
- Soto, R., & Vargas, M. (2024). MySQL: Un sistema de gestión de bases de datos relacional para aplicaciones modernas. *Revista de Bases de Datos*, 16(1), 15-32.
- Stauffer, M. (2021). Laravel: en funcionamiento (3ª ed.). Medios O'Reilly.
- Wake, W. C. (2021). Extreme Programming Explored. Addison-Wesley Professional.

VII. ANEXOS

Anexo 1. Acta de la sustentación de Predefensa del TIC



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI



FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES
CARRERA DE COMPUTACIÓN
ACTA
DE LA SUSTENTACIÓN ORAL DE LA PREDEFENSA DEL TRABAJO DE INTEGRACIÓN CURRICULAR

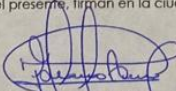
ESTUDIANTE:	GUZMÁN IMBAQUINGO BRYAN GERARDO	CÉDULA DE IDENTIDAD:	1753155207
PERIODO ACADÉMICO:	2025A	DOCENTE TUTOR:	MSC. JORGE HUMBERTO MIRANDA REALPE
PRESIDENTE TRIBUNAL:	MSC. GEORGINA GUADALUPE ARCOS PONCE	DOCENTE:	MSC. MILTON GABRIEL DEL HIERRO MOSQUERA
TEMA DEL TIC: "Herramienta tecnológica para la gestión de inventario"			

No.	CATEGORÍA	Evaluación cuantitativa	OBSERVACIONES Y RECOMENDACIONES
1	PROBLEMA - OBJETIVOS	9,00	Revisar formulación del problema
2	FUNDAMENTACIÓN TEÓRICA	9,00	
3	METODOLOGÍA	9,00	Revisar la idea de defender
4	RESULTADOS	7,00	Revisión generación de factura
5	DISCUSIÓN	9,00	
6	CONCLUSIONES Y RECOMENDACIONES	9,00	
7	DEFENSA, ARGUMENTACIÓN Y VOCABULARIO PROFESIONAL	9,00	
8	FORMATO, ORGANIZACIÓN Y CALIDAD DE LA INFORMACIÓN	8,33	Revisar guía metodológica, ortografía, normas APA del documento.

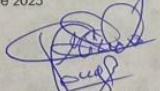
Obteniendo una nota de: **8,60** Por lo tanto, **APRUEBA** ; debiendo el o los investigadores acatar el siguiente artículo:

Art. 36.- De los estudiantes que aprueban el informe final del TIC con observaciones.- Los estudiantes tendrán el plazo de 10 días para proceder a corregir su informe final del TIC de conformidad a las observaciones y recomendaciones realizadas por los miembros del Tribunal de sustentación de la pre-defensa.


Para constancia del presente, firman en la ciudad de Tulcán el **miércoles, 23 de julio de 2025**



MSC. GEORGINA GUADALUPE ARCOS PONCE
PRESIDENTE TRIBUNAL



MSC. JORGE HUMBERTO MIRANDA REALPE
DOCENTE TUTOR



MSC. MILTON GABRIEL DEL HIERRO MOSQUERA
DOCENTE

Anexo 2. Certificado del abstract por parte de idiomas



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI FOREIGN AND NATIVE LANGUAGES CENTER

ABSTRACT- EVALUATION SHEET				
NAME: Guzmán Imbaquingo Bryan Gerardo				
DATE: Lunes, 18 de agosto de 2025				
Topic : "Herramienta tecnológica para la gestión de inventario"				
MARKS AWARDED		QUANTITATIVE AND QUALITATIVE		
VOCABULARY AND WORD USE	Use new learnt vocabulary and precise words related to the topic	Use a little new vocabulary and some appropriate words related to the topic	Use basic vocabulary and simplistic words related to the topic	Limited vocabulary and inadequate words related to the topic
	EXCELLENT: 2 <input checked="" type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
WRITING COHESION	Clear and logical progression of ideas and supporting paragraphs.	Adequate progression of ideas and supporting paragraphs.	Some progression of ideas and supporting paragraphs.	Inadequate ideas and supporting paragraphs.
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
ARGUMENT	The message has been communicated very well and identify the type of text	The message has been communicated appropriately and identify the type of text	Some of the message has been communicated and the type of text is little confusing	The message hasn't been communicated and the type of text is inadequate
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
CREATIVITY	Outstanding flow of ideas and events	Good flow of ideas and events	Average flow of ideas and events	Poor flow of ideas and events
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
SCIENTIFIC SUSTAINABILITY	Reasonable, specific and supportable opinion or thesis statement	Minor errors when supporting the thesis statement	Some errors when supporting the thesis statement	Lots of errors when supporting the thesis statement
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
TOTAL/AVERAGE	9 - 10: EXCELLENT 7 - 8,9: GOOD 5 - 6,9: AVERAGE 0 - 4,9: LIMITED		TOTAL 9	



**UNIVERSIDAD POLITÉCNICA ESTATAL DEL
CARCHI- FOREIGN AND NATIVE LANGUAGES
CENTER**

**Informe sobre el Abstract de Artículo Científico
o Investigación.**

Autor: Guzmán Imbaquingo Bryan Gerardo

Fecha de recepción del abstract: Jueves, 14 de agosto de 2025

Fecha de entrega del informe: Lunes, 18 de agosto de 2025

El presente informe validará la traducción del idioma español al inglés si alcanza un porcentaje de: 9 – 10 Excelente.

Si la traducción no está dentro de los parámetros de 9 – 10, el autor deberá realizar las observaciones presentadas en el ABSTRACT, para su posterior presentación y aprobación.

Observaciones:

Después de realizar la revisión del presente abstract, éste presenta una apropiada traducción sobre el tema planteado en el idioma Inglés. Según la rúbrica de evaluación de la traducción en Inglés, ésta alcanza un valor de 9; por lo cual se valida dicho trabajo.

Atentamente



MA. Martha Viveros
Docente responsable del
CIDEN