PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADORFACULTAD DE INGENIERÍA
CARRERA DE: SISTEMAS DE INFORMACIÓN



Trabajo de Titulación

Tema: Desarrollo de prototipo funcional de un sistema de gestión deinventario y ventas web para una óptica.

Caso de estudio: ProdeOptica

AUTOR:

Matías Nicolas Gamboa Escobar

QUITO DM, OCTUBRE DE 2024

Contenido

CAPÍTULO I: INTRODUCCIÓN 1
1.1 Objetivos del proyecto 1
1.1.1 Objetivo general 1
1.1.2 Objetivos específicos 1
1.2 Descripción de la óptica 1
1.3 Procesos actuales de la gestión de inventario y ventas 2
1.4 Problemática actual 3
1.5 Justificación del proyecto 4
1.6 Alcance del proyecto 5
CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA 6
2. Marco teórico 6
2.1 Desarrollo web 6
2.1.1 Frontend 6
2.2.1 Backend 7
2.2 Herramientas de desarrollo y frameworks 7
2.2.1 React 7
2.2.2 Next.js 8
2.2.3 Tailwind CSS 9
2.2.4 Django 9
2.2.5 Fetch 10
2.2.6 Tabla comparativa frameworks frontend 10
2.2.7 Tabla comparativa de frameworks backend 10
2.2.8 Tabla comparativa de librerías CSS 11
2.3 Base de datos 12
2.3.1 PostgreSQL 12
2.3.2 Django ORM 12
2.4 Tecnologías de soporte 12
2.4.1 Git 12
2.4.2 GitHub 13
2.4.3 Postman 13
2.4.5 Excalidraw 13
2.4.6 Visual Studio Code 13
2.4.7 PyCharm 13
2.4.8 Notion 14
2.5 Gestión de inventario y ventas 14

2.5.1 Gestión de inventario 14
2.5.2 Sistema de inventario perpetuo 15
2.5.3 Stock 15
2.5.4 Registro de ventas 15
2.5.5 Kardex 15
2.6 Seguridad en el desarrollo web 16
2.6.1 Autenticación 16
2.6.2 Autorización 16
2.6.3 JSON Web Tokens 16
2.7 Diagramas 17
2.7.1 Mockups 17
2.7.2 Diagrama de arquitectura de software 17
2.7.3 Diagrama de flujo de navegación 17
2.8 Hosting del sistema 18
2.8.1 Plataformas de hosting 18
2.8.2 Vercel 18
2.8.3 Heroku 18
2.9 Desarrollo de software y software a medida 18
CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA 19
3.1 Requisitos funcionales y no funcionales 19
3.1.1 Requisitos funcionales 19
3.1.2 Requisitos no funcionales 21
3.2 Arquitectura del sistema 22
..... 24
3.2.1 Base de datos – Servidor 25
3.2.2 Servidor – Cliente 25
3.2.3 Firewall 26
3.2.4 Internet 26
3.2.5 Cliente 26
3.3 Metodologías y gestión de proyectos 26
3.3.1 Metodologías ágiles 26
3.3.2 Kanban 27
3.3.3 SCRUM 27
3.3.4 Principios y valores SCRUM 28
3.3.5 Artefactos SCRUM 29
3.3.6 Roles SCRUM 30

3.3.7 Eventos SCRUM 31
3.3.8 Tabla comparativa de metodologías 31
3.3.9 Monday.com 32
3.3.10 Viabilidad de desarrollo individual en un marco ágil 33
3.3.11 Gestión de múltiples roles y mitigación de conflictos 33
CAPÍTULO IV: DESARROLLO 34
4.1 Diseño de base de datos y programación de API REST 34
4.1.1 Introducción y planificación del sprint 34
4.1.2 Sprint backlog 35
4.1.3 Historias de usuario 36
4.1.4 Resultados y retrospectiva del sprint 37
4.1.5 Pruebas realizadas 37
4.2 Registro de usuarios e inicio de sesión 38
4.2.1 Introducción y planificación del sprint 38
4.2.2 Sprint backlog 39
4.2.3 Historias de usuario 40
4.2.4 Resultados y retrospectiva del sprint 41
4.2.5 Pruebas realizadas 42
4.3 Gestión de inventario 1/2 43
4.3.1 Introducción y planificación del sprint 43
4.3.2 Sprint backlog 44
4.3.3 Historias de usuario 45
4.3.4 Resultados y retrospectiva del sprint 46
4.3.5 Pruebas realizadas 46
4.4 Gestión de inventario 2/2 47
4.4.1 Introducción y planificación del sprint 47
4.4.2 Sprint backlog 48
4.4.3 Historias de usuario 49
4.4.4 Resultados y retrospectiva del sprint 49
4.4.5 Pruebas realizadas 50
4.5 Gestión de ventas 51
4.5.1 Introducción y planificación del sprint 51
4.5.2 Sprint backlog 52
4.5.3 Historias de usuario 53
4.5.4 Resultados y retrospectiva del sprint 54
4.5.5 Pruebas realizadas 55

4.6 Gestión de reportes gráficos 56
4.6.1 Introducción y planificación del sprint 56
4.6.2 Sprint backlog 57
4.6.3 Historias de usuario 58
4.6.4 Resultados y retrospectiva del sprint 59
4.6.5 Pruebas realizadas 60
4.7 Gestión de stock y optimización de la interfaz de usuario 60
4.7.1 Introducción y planificación del sprint 60
4.7.2 Sprint backlog 61
4.7.3 Historias de usuario 62
4.7.4 Resultados y retrospectiva del sprint 62
4.7.5 Pruebas realizadas 63
CONCLUSIONES Y RECOMENDACIONES 64
5.1 Conclusiones 64
5.2 Recomendaciones 65
REFERENCIAS 66
ANEXOS 72

CAPÍTULO I: INTRODUCCIÓN

1.1 Objetivos del proyecto

1.1.1 Objetivo general

Desarrollar un prototipo funcional de un sistema de gestión de inventario y ventas para una óptica, que permita maximizar la eficiencia de los procesos operativos mediante la automatización y la mejora en el manejo de información.

1.1.2 Objetivos específicos

- Diseñar una interfaz intuitiva y amigable para el usuario que permita el control del inventario de una manera eficiente, reduciendo errores y tiempos de procesamiento.
- Implementar un sistema de almacenamiento centralizado que garantice la consistencia e integridad de los datos mediante validaciones en el backend de la aplicación, facilitando así el manejo de un inventario unificado y sincronizado entre las sucursales.
- Incorporar funcionalidades de automatización, como alertas y generación de reportes personalizados, mejorando la eficiencia operativa de la óptica.

1.2 Descripción de la óptica

ProdeOptica es una empresa dedicada a la salud visual de sus clientes, ofreciendo una gran cantidad de productos de alta calidad. Además de sus productos, la óptica cuenta con especialistas altamente certificados en distintas áreas de la salud visual, como oftalmología, contactología y

optometría, con formación destacada en instituciones europeas.

Además de sus productos y servicios, ProdeOptica destaca por su énfasis en la innovación tecnológica, utilizando las mejores herramientas disponibles en el mercado con el fin de mejorar la experiencia de sus pacientes. El compromiso de la empresa requiere un esfuerzo continuo para estar al día con todos los avances en el ámbito de la optometría y así proporcionar soluciones a las necesidades del mercado.

En la actualidad, el proceso de trabajo incluye tareas manuales para la administración y gestión del inventario y de las ventas. Esta perspectiva de los procesos de la empresa nos muestra la importancia de establecer un sistema de gestión que ayude a asegurar la exactitud de los datos y también optimice el servicio al cliente.

1.3 Procesos actuales de la gestión de inventario y ventas

En la actualidad, la empresa ProdeOptica hace uso de hojas de cálculo para el manejo de inventario, una solución viable para pequeñas pymes; sin embargo, la óptica ya cuenta con un flujo mayor de ventas, por lo que se han empezado a evidenciar varias limitaciones. Estas hojas de cálculo son administradas por dos o tres personas por cada sucursal, que ingresan manualmente las entradas y salidas de datos.

El procedimiento operativo actual implica la actualización y el seguimiento de los datos de inventario y venta. Además, la gestión del inventario se realiza a nivel local, sin que haya conexión entre las sucursales, lo que complica de gran manera lograr obtener una perspectiva general de la empresa. El incremento en la demanda operativa crea la necesidad de un sistema que posibilite mejor gestión.

1.4 Problemática actual

Actualmente, la empresa hace manejo del inventario y ventas mediante hojas de cálculo, lo cual representa una limitación significativa. Esta forma de manejar los procesos de la empresa ha generado varios problemas de consistencia de datos entre hojas de cálculo, como errores en los datos del inventario. Llegando a ofrecer productos o modelos inexistentes. Después de realizar un análisis, se determinó que existe aproximadamente un 8% de discrepancias debido a los registros manuales. Por ejemplo, en una ocasión reciente se han presentado inconsistencias de inventario, donde se presentaba un modelo como disponible, pero al buscarlo, no se encontró el producto, lo que resultó en que el cliente tuviera que buscar otro modelo, lo cual generó un descontento y pérdida de confianza del cliente.

Además, esta herramienta consume una gran cantidad de tiempo, especialmente en situaciones con múltiples clientes. El personal de ambas sucursales invierte en promedio 10 horas semanales verificando el inventario físico y comparando los datos de las hojas de cálculo, lo que afecta la experiencia de los clientes y también generando una demora operativa. Con la sucursal nueva, existe la necesidad de manejar un inventario centralizado, lo cual resulta complicado sin un sistema diseñado con esta funcionalidad.

Aunque el plan para el sistema ya incluye una fase de pruebas y de validación, es importante aclarar que se realizarán pruebas exhaustivas, incluyendo pruebas unitarias, pruebas de integración y pruebas de usuario. Esto se realizará con el fin de garantizar el cumplimiento de los requisitos planteados. Por otro lado, es de suma importancia establecer formas de proteger los datos dentro del sistema informático.

1.5 Justificación del proyecto

El desarrollo de este sistema es necesario para conseguir una adecuada gestión en el inventario en esta empresa. Al ser una empresa pequeña, un sistema de este tipo permitirá lamejora de la eficiencia operativa y en la atención al cliente. El sistema está diseñado para facilitar la entrada y salida de productos. Según un informe de McKinsey & Company detalla cómo las empresas pequeñas o de mediano tamaño pueden automatizar hasta un 45% de sus actividades y esto a su vez mejora la productividad hasta en un 30% (McKinsey Global Institute, 2017). Esto ayudará a mantener un manejo más confiable y ágil del inventario de la empresa. El sistema fue desarrollado con la finalidad de simplificar la entrada y salida de productos, minimizar los errores humanos y también automatizar procesos clave, garantizando un manejo más eficiente y seguro del inventario.

Este sistema es esencial para permitir una adecuada administración de los recursos disponibles. Dado que ProdeOpticva es una empresa que cuenta con recursos limitados, es necesario maximizar la eficiencia de los procesos operativos con el fin de generar un impacto positivo en la experiencia del cliente. También se busca garantizar una disponibilidad precisa de productos, mejorando la capacidad de respuesta frente a la demanda. De esta manera, se logrará fortalecer la relación con los clientes, mejorando no solo su experiencia, sino también su satisfacción y fidelidad.

En el mercado actual, existe una gran variedad de soluciones; sin embargo, muchas de ellas presentan costos elevados y falta de flexibilidad para adaptarse a la empresa ProdeOptica. En contraste, un sistema a medida creado exclusivamente para la empresa garantiza una solución más eficaz y alineada con sus objetivos. Por ejemplo, una plataforma comercial puede costar entre 500$ y 2000$ solo en las licencias mientras que el sistema creado a la medida ofrece un

control total y una inversión inicial fija (Zoho Inventory, 2025). Además, este enfoque le permitirá a ProdeOptica posicionarse en el mercado como una empresa competitiva mejorando sus procesos internos.

De igual manera, teniendo en cuenta un punto de vista de costo-beneficio, el sistema disminuirá considerablemente el tiempo dedicado a tareas manuales y los errores en la administración del inventario. Esto mejorará tanto los procesos internos como la experiencia de los clientes, reforzando la competitividad de la empresa. Al reducir el tiempo usado en conciliaciones manuales y prevenir los problemas de datos, el sistema se convertirá no solo en una herramienta funcional, sino también en un soporte para la toma de decisiones.

1.6 Alcance del proyecto

El proyecto consiste en desarrollar un software a medida para la empresa, incorporando un prototipo funcional con gestión de inventario y ventas que serán definidas en conjunto con el cliente. El sistema debe ser capaz de solucionar los problemas de gestión actuales de la empresa, además de incorporar funcionalidades adicionales acordadas durante el proceso de desarrollo, como la generación de informes, alertas de inventario y visualización de Kardex.

Durante y posterior al desarrollo, se realizarán pruebas del prototipo con la participación del personal de la óptica; de esta manera, se puede identificar posibles mejoras o ajustes previo a la entrega del prototipo, garantizando que el sistema cumpla con las expectativas del cliente.

Además de lo anteriormente mencionado, se considerará la escalabilidad del sistema y el mantenimiento que este va a requerir a largo plazo. Esto teniendo en mente que la empresa podría continuar creciendo e incluso abrir nuevas sucursales. El sistema será desarrollado para poder manejar un mínimo de 5 usuarios concurrentes, lo que, a pesar de ser poco,

satisface las necesidades actuales de la empresa con un margen de crecimiento. El prototipo utilizará servicios de hosting gratuitos, por lo que las configuraciones del servidor son aproximadamente 2 núcleos CPU, 512 MB a 1 GB de RAM y almacenamiento SDD, lo cual es suficiente para el tráfico esperado. Además, en caso de que exista un crecimiento en el uso, los recursos son fácilmente escalables utilizando planes más avanzados de las mismas plataformas.

CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA

2. Marco teórico

2.1 Desarrollo web

2.1.1 Frontend

El frontend de un sistema web se refiere a todos los elementos que pueden ser vistos por el usuario, incluyendo botones, texto, gráficos, entre otros elementos que permiten al usuario interactuar con la aplicación. De manera más general, hablamos de interfaz gráfica del usuario, también conocida como GUI (AWS, 2023).

De manera más general, el frontend se refiere específicamente a la sección del desarrollo web que interactúa con los usuarios de un sistema. El frontend está compuesto por tecnologías como JavaScript, HTML y CSS, además de los frameworks actuales como Next, Angular o Vue. Estas herramientas son utilizadas para la creación de interfaces más escalables, interactivas y adaptables (Vyas, 2022). Por ejemplo, Netflix es una de las plataformas más grandes a nivel global que utiliza React para poder ofrecer las mejores experiencias a los usuarios. La desarrollo sobre este framework ha permitido mejorar la interacción con los usuarios, lo que a su vez ha mejorado la retención de los usuarios (*Netflix TechBlog*, 2025).

2.2.1 Backend

El manejo de los datos y la estructura es fundamental en cualquier aplicación para su correcto funcionamiento. El backend permite administrar todas las funcionalidades de la aplicación web, es decir, es el componente encargado de procesar las solicitudes realizadas por los usuarios y devolver una respuesta para que pueda ser visualizada en el frontend (AWS, 2023). Además, el backend es el encargado de gestionar la lógica del negocio implementando las reglas.

En general, el backend es el encargado de controlar y gestionar las distintas validaciones de los datos que se ingresen en el sistema. Además, el backend también permite la implementación de alertas o notificaciones. Esto nos permitirá asegurar la integridad y la consistencia de los datos relacionados con el inventario (Yohan et al., 2025). Por ejemplo, una de las redes sociales más grandes del mundo, Instagram, utiliza el framework Django para su backend. Esto les ha permitido manejar de manera óptima el gran volumen de datos y solicitudes de su plataforma. También les ha permitido la implementación de lógica compleja que permite realizar recomendaciones y gestión de contenido personalizado (*LinkedIn*, 2025).

2.2 Herramientas de desarrollo y frameworks

2.2.1 React

Es una librería de JavaScript sumamente popular creada para el desarrollo de interfaces dinámicas en aplicaciones web. Esta herramienta permite la creación de componentes personalizables y reutilizables que conforman las páginas de una aplicación web. Es importante mencionar que React no es un framework; sin embargo, además de una librería, puede ser considerado una arquitectura debido a la forma que ofrece para estructurar aplicaciones (*React*, 2015).

En comparación con otros frameworks como Angular o Vue, React y los frameworks basados en

esta librería destacan por su capacidad para actualizar y renderizar los componentes necesarios, esto debido a su enfoque en el sistema de virtualización del DOM. Esto garantiza una mejora en el rendimiento de un sistema “al minimizar las actualizaciones innecesarias en la interfaz de usuario” (Borovskoy, 2025).

React también sobresale por tener una curva de aprendizaje más accesible. Por otro lado, Angular cuenta con una solución igual o incluso más completa, pero también con una estructura mucho más rígida y una mayor curva de aprendizaje, React permite tener una mayor flexibilidad y libertad al crear páginas web. Por otro lado, Vue también es una solución flexible y más sencilla de aprender; sin embargo, React cuenta con un mayor soporte y una mayor adopción en proyectos empresariales, convirtiéndola en una excelente elección para el proyecto (*Comparación Con Otros Frameworks — Vue.js*, 2015).

2.2.2 Next.js

Framework desarrollado por Vercel que ha tomado un gran impulso en el desarrollo debido a ser impulsado por herramientas como React, Turbopack o Speedy. Next.js cuenta con un gran número de funcionalidades que lo hacen destacar, por ejemplo, la optimización integrada para mejorar la experiencia de los usuarios, transmisión dinámica de HTML mediante App Router, el renderizado cliente y servidor flexible, entre otras características que lo hacen sumamente útil en cualquier tipo de proyecto (Vercel, 2024). A pesar de presentar herramientas innovadoras para la programación web, la alianza con React permite un framework muy flexible y relativamente fácil de usar, con una curva de aprendizaje baja si se tienen conocimientos previos con el uso de React.

Entre las características más importantes de este framework, se encuentra su capacidad de renderizado híbrido. Esta característica integra tanto la generación de sitios estáticos (SSG) y el renderizado en el servidor (SSR). Este enfoque permite a las aplicaciones y páginas web

aprovechar las ventajas de ambos enfoques, proporcionando contenido dinámico solo cuando sea necesario. Esto es especialmente útil cuando se trabaja en una aplicación que maneje datos que cambian muy frecuentemente (Vercel, 2024).

2.2.3 Tailwind CSS

Es un framework CSS que busca dar mayor importancia al uso que al propio estilo. En otras palabras, Tailwind CSS no proporciona componentes predefinidos como otros frameworks; en su lugar tiene clases CSS que permiten crear la estructura y estilizar componentes de manera personalizada (Huet, 2022).

Utilizando Tailwind CSS, se pueden crear tablas y visualizaciones de todo tipo con estilos únicos aplicando las clases predefinidas directamente en el HTML. Esto facilita en gran medida la personalización de los diseños sin necesidad de añadir más líneas de CSS. Por ejemplo, mediante clases como border, bg-gray-100 o text- center y otras similares, se logra un diseño atractivo y adaptable a las necesidades del proyecto.

2.2.4 Django

En la actualidad, uno de los frameworks backend de alto nivel más destacados a nivel mundial. Django fue desarrollado con el fin de solucionar una gran cantidad de complicaciones del desarrollo web. Fue diseñado para ser sumamente rápido en cuanto al desarrollo, sin descuidar la seguridad de las aplicaciones. Además, Django es sumamente escalable, permitiendo el crecimiento de una manera rápida y flexible (Django, 2024).

Otro punto a favor de este framework es su ORM (Object Relational Mapping), una herramienta diseñada para simplificar significativamente la interacción con bases de datos. El ORM de Django permite manejar los datos como objetos de Python, eliminando la necesidad de escribir SQL. Esto facilita considerablemente el desarrollo y la gestión de consultas, actualizaciones y transacciones,

mejorando la eficiencia y reduciendo los errores.

2.2.5 Fetch

Una de las API más destacadas para la comunicación HTTP, es decir, el elemento que permite comunicar el frontend con el backend. También aporta en la definición de conceptos relacionados con HTTP como los CORS y otras extensiones (*Uso de Fetch - Referencia de La API Web | MDN*, 2024).

2.2.6 Tabla comparativa frameworks frontend

Tabla 1.

Comparación frameworks frontend

Característica	**Next.js**	**Angular**	**Vue.js**
Renderizado híbrido	X		
Integración con React	X		
Arquitectura basada en componentes	X	X	X
Mayor número de herramientas y librerías	X		
Curva de aprendizaje			X
Comunidad	X	X	
Documentación	X	X	
Escalabilidad	X	X	
Total	7	4	2

2.2.7 Tabla comparativa de frameworks backend

Tabla 2.

Comparación de frameworks backend

Características	Django	Express.js	Laravel
Seguridad integrada	**X**		
Escalabilidad	**X**	**X**	**X**
Comunidad	**X**	**X**	**X**
Python (lenguaje popular)	**X**		
Configuración inicial	**X**		
Integración con Node.js		**X**	
Desarrollo rápido de APIs REST	**X**		**X**
Total	**6**	**3**	**3**

2.2.8 Tabla comparativa de librerías CSS

Tabla 3.

Comparación de librerías CSS

Características	Tailwind CSS	Bootstrap
Personalización de componentes	X	
Facilita la creación de estilos	X	X
Rendimiento	X	
Curva de aprendizaje		X
Componentes predefinidos		X
Mayor flexibilidad	X	
Total	4	3

2.3 Base de datos

2.3.1 PostgreSQL

Es un sistema gestor de bases de datos objeto-relacional de código abierto que ha ganado una gran reputación debido a su arquitectura, fiabilidad, integridad de datos, entre otras características que lo hacen un DBMS robusto y confiable (PostgreSQL, 2024). PostgreSQL cuenta con más características, mejor rendimiento de escritura y lectura, soporte de características avanzadas, además de un soporte para transacciones complejas que los hace una de las mejores opciones frente a otros DBMS como MySQL (AWS, 2023).

2.3.2 Django ORM

Un ORM (Object Relational Mapping) es el encargado de mapear las bases de datos a objetos, de esta manera realizar consultas de todo tipo sin comunicarse directamente con la base de datos (Ortega, 2024). Django maneja su propio ORM, simplificando todo tipo de interacción con la base de datos; para muchos usuarios, es una de las características más destacables de este framework (Django Software Foundation, 2024).

2.4 Tecnologías de soporte

2.4.1 Git

Es un sistema de control de versiones, actualmente considerado el estándar mundial (mijacobs et al., 2023). Utiliza un clon local, permitiendo trabajar en estos y realizar cambios sin afectar lo anteriormente hecho. Una vez que el código realizado de manera local sea revisado, se pueden sincronizar los cambios en el repositorio.

2.4.2 GitHub

Es uno de los repositorios más usados para poder almacenar y compartir código con otros usuarios (GitHub, 2024). La capacidad de realizar trabajo colaborativo y su integración con Git son la característica clave frente a otros repositorios.

2.4.3 Postman

Plataforma desarrollada para simplificar el ciclo de vida de las APIs. Cuenta con varias herramientas para el desarrollo de APIs, desde el diseño, las pruebas y la documentación (Postman, 2022). El principal uso de Postman son las pruebas de funcionamiento de las APIs mediante sus endpoints.

2.4.5 Excalidraw

Excalidraw es una herramienta desarrollada para la creación de diagramas de todo tipo. En otras palabras, es una plataforma colaborativa que permite a sus usuarios trabajar de manera simultánea (Excalidraw, 2024), ofreciendo una experiencia sencilla y ágil.

2.4.6 Visual Studio Code

Visual Studio Code es el editor de código más popular del mundo. Es sumamente versátil, permitiendo cualquier lenguaje de programación y framework, además de tener una amplia variedad de extensiones. Esta aplicación cuenta con varias características que la hacen destacable frente a otros editores de código, como terminal integrada, control de versiones integrado con git, soporte para depuración, entre otras (Microsoft, 2021).

2.4.7 PyCharm

PyCharm es un IDE centrado en el desarrollo web y la ciencia de datos, diseñado específicamente para el lenguaje de programación Python (Jetbrains, 2024). La principal ventaja radica en las

herramientas integradas y listas para usar, minimizando posibles problemas de incompatibilidad de versiones.

2.4.8 Notion

Notion es una herramienta multiuso diseñada para la gestión de proyectos y la documentación (Notion, 2024). Además de esto, permite integración con distintas aplicaciones como GitHub, lo que permite que la documentación de proyectos de código sea más sencilla.

2.5 Gestión de inventario y ventas

2.5.1 Gestión de inventario

La gestión de inventario no solo se refiere a la visualización de productos disponibles. El objetivo del manejo de inventario es la confirmación y verificación de existencias dentro de la empresa. La gestión de inventario permite un control sobre el inventario físico mediante los datos anotados en el sistema o registro que se maneje. El manejo de inventario de manera adecuada permite a una empresa saber con exactitud qué productos se tienen, qué cantidad, su valor, entre otros atributos. En otras palabras, permite tener un control de los productos mediante detalles suyos y de sus movimientos (Mena, 2024). Una gestión eficaz resulta en una mejora en la toma de decisiones, optimización de recursos y de procesos y una mejora en la satisfacción del cliente.

Un ejemplo notable es el caso de Walmart, el cual adoptó un sistema sumamente avanzado basándose en la reposición continua mediante datos en tiempo real; con esto no solo logró mejorar la disponibilidad de sus productos, sino que también logró reducir costos y mejorar su eficiencia operativa en general (Facultad de Contaduría y Administración, Universidad Nacional Autónoma de México [UNAM], 2019).

2.5.2 Sistema de inventario perpetuo

Un sistema de inventario perpetuo hace referencia a un sistema que se actualiza de manera continua, registrando dentro del inventario cada entrada y salida de productos. Este método se encarga de recoger y almacenar todos los datos de los productos; posteriormente toda esta información es registrada en la base de datos. Además de esto, el sistema de inventario perpetuo define tres métodos para el manejo de productos: método UEPS (último en entrar, primero en salir), PEPS (primero en entrar, primero en salir) o el de promedio (Transeop, 2021).

2.5.3 Stock

"El stock se define como la mercancía que se almacena en un almacén u otro tipo de local para ser vendida o comercializada". (Galiana, 2024). La gestión de stock debe adaptarse a cada empresa y a las políticas que estas manejen.

2.5.4 Registro de ventas

Es un elemento clave en cualquier empresa, ya que sirve para monitorear los movimientos relacionados con las ventas realizados por la empresa. Esto permite evaluar de manera continua el desempeño del negocio, determinando su rentabilidad (Manzanilla, 2023).

2.5.5 Kardex

Un registro kardex es el nombre con el que se le conoce a un documento o tarjeta que se utiliza para el registro de movimientos de mercadería. El método que se emplea al usar este tipo de registro es conocido como permanencia de inventarios. El kardex permite registrar tanto entradas como salidas de productos y conocer existencias de los artículos (Gómez Ramírez, 2010).

2.6 Seguridad en el desarrollo web

2.6.1 Autenticación

Es un proceso que busca validar la identidad de un usuario para acceder a dispositivos, aplicaciones o distintos servicios. En el desarrollo de software, usualmente se maneja como factor de autenticación un usuario y una contraseña. Una vez se validen estos campos, se permite el ingreso al sistema (Okta, 2024).

2.6.2 Autorización

La autorización es un término estrechamente relacionado con la autenticación; sin embargo, esta se refiere a verificar los recursos, acciones o apartados que un usuario puede acceder dependiendo de su rol o posición (Okta, 2024).

2.6.3 JSON Web Tokens

"Los JSON Web Tokens son un método abierto y estándar de la industria RFC 7519 para representar reclamaciones de forma segura entre dos partes". (auth0.com, 2024). La seguridad de aplicaciones web mediante el uso de JWT se basa en dos pilares fundamentales (Auth0, 2025):

- Autenticación: Se encarga directamente de la autenticación de los distintos usuarios registrados en el sistema (Auth0, 2025).
- Autorización: Determina qué acciones puede acceder cada usuario autenticado (Auth0, 2025).

Un JWT consta de tres partes.

- Header: Este componente contiene el tipo de token y el algoritmo.
- Payload: Contiene una gran cantidad de componentes con información como roles, permisos, fecha de expiración, etc. (Auth0, 2025).

- Signature: Este componente se encarga de garantizar la integridad y la veracidad de un token (Auth0, 2025).

Un sistema de cualquier tipo puede verse beneficiado del uso de JWT; por ejemplo, en un sistema de gestión inventario, pueden registrarse usuarios con distintos roles, como administrador, gerente y usuario. Cada uno de estos permite el acceso a funcionalidades personalizadas para cada rol.

2.7 Diagramas

2.7.1 Mockups

Es una representación visual previa a la programación que muestra el aspecto final del GUI de un sistema (Miro, 2024). Los mockups permiten que los interesados visualicen y aporten retroalimentación respecto al aspecto general del sistema. La creación de mockups se suele realizar después de haber definido la estructura de una manera general, de esta manera teniendo una idea clara del funcionamiento del sistema.

2.7.2 Diagrama de arquitectura de software

Un diagrama de arquitectura es una representación visual de un sistema, en el que se especifican de la manera más precisa los componentes que lo conforman. Además de los componentes, los diagramas de arquitectura deben mostrar de manera clara las comunicaciones e interacciones entre los componentes y los flujos que existirán dentro del sistema. Los diagramas de arquitectura permiten reducir riesgos al desarrollar un sistema, además de mejorar la eficacia y la escalabilidad (AWS, 2023).

2.7.3 Diagrama de flujo de navegación

Es una herramienta que permite visualizar la navegación del entorno, trazando la estructura de una aplicación, aplicación web o sitio web (Miro, 2024). De manera más sencilla, se puede definir

como una herramienta de planificación de la navegación de un sistema.

2.8 Hosting del sistema

2.8.1 Plataformas de hosting

Es un servicio de alojamiento que permite que una aplicación web sea publicada en el internet (Gustavo B., 2018). Existe una gran variedad de hostings, tanto gratuitos como de pago, que permiten el alojamiento de bases de datos, backend y frontend de una aplicación.

2.8.2 Vercel

Es una plataforma que unifica múltiples herramientas para el desarrollo, gestión y despliegue de sitios web. En cuanto a hosting, Vercel proporciona un hosting simple de proyectos mediante su integración con GitHub (Vercel, 2018).

2.8.3 Heroku

Es una plataforma que permite a los desarrolladores crear, monitorear, desplegar y escalar aplicaciones. Heroku facilita los procesos de implementación para los desarrolladores y las distintas empresas que hacen uso de sus servicios en la nube (Heroku, 2015).

2.9 Desarrollo de software y software a medida

Podemos definir el software como un grupo de instrucciones que le indican a un computador qué hacer. Existen tres principales tipos de software: software de sistema, software de programación y software de aplicación. Además, en los últimos años, se ha popularizado un cuarto tipo de software, conocido como software embebido. Sin embargo, en nuestro caso, nos centraremos específicamente en el software de aplicación. De manera sencilla, podemos definirlo como aplicaciones que tienen el objetivo de facilitar a los usuarios realizar distintas tareas (IBM, 2023).

En el desarrollo de software, usualmente se involucran tres roles principales que interactúan entre sí para poder llevar a cabo un proyecto de este tipo. Los programadores, ingenieros de software y desarrolladores de software trabajan de manera conjunta para poder analizar, diseñar, desarrollar e implementar las soluciones tecnológicas (IBM, 2023).

Finalmente, el desarrollo de software a medida se refiere a un desarrollo personalizado que busca solucionar necesidades específicas de una empresa u organización. La finalidad de los programas a medida es optimizar recursos, mediante ahorros de costes, optimizando procesos, permitiendo de esta forma mejorar la eficiencia e incrementar la productividad de una empresa (Yeeply, 2022). "En resumen, el desarrollo de software a medida se basa en diseñar, crear, implementar y mantener software para un conjunto específico de usuarios u organizaciones" (IBM, 2023).

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Requisitos funcionales y no funcionales

3.1.1 Requisitos funcionales

Los requisitos funcionales de la óptica aclaran de mejor manera el objetivo del sistema, mostrando de manera mucho más clara las características fundamentales y las restricciones del sistema. A continuación, se presentan los requisitos funcionales establecidos.

- Registro de usuarios e inicio de sesión
    - Ingreso de datos personales básicos de cada usuario como cédula, nombre, correo, etc.
    - Permite la creación de un usuario y contraseña para el ingreso al sistema.
    - Asignar roles con los permisos necesarios para el acceso a diferentes recursos y funcionalidades del sistema. Estos roles son administrador, gerente y usuario.

    - Permitir el ingreso de usuario y contraseña para poder utilizar el sistema.
    - Redirigir al usuario a la página principal del sistema una vez iniciada sesión.
    - Mostrar mensajes y alertas en caso de que se ingresen credenciales incorrectas.
- Gestión de inventario
    - Permitir el registro de productos nuevos en el inventario.
    - Visualizar todos los productos disponibles, incluyendo información clave.
    - Permitir modificar la información de productos ya existentes en el sistema.
    - Eliminar productos que ya no se encuentren disponibles o sean descontinuados.
    - Permitir el registro automático de los movimientos de productos en el Kardex al realizar operaciones de ingreso o salida.
    - Manejo de mínimos y máximos personalizados por producto.
- Gestión de ventas
    - Registrar las ventas realizadas, incluyendo información de los productos y los clientes.
    - Permitir buscar productos mediante filtros como nombre o categoría.
    - Actualizar de manera automática el stock del inventario cuando una venta sea registrada.
- Generación de reportes gráficos
    - Permitir la creación de gráficos personalizados a partir de los datos del sistema.
    - Ofrecer la elección de distintos tipos de gráficos.
    - Exportar los reportes en formato PDF para facilitar la interpretación.
- Manejo de stock
    - Genera alertas para informar sobre los productos que requieren restablecerse o que,

por el contrario, han alcanzado el stock máximo.

- Mostrar un resumen del stock de productos en la página de gestión de productos.

3.1.2 Requisitos no funcionales

Los requisitos no funcionales no se refieren al comportamiento del sistema como en los requisitos funcionales; se refieren a cómo debe "ser" el sistema. A continuación, se presentan los requisitos no funcionales establecidos.

- Interfaz de usuario
  - Debe ser intuitiva y fácil de usar, permitiendo a los usuarios manejar el sistema sin necesidad de instrucciones.
  - El sistema debe ser responsivo, adaptándose a cualquier tipo de pantalla para una experiencia consistente.
  - La navegación del sistema debe ser sencilla y eficaz.
  - Definir las paletas de colores y tipografía que se alineen con la identidad visual de la óptica.
- Rendimiento
  - El sistema debe soportar múltiples usuarios concurrentes, con un mínimo de 5 usuarios conectados.
  - El tiempo de respuesta a las acciones realizadas en el sistema debe ser inferior a 2 segundos.
  - El sistema debe ser escalable, permitiendo adaptarse al crecimiento de la empresa.
- Fiabilidad
  - El sistema debe encontrarse disponible para los usuarios al menos el 99% del tiempo operativo.

    - Garantizar que la información y datos almacenados sean consistentes y precisos.

- Usabilidad
    - El tiempo de aprendizaje del sistema debe ser como máximo 5 horas.
    - La tasa de errores del sistema debe ser menor al 5% de transacciones, incluyendo las validaciones en tiempo real.
    - Realización de pruebas para evaluar el sistema previo a la implementación del prototipo.

3.2 Arquitectura del sistema

Este sistema utilizará una arquitectura de capas; esto se debe a que, a pesar de incrementar la complejidad del sistema, permite obtener un sistema más escalable y modular. Esto separando las responsabilidades entre las distintas capas. El proyecto se trabajará con tres capas lógicas y tres niveles físicos, ya que cada capa estará en un servidor o hosting independiente. Esto se puede ver representado en la Figura 1.

- Capa de datos
    - En esta arquitectura, la capa de datos hace referencia a la base de datos relacional que contendrá los datos clave de la empresa. En este caso, se utilizará el sistema gestor de base de datos PostgreSQL.
- Capa de lógica
    - En esta capa, hacemos referencias al backend, es decir, donde se implementan las reglas del negocio del sistema. En este caso, esta capa será desarrollada en el entorno Python utilizando el framework Django. Esta capa estará a cargo de procesar las solicitudes y devolver los resultados obtenidos a través de un endpoint de una API REST.

- Capa de interfaz
    - El frontend será implementado con el framework Next.js y todos los lenguajes y herramientas que implican esta tecnología. Esta capa permite a los usuarios la interacción con el sistema mediante una interfaz gráfica desde cualquier navegador web.



Ilustración 1. Arquitecturas de 3 niveles físicos y 3 capas lógicas
Fuente: Elaboración propia

El sistema también seguirá el modelo cliente-servidor donde:

- Cliente
    - El cliente puede acceder a la aplicación mediante una URL sin necesidad de instalar ningún tipo de software adicional.
- Servidor
    - El servidor que aloja la API recibe las solicitudes de las clientes; este servidor se comunica con la base de datos y devuelve los resultados hacia el cliente. En el caso de este sistema web, el servidor estará alojado en un hosting en la nube.

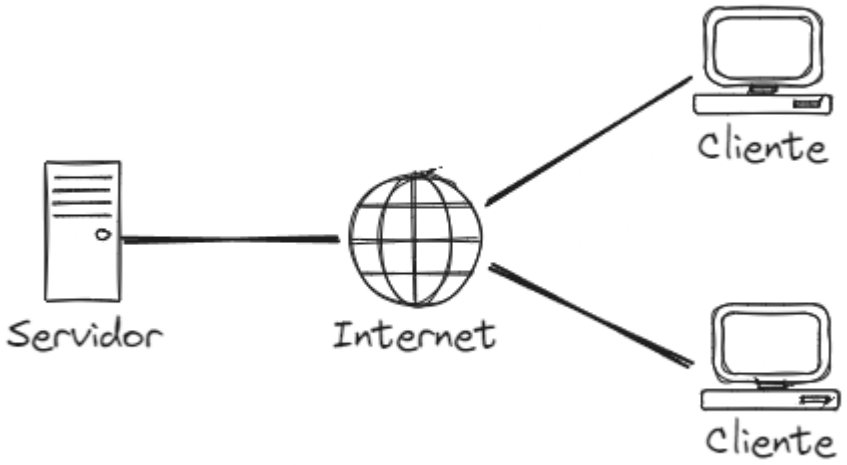


Ilustración 2. Modelo Cliente-Servidor
Fuente: Elaboración propia

Finalmente, la figura 3 representa de mejor manera el flujo de comunicación del sistema para ProdeOptica. En esta figura se observa de mejor manera la arquitectura multicapa y el modelo cliente-servidor, además de elementos extra como la API REST y la base de datos. A continuación, se presentan detalles del flujo de comunicación.

/

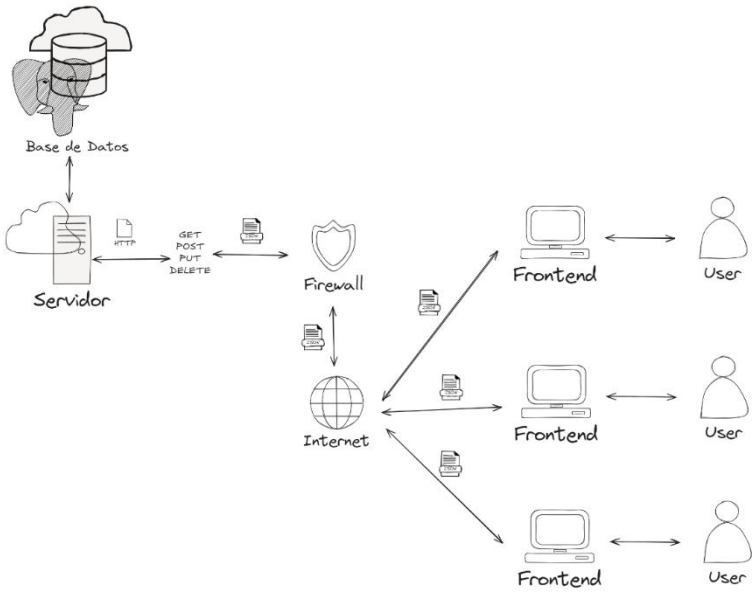


Ilustración 3. Arquitectura del sistema web
Fuente: Elaboración propia

3.2.1 Base de datos – Servidor

Cuando el cliente realice el “pedido” a través de la interfaz, la solicitud llega al servidor a través de la API. El servidor realiza una o varias consultas SQL a la base de datos. La consulta puede ser de lectura y también de escritura (SELECT, INSERT, UPDATE, DELETE). La base de datos responde al servidor con un conjunto de datos, donde la API se encarga de transformar al formato correcto.

Justificación del uso de PostgreSQL

Para el desarrollo de este sistema se eligió PostgreSQL como sistema gestor de bases de datos, principalmente por su capacidad para gestionar transacciones ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad). Esto con el fin de asegurar la integridad y la seguridad de las transacciones que se realicen durante los procesos. Como se puede intuir, estos atributos mencionados previamente son fundamentales para la gestión de inventario y ventas, donde la exactitud de los datos es de suma importancia. A diferencia de los sistemas gestores de bases de datos no relacionales como MongoDB, que tienen como principal característica la flexibilidad del esquema, PostgreSQL brinda un mayor control en sistemas que tengan una alta naturaleza transaccional.

3.2.2 Servidor – Cliente

Cuando la API recibe los datos de la base de datos, se encarga de transformar los datos a un formato distinto que facilite la lectura de los datos; en este caso hablamos de un archivo JSON. La API se conecta con el cliente mediante el endpoint de la misma, respondiendo el archivo JSON que contiene los datos obtenidos de la consulta a la base de datos.

3.2.3 Firewall

Dentro de la arquitectura presentada, el firewall es fundamental para la seguridad del sistema. Su función en la comunicación es inspeccionar y filtrar las peticiones que se realicen. De esta manera, se encarga de prevenir posibles ataques antes de que estos sean realizados; es por esto por lo que es considerado una barrera que evita intrusiones y accesos no permitidos (HP, 2021).

3.2.4 Internet

El internet actúa como la red de comunicación que comunicará a los clientes con los componentes del sistema que se encuentren alojados en un servidor (hosting), esto incluyendo la base de datos (NIC Argentina, 2018).

3.2.5 Cliente

El cliente representa la interacción que tienen los usuarios con el sistema. Es el componente responsable de empezar las solicitudes realizadas al servidor y también de recibir y mostrar las respuestas de manera visual. Dentro de la arquitectura, el cliente tiene tres elementos fundamentales: el frontend, el navegador y el usuario (Mozilla Developer Network, 2024).

3.3 Metodologías y gestión de proyectos

3.3.1 Metodologías ágiles

Una metodología ágil, específicamente dentro del desarrollo de software, tiene como objetivo proporcionar pequeñas funcionalidades de manera periódica con el fin de satisfacer a los clientes. Además, este tipo de metodologías también busca utilizar enfoques flexibles, para de esta manera poder realizar cambios y mejoras de manera constante. Basándose en un ciclo de vida del desarrollo de software, favoreciendo un enfoque mucho más sencillo en la documentación durante el desarrollo y también flexibilidad durante los cambios que puedan suceder durante cualquier

etapa dentro del ciclo del desarrollo (Red Hat, 2024).

Las metodologías ágiles nacen posteriores a los enfoques de desarrollo en cascada. Para esto, priorizaron algunos enfoques y características esenciales durante el desarrollo. Es así que se establecieron las prioridades en un equipo de desarrollo ágil (Red Hat, 2024).

- Prioriza la importancia de las personas y las interacciones del equipo sobre cualquier tipo de proceso o herramienta.
- Es más importante un software funcional que la documentación de este.
- Favorecer la colaboración con el cliente a lo largo de las distintas fases de desarrollo, en lugar de basarse en contratos estrictos.
- Promover totalmente la capacidad de adaptarse a cambios o mejoras durante el desarrollo en lugar de un único plan inicial.

3.3.2 Kanban

Kanban es una de las metodologías ágiles más importantes a nivel mundial, esto ya que permite mejorar el flujo de trabajo mediante un enfoque netamente visual. Para esto, se generan tareas y se visualizan las mismas mediante un tablero, se establece la limitación del trabajo en progreso y también ayuda a la mejora continua basándose en datos. Además, cumple con todas las demás características que definen una metodología ágil, como un enfoque flexible, respeto a los distintos roles y responsabilidades, entre otros. Finalmente, también promueve y facilita la identificación de cuellos de botella, lo que permite reducir los ciclos de desarrollo y mejorar en gran medida la productividad (Kanban University, 2021).

3.3.3 SCRUM

Marco de trabajo que se utiliza para la organización de áreas para llegar a un objetivo en común.

SCRUM define términos como artefactos, roles y eventos asociados que permiten organizar de mejor manera un proyecto. El eje central de Scrum es un equipo organizado por sí mismo y aportar valor a los clientes en ciertos periodos llamados sprint (AWS, 2022).

3.3.4 Principios y valores SCRUM

La metodología SCRUM define varios términos que facilitan la comprensión de su funcionamiento. Dos de estos fundamentos son los principios y valores definidos para el éxito de los proyectos SCRUM. A continuación, se presentarán los principios y los valores de la metodología:

Principios

- Transparencia

  Todos los miembros del equipo conocen los desafíos a los que se enfrentan. Esto con el fin de que los miembros conozcan perfectamente el proyecto, de esta manera evitar cualquier tipo de malentendido (Schwaber & Sutherland, 2020).

- Reflexión

  La reflexión respecto al progreso es importante, ya que permite revisar todos los logros respecto a un proyecto. La información recolectada es de suma importancia, ya que permite mejorar en cuanto a la planificación y estimación de futuros proyectos (Schwaber & Sutherland, 2020).

- Adaptación

  El equipo debe ser flexible a cambios; es común que existan cambios durante el desarrollo, por lo que el orden de las tareas puede cambiar (Schwaber & Sutherland, 2020).

Valores

- Compromiso

Comprometerse al realizar las tareas y los objetivos del proyecto.

- Valentía

Valentía para poder hacer preguntas difíciles y desafiar ideas.

- Concentración

Durante un tiempo específico, el equipo debe concentrarse en completar las tareas más importantes y complejas del proyecto.

- Actitud receptiva

Los miembros del equipo deben mantener durante todo el proyecto una actitud receptiva con el fin de mejorar tanto habilidades propias como el proyecto en general.

- Respeto

Existe un respeto entre todos los miembros del equipo y los directores del proyecto (AWS, 2022).

3.3.5 Artefactos SCRUM

Los artefactos SCRUM son elementos creados con el fin de representar el trabajo o valor. Cada artefacto debe asegurarse de maximizar la transparencia de la información clave. También, cada artefacto cuenta con un enfoque, el cual permite medir el progreso; a estos se los conoce como compromisos (Schwaber & Sutherland, 2020).

- Backlog del producto – Objetivo del producto
- Sprint backlog – Objetivo del sprint
- Incremento – Definición de hecho.

Product backlog

De manera resumida, un product backlog es una lista de tareas, la cual contiene de manera ordenada las necesidades para mejorar el producto. Este artefacto es la única fuente de trabajo que

realiza el equipo de trabajo.

Sprint backlog

De manera similar al producto backlog, el sprint backlog es una lista de tareas; sin embargo, en este caso se refiere a las tareas en tiempo real que están siendo realizadas en el sprint.

Incremento

El incremento es definido como "un paso concreto hacia el Product Goal" (Schwaber & Sutherland, 2020). Cada incremento debe proporcionar valor y para esto el incremento debe cumplir con la definición de hecho. En otras palabras, debe cumplir con los criterios de calidad y usabilidad requeridos (Schwaber & Sutherland, 2020).

3.3.6 Roles SCRUM

En la metodología SCRUM se definen tres roles principales y específicos, los cuales son:

Product Owner

El propietario del producto es el encargado de maximizar el valor del producto. Es el encargado de comprender las necesidades cambiantes de los clientes y el negocio.

Scrum Master

Son los "líderes", y se encargan hasta cierto punto de asegurarse la correcta implementación de la metodología en los equipos. Además, también son los encargados de formar los equipos, los propietarios y el negocio para mejorar los procesos de SCRUM.

Desarrolladores

En este rol, se encuentran todos los profesionales que realizan tareas con el fin de entregar el producto o el incremento. Cada miembro cuenta con las habilidades necesarias para completar las tareas y los trabajos sin depender de otros equipos (AWS, 2022).

3.3.7 Eventos SCRUM

Los eventos SCRUM son diseñados específicamente para poder inspeccionar o adaptar los artefactos. También son útiles para poder "crear regularidad y reducir al mínimo la necesidad de reuniones no definidas en SCRUM" (Schwaber & Sutherland, 2020).

Sprint

Se refiere específicamente al periodo donde los equipos de trabajo realizan actividades o tareas para poder conseguir un incremento. El tiempo que dura un sprint varía dependiendo de las necesidades del proyecto y del equipo.

Planificación de sprint

Los equipos deben estimar su trabajo para un próximo sprint. Además, se deben definir los objetivos, los cuales deben ser específicos, medibles y viables.

Reunión diaria de Scrum

Es una reunión breve en la que todos los miembros del equipo planifican su día. También, se informa sobre las tareas culminadas y se expresan respecto a cualquier desafío que se produzca durante el sprint.

Revisión del sprint

Cuando el sprint termine, se realiza otra reunión informal para poder mostrar y revisar el progreso a las partes interesadas. Durante estas reuniones, el Product Backlog puede ser modificado.

Retrospectiva del sprint

Se documenta y se discute sobre cómo mejorar durante futuros sprints basándose en las experiencias de cada uno (AWS, 2022).

3.3.8 Tabla comparativa de metodologías

Tabla 4.

Comparación de metodologías de desarrollo

Características	**SCRUM**	**Kanban**
Roles	X	
Duración de iteraciones	X	
Límites de trabajo en proceso (WIP)		X
Entrega continua	X	X
Flexibilidad		X
Reuniones	X	
Tableros visuales	X	X
Estructura	X	
Priorización de tareas	X	
Retroalimentación	X	X
Equipos de todo tamaño	X	
Seguimientos de progreso	X	
Total	9	5

3.3.9 Monday.com

Monday es un sistema creado para facilitar el trabajo cooperativo, permitiendo ejecutar proyectos y flujos de trabajo. Es un sistema sumamente flexible y personalizable en cuanto a la creación de sus tableros, los cuales permiten visualizar de manera mucho más sencilla el estado del proyecto. Además, Monday cuenta con una gran variedad de integraciones, entre estas herramientas como Slack o GitHub, ayudando a centralizar los datos y la información dentro de una sola plataforma.

Finalmente, cuenta con plantillas y tableros creados específicamente para el desarrollo de proyectos con SCRUM, lo que facilita la organización con esta metodología (Monday.com, 2024).

3.3.10 Viabilidad de desarrollo individual en un marco ágil

El desarrollo de un proyecto de manera individual en un marco ágil es factible siempre que se apliquen los principios y prácticas de manera adecuada, adaptándolos a la dinámica de un solo desarrollador. La flexibilidad de SCRUM permite personalizar todos los procesos para que una sola persona pueda asumir múltiples roles (Cockburn & Highsmith, 2001).

- Las metodologías ágiles facilitan la planificación, lo que es especialmente útil en el desarrollo de manera individual.
- Existe un control absoluto sobre las decisiones respecto a la planificación.
- Las plataformas de gestión de tareas simplifican la administración incluso para un solo desarrollador.
- Es posible la combinación de roles para una gestión y desarrollo efectivo.

3.3.11 Gestión de múltiples roles y mitigación de conflictos

El desarrollo mediante la metodología SCRUM facilita el desarrollo de proyectos; sin embargo, las definiciones de roles y responsabilidades en esta metodología se centran en grupos de desarrollo y no solo en un desarrollador. Esto puede generar problemas en varios ámbitos, por ejemplo, la superposición de responsabilidades y la definición de prioridades. En este contexto, Monday.com permitió mejorar la gestión y la organización de estrategias al facilitar la estructuración y el seguimiento del proyecto.

Durante el desarrollo, uno de los mayores desafíos es cuando solo una persona asume todos los roles como desarrollador, SCRUM Master, etc. Es por esta razón que también fue necesaria la

creación e implementación de estrategias que ayudan a mitigar la carga asociada a la sobrecarga de responsabilidades.

Monday.com facilita la organización de las tareas a través de tableros personalizables. Esto permite gestionar de manera estructurada y visual las tareas de manera organizada y también definir las prioridades durante los distintos sprints. Para lograr esto, se requiere la definición de objetivos específicos en cada etapa. Esto ayuda a mantener un flujo continuo de desarrollo, evitando la mezcla de responsabilidades entre las distintas etapas de los sprints.

La principal estrategia que se implementa utilizando la plataforma Monday.com e itinerarios definidos fue la separación de roles por tiempo. Esta estrategia implica dividir las jornadas de desarrollo en etapas específicas para cada actividad a realizar. Dedicando un tiempo limitado para actividades técnicas, es decir, codificación, diseño y otras tareas como gestión y documentación asociada a SCRUM, pero optimizados para la gestión individual.

Finalmente, la flexibilidad de las metodologías ágiles permite dar prioridad a tareas esenciales, estructurando los sprints según la metodología elegida, en este caso SCRUM. La fusión de recursos, la organización de las etapas y la priorización de tareas clave permiten reducir los conflictos que surgen del desempeñar varios roles por cuenta propia.

CAPÍTULO IV: DESARROLLO

4.1 Diseño de base de datos y programación de API REST

4.1.1 Introducción y planificación del sprint

Planificación del sprint:

El sprint 1 se centró en su totalidad en el diseño y el desarrollo backend de la aplicación, teniendo como meta principal la programación de las operaciones CRUD para las entidades clave. De esta

manera, se establece una base sólida para el desarrollo frontend y funcionalidades que serán implementadas posteriormente.

- **Duración:** Desde el 4 hasta el 10 de noviembre del 2024
- **Objetivo del sprint:** Diseñar la base de datos y desarrollar la lógica base de la API REST, incluyendo los endpoints de las operaciones CRUD para las entidades obtenidas durante el análisis.
- **Tareas:**
    - Identificar las entidades clave necesarias para el sistema.
    - Definir los atributos de las entidades identificadas.
    - Diseñar la base de datos con la herramienta DbSchema definiendo relaciones, tipos de datos, claves primarias, etc.
    - Generar script SQL desde la herramienta e implementar el modelo en el gestor de base de datos PostgreSQL.
    - Configurar el entorno de programación del framework backend.
    - Implementación de endpoints CRUD para las entidades creadas en la base de datos.

4.1.2 Sprint backlog

Tabla 5.

Sprint backlog sprint 1

Tarea	Descripción	Estado	Tiempo estimado
Análisis del negocio e identificación de entidades	Comprender el negocio e identificar las entidades fundamentales para el desarrollo del sistema.	Completado	4 – 6 horas
Diseño de la base de	Crear el modelo de datos utilizando la	Completado	10 – 12 horas

datos	herramienta DbSchema definiendo todos los componentes clave de la base de datos y generar el script SQL para la implementación en el DBMS.		
Configuración del entorno backend	Configurar el entorno de desarrollo, incluyendo la instalación de paquetes, creación de archivos y carpetas y la conexión a la base de datos	Completado	1 – 2 horas
CRUD de entidades	Implementar los endpoints CRUD de todas las entidades identificadas.	Completado	8 – 12 horas
Pruebas de la API REST	Probar el correcto funcionamiento de los endpoints obtenidos en el desarrollo backend.	Completado	1 – 2 horas

4.1.3 Historias de usuario

Tabla 6.

Historias de usuario sprint 1

ID	Título	Descripción	Criterios de aceptación
HU - 01	Diseño de la base de datos	Como administrador del sistema, quiero almacenar los datos de inventario y ventas, para mantener un registro confiable.	- El modelo debe incluir todas las entidades y atributos necesarios. - La base de datos debe estar implementada y disponible en el DBMS.
HU - 02	Desarrollo de la API REST	Como usuario del sistema, quiero tener la información disponible de manera rápida y	- Los endpoints CRUD implementados para todas las entidades

	en tiempo real, para facilitar mis tareas diarias.	disponibles.

4.1.4 Resultados y retrospectiva del sprint

Tabla 7.

Resultados del sprint 1

ID	Estado	Observaciones
HU - 01	Completado	Las entidades fundamentales fueron identificadas e implementada en el diseño de la base de datos, también se definieron los componentes clave para la lógica del negocio.
HU - 02	Completado	Todos los endpoints CRUD fueron implementados con éxito.

Fortalezas

- El diseño de la base de datos se completó con éxito cumpliendo los requisitos.
- La API REST funciona de manera apropiada para operaciones CRUD básicas.

Debilidades

- Actualmente, solo se manejan las operaciones básicas, por falta de tiempo no se pudieron implementar endpoints personalizados.

Mejoras

- Completar las configuraciones de backend y los endpoints personalizados incluyendo la implementación de autenticación mediante JWT.

Métricas de éxito del sprint

- **Cumplimiento de las tareas**

- **Tareas planeadas:** 5
- **Tareas completadas:** 5
- **Cumplimiento:** 100%

- **Indicadores de calidad**
  - Modelo de base de datos diseñado y verificado.
  - Endpoints CRUD implementados.

4.1.5 Pruebas realizadas

Tabla 8.

Pruebas realizadas sprint 1

ID	Caso de prueba	Resultado	Observaciones
HU – 01	Operaciones básicas con SQL.	Aprobado	Respuesta apropiada a consultas SQ sin errores.
HU - 02	Pruebas CRUD con la herramienta postman.	Aprobado	Endpoints respondiendo de manera correcta con los distintos códigos HTTP.

4.2 Registro de usuarios e inicio de sesión

4.2.1 Introducción y planificación del sprint

Planificación del sprint:

El sprint 2 se centró en el desarrollo de la funcionalidad de registro e inicio de sesión, centrándose tanto en el desarrollo backend (JWT) como en el frontend (formularios). Estas funcionalidades son fundamentales para asegurar la seguridad del sistema mediante la autenticación y la asignación de roles.

- **Duración:** Desde el 11 hasta el 17 de noviembre del 2024
- **Objetivo del sprint:** Desarrollar las funcionalidades esenciales de registro de usuarios e

inicio de sesión que faciliten la creación de cuentas, asignación de roles y el acceso seguro al sistema.

- **Tareas:**
    - Completar la configuración de JWT en el backend para manejar de manera segura las sesiones de los usuarios.
    - Crear el formulario de registro que incluya todos los atributos identificados y necesarios.
    - Crear el formulario de inicio de sesión con los campos básicos de usuario y contraseña.
    - Implementar validaciones de credenciales mediante los endpoints generados en base a los JWT.
    - Redireccionar a las vistas apropiadas basándose en los roles de cada usuario.
    - Implementar alerta de error y de éxito que permitan al usuario sobre los errores cometidos o las acciones exitosas.
    - Validar el funcionamiento apropiado del registro y del inicio de sesión.

4.2.2 Sprint backlog

Tabla 9.

Sprint backlog sprint 2

Tarea	Descripción	Estado	Tiempo estimado
Desarrollo de endpoints para el registro e inicio de sesión.	Desarrollar los endpoints involucrados en el manejo de sesiones mediante el estándar JWT.	Completado	16 – 18 horas
Diseño de los	Diseñar las interfaces	Completado	3 horas

formularios de registro y de inicio de sesión.	involucradas en el registro e inicio de sesión de los usuarios.		
Implementación de la lógica de autenticación.	Implementar la lógica de inicio de sesión mediante los endpoints realizados.	Completado	4 – 6 horas
Redireccionamiento posterior al inicio de sesión.	Configurar la redirección de las páginas a los respectivos dashboards tras el inicio de sisón exitosamente.	Completado	2 horas
Creación y configuración de alertas de errores y de éxito.	Crear alertas que muestren a los usuarios los errores o éxitos al completar los distintos formularios.	Completado	1 hora

4.2.3 Historias de usuario

Tabla 10.

Historias de usuario sprint 2

ID	Título	Descripción	Criterios de aceptación
HU – 03	Registro de usuarios	Como administrador, quiero poder registrar nuevos usuarios en el sistema, para que estos tengan acceso a las funcionalidades del sistema.	- El formulario de registro valida y almacena de manera apropiada los datos personales de los usuarios. - El sistema debe validar los requisitos de seguridad de las contraseñas.

			- El sistema devuelve mensajes mostrando errores o éxitos. - El sistema verificara que las credenciales sean correctas. - Los tokens generados deben estar configurados adecuadamente.
HU – 04	Inicio de sesión de usuarios	Como usuario, quiero poder iniciar sesión en el sistema, para acceder a las funcionalidades del sistema.	
HU - 05	Roles	Como administrador, quiero poder asignar un rol específico a los empleados, para designar funcionalidades específicas.	- Los roles deben ser seleccionables al registrar usuarios. - Las páginas web y los endpoints deben restringir su acceso dependiendo de los roles establecidos.

4.2.4 Resultados y retrospectiva del sprint

Tabla 11.

Resultados sprint 2

ID	Estado	Observaciones
HU – 03	Completado	El sistema permite de manera exitosa el registro de usuarios nuevos incluyendo datos personales, usuario y además de la selección de roles predefinidos para los mismo
HU – 04	Completado	El sistema valida de manera correcta las credenciales de los usuarios permitiendo acceder al sistema.
HU – 05	Completado	Una vez que el usuario es registrado e inicia sesión, tiene

	acceso solo a las páginas para las cuales su rol otorga permisos.

Fortalezas

- Tanto el registro como inicio de sesión funciona de manera correcta validando los datos que se ingresen y mostrando errores o éxito en los formularios.
- El manejo de JWT y contraseñas cifradas, permite ofrecer una alta seguridad en el sistema.

Debilidades

- El manejo de errores es apropiado, sin embargo, es necesario realizar pruebas más exhaustivas para poder manejar todo tipo de excepción.

Mejoras

- Mejorar los mensajes de error para que sean entendibles por todos los usuarios.

Métricas de éxito del sprint

- **Cumplimiento de las tareas**
  - **Tareas planeadas:** 5
  - **Tareas completadas:** 4
  - **Cumplimiento:** 80%
- **Indicadores de calidad**
  - JWT funcionales para registro e inicio de sesión.
  - Las validaciones de usuarios y roles implementadas.

4.2.5 Pruebas realizadas

Tabla 12.

Pruebas realizadas sprint 2

ID	Caso de prueba	Resultado	Observaciones
HU – 03	Registro de usuario exitoso.	Aprobado	La funcionalidad de registro de usuarios nuevo cuando se ingresa datos correctos funciona de manera apropiada.
HU – 03	Registro de usuario con valores erróneos.	Aprobado	El registro de usuarios nuevos con datos erróneos funciona de manera correcta, validando que la información ingresada sea correcta, en caso de no serlo, evita que los datos se almacenen hasta que se corrijan los errores.
HU – 04	Inicio de sesión con datos correctos	Aprobado	La funcionalidad de inicio de sesión funciona de manera correcta al ingresar credenciales correctas.
HU – 04	Inicio de sesión con datos erróneos	Aprobado	El inicio de sesión con credenciales inválidas no es permitido, además de mostrar los mensajes de error o éxito dependiendo de los datos que se ingresen.
HUU – 05	Acceso a funcionalidades en distintos roles.	Aprobado	El inicio de sesión con usuarios asignados en distintos roles redirige a las funcionalidades adecuadas.

4.3 Gestión de inventario 1/2

4.3.1 Introducción y planificación del sprint

El sprint 3 se centra específicamente en el desarrollo de las funcionalidades fundamentales para la gestión de inventario, en la semana 1 del desarrollo de la gestión de inventario, las funcionalidades desarrolladas son específicas de la gestión de productos y parcialmente en la gestión del stock. Funcionalidades avanzadas como la generación de Kardex y el ingreso de pedidos serán

desarrollados durante el siguiente sprint.

Planificación del sprint:

- **Duración:** Desde el 18 hasta el 24 de noviembre de 2024
- **Objetivo del sprint:** Desarrollar las interfases y las funcionalidades de la gestión de productos basándose en la API REST y las operaciones CRUD.
- **Tareas principales**
    - Diseñar la interfaz de gestión de productos que permita ver, crear, editar y eliminar productos nuevos.
    - Implementar los endpoints apropiados para el manejo de solicitudes en todas las tablas involucradas.
    - Implementar la lógica que permita insertar información en las tablas relacionadas.
    - Probar las funcionalidades con datos de prueba que permitan observar su funcionamiento.

4.3.2 Sprint backlog

Tabla 13.

Sprint backlog sprint 3

Tarea	Descripción	Estado	Tiempo estimado
Diseño de interfaz	Diseñar e integrar la interfaz destinada a la gestión de productos.	Completado	8 – 10 horas
CRUD para la gestión de productos	Implementar las funcionalidades CRUD en la interfaz mediante los endpoints desarrollados	Completado	12 - 14 horas

	previamente.		
Gestión de stock básica	Desarrollar junto a la implementación de los endpoints CRUD funcionalidades básicas de la gestión de stock como la actualización de productos manual.	Completado	2 – 4 horas

4.3.3 Historias de usuario

Tabla 14.

Historias de usuario

ID	Título	Descripción	Criterios de aceptación
HU – 06	Gestión de productos	Como usuario, quiero poder ver, agregar, editar, eliminar productos, para poder mantener los productos y el inventario debidamente actualizados.	- Los endpooints responden de manera adecuada. - Los atributos principales de los productos son visibles.
HU – 07	Actualización de stock	Como gerente, quiero poder actualizar de manera manual el stock de productos, para realizar cambios en tiempo real.	- Los cambios realizados en la interfaz deben verse reflejados en la base de datos de manera inmediata. - Permite el cambio manual de para aumentar o reducir

	stock.

4.3.4 Resultados y retrospectiva del sprint

Tabla 15.

Resultados sprint 3

ID	Estado	Observaciones
HU – 06	Completado	La gestión de productos mediante operaciones CRUD es realizada de manera exitosa.
HU – 07	Completado	Las funcionalidades básicas para la gestión de stock no fueron completadas en su totalidad, sin embargo, las funcionalidades fundamentales funcionan de manera correcta.

Fortalezas

- Las funcionalidades CRUD fueron implementadas con éxito.

Debilidades

- El sistema requiere optimización en términos de diseño que faciliten el uso de la interfaz para los usuarios.

Mejoras

- Complementar las funcionalidades relacionadas con la gestión de stock.
- Integrar formas más sencillas e intuitivas para buscar productos y mejorar la experiencia del usuario.

Métricas de éxito del sprint

- **Cumplimiento de las tareas**
    - **Tareas planeadas:** 3
    - **Tareas completadas:** 3
    - **Cumplimiento:** 100%

- **Indicadores de calidad**
    - CRUD para la gestión de productos desarrollado.
    - Actualización manual del stock para todos los productos implementado.
    - Las alertas de stock bajo se configuraron de manera apropiada.

4.3.5 Pruebas realizadas

Tabla 16.

Pruebas sprint 3

ID	Caso de prueba	Resultado	Observaciones
HU – 06	Gestión de productos completada.	Aprobado	Las funcionalidades para poder leer, crear, actualizar y eliminar productos funcionan correctamente.
HU - 07	Actualización de stock.	Aprobado	La actualización de stock de productos específicos funciona de manera apropiada.

4.4 Gestión de inventario 2/2

4.4.1 Introducción y planificación del sprint

En el sprint 4 se terminaron de desarrollar las funcionalidades relacionadas a la gestión automatizada del stock, desarrollando los apartados de ingreso de productos al inventario y la lógica para la posterior implementación de la salida de productos mediante la interfaz y las funcionalidades de ventas.

Planificación del sprint:

- **Duración:** Desde el 25 de noviembre hasta el 1 de diciembre de 2024
- **Objetivo del sprint:** Desarrollar y complementar las funciones relacionadas con la gestión

de inventario y productos.

- **Tareas principales**
    - Crear la interfaz para la gestión de ingresos tanto de pedidos como de diversas razones al inventario.
    - Automatizar los movimientos en el stock realizados mediante las interfaces de ingresos o de ventas.
    - Planificar el funcionamiento y la lógica del apartado de ventas y las salidas de productos.
    - Probar todas las funcionalidades desarrolladas en conjunto con las desarrolladas e implementadas en el anterior sprint.

4.4.2 Sprint backlog

Tabla 17.

Sprint backlog sprint 4

Tarea	Descripción	Estado	Tiempo estimado
Diseño interfaz ingreso a inventario.	Diseñar la interfaz que permita realizar los ingresos de productos existentes en el inventario sin modificar manualmente el stock.	Completado	10 – 12 horas
Implementación de los endpoints en la interfaz.	Implementar las funcionalidades backend dentro de la interfaz.	Completado	2 – 4 horas
Automatización del stock.	Implementar la lógica que permita automatizar el movimiento del stock.	Completado	3 horas

Lógica para las ventas y el kardex.	Desarrollar la lógica utilizada para la reducción de stock mediante la pantalla venta que se desarrollara posteriormente además de la lógica para la generación del Kardex.	Completado	4 – 6 horas
Pruebas generales gestión de inventario.	Probar las funcionalidades implementadas durante el sprint.	Completado	2 horas

4.4.3 Historias de usuario

Tabla 18.

Historias de usuario sprint 4

ID	Título	Descripción	Criterios de aceptación
HU – 08	Ingreso de productos al inventario	Como usuario, quiero poder registrar ingresos de producto al inventario, para actualizarlo de manera precisa.	- El sistema permite el ingreso de productos al inventario sin modificación manual del stock. - El sistema debe manejar errores en los datos ingresados.

4.4.4 Resultados y retrospectiva del sprint

Tabla 19.

Resultados sprint 4

ID	Estado	Observaciones

HU – 08	Completado	El sistema permite el aumento de stock mediante una pestaña de ingreso al inventario. Además, se implementó la lógica en el backend para poder reducir de manera apropiada el stock al desarrollar el módulo de ventas.

Fortalezas

- El sistema cumple con la funcionalidad plantada para el ingreso de productos al stock.
- La interfaz se logró reducir a un solo formulario que permita que los ingresos sean sencillos.

Debilidades

- La funcionalidad de salidas de productos, es decir ventas no se completó.
- La funcionalidad de Kardex no pudo ser desarrollada en su totalidad, esto se debe a la falta de registro de datos de salida.

Mejoras

- Completar la lógica respecto a las salidas de productos e implementar esta funcionalidad con el módulo de ventas.
- Una vez realizado el módulo ventas, realizar tanto la lógica como la vista para la generación del Kardex de cada producto.
- Implementación de mejores validaciones, teniendo en cuenta no solo los datos ingresados, sino también los existentes en la base de datos.

Métricas de éxito del sprint

- **Cumplimiento de las tareas**
    - **Tareas planeadas:** 5
    - **Tareas completadas:** 5
    - **Cumplimiento:** 100%

- **Indicadores de calidad**
    - Automatización del movimiento de stock preparada.
    - La lógica backend de Kardex y ventas desarrollada.

4.4.5 Pruebas realizadas

Tabla 20.

Pruebas sprint 4

ID	Caso de prueba	Resultado	Observaciones
HU – 08	Ingreso de productos	Aprobado	El sistema almacena de manera correcta los ingresos y actualiza el stock reflejando los cambios de manera correcta.
HU - 08	Validación de datos	Aprobado	Al registrar datos incorrectos en el ingreso de productos, la lógica presenta mensajes de error.

4.5 Gestión de ventas

4.5.1 Introducción y planificación del sprint

En el sprint 5 se desarrolló por completo el módulo de gestión de ventas, como se habló durante los requerimientos se decidió que este módulo funcionara como un registro de ventas con datos de importancia tanto de los productos como de los clientes, incluyendo total de productos, precio total, datos del cliente, método de pago, etc.

Planificación del sprint:

- **Duración:** Desde el 2 hasta el 8 de diciembre de 2024
- **Objetivo del sprint:** Complementar la funcionalidad de ventas que permita la reducción

de stock de manera automatizada.

- **Tareas principales**
    - Completar las funcionalidades backend necesaria para la automatización de la reducción de stock.
    - Diseñar la interfaz que permita realizar venta de productos y registrarlas en el sistema.
    - Diseñar la interfaz que permite el ingreso de datos de los clientes relacionados a una venta.
    - Integrar el frontend con los endpoints obtenidos para la automatización en la reducción del stock al realizar ventas.
    - Implementar alertas que notifiquen al usuario cunado de venda un producto con bajo stock.
    - Implementar lógica que elimine temporalmente la visualización de productos sin stock.
    - Completar la lógica del registro Kardex para que funcione con ingresos y salidas de productos.
    - Probar la funcionalidad del módulo completo.

4.5.2 Sprint backlog

Tabla 21.

Sprint backlog sprint 5

Tarea	Descripción	Estado	Tiempo estimado
Finalización funcionalidades	Desarrollar las funcionalidades y endpoint	Completado	3 horas

backend pendientes.	faltantes para el apartado de ventas, incluyendo búsquedas, reducción de stock, etc.		
Diseño de la interfaz para realizar ventas.	Diseñar las dos interfaces involucradas en el registro de las ventas.	Completado	8 – 10 horas
Implementación de la lógica en la interfaz.	Implementar los endpoints desarrollados en la interfaz.	Completado	4 – 6 horas
Desarrollo de la funcionalidad Kardex.	Desarrollar e implementar el módulo que permita generar el registro Kardex de los distintos productos.	Completado	10 – 12 horas
Prueba del módulo de ventas	Probar de manera general el módulo de ventas.	Completado	2 horas

4.5.3 Historias de usuario

Tabla 22.

Historias de usuario sprint 5

ID	Título	Descripción	Criterios de aceptación
HU – 09	Registro de venta (productos).	Como usuario del sistema, quiero registrar las ventas de manera sencilla, para facilitar mi trabajo diario.	- La interfaz permite el registro de productos incluyendo cantidades. - El sistema presenta alerta de productos con bajo stock.
HU – 010	Registro de venta (clientes).	Como usuario del sistema, quiero registrar y guardar los	- La interfaz permite registrar los datos de

		datos de los clientes, para facilitar el proceso de próximas ventas.	los clientes. - La interfaz muestra de manera clara la cantidad de productos y el precio total.
HU - 11	Búsqueda de productos.	Como usuario del sistema, quiero poder encontrar los productos de manera sencilla, para agilizar las ventas.	- La interfaz presenta filtros de búsqueda que facilitan el encontrar productos.
HU - 12	Generación de Kardex.	Como gerente del sistema, quiero poder generar registros Kardex de cada producto, para mantener un control detallado del stock.	- La interfaz Kardex permite cargar información sobre las entradas y salidas de los productos.

4.5.4 Resultados y retrospectiva del sprint

Tabla 23.

Resultado sprint 5

ID	Estado	Observaciones
HU – 09	Completado	El registro de productos al realizar ventas se realiza de manera apropiada mediante un “carrito” que permite agregar cualquier producto dentro del inventario.
HU – 10	Completado	Como siguiente paso, se registra los datos de los clientes de manera correcta, además, carga la información de clientes ya registrados.
HU - 11	Completado	Se estableció un buscador de productos y también un filtro por categorías de productos.
HU - 12	Completado	El sistema registra toda la información de ingresos y salidas y

permite presentar esta información en forma de Kardex.

Fortalezas

- El sistema registra las ventas y toda la información relacionada a la misma de manera exitosa.
- El sistema ya permite la visualización del registro Kardex de cada producto.

Debilidades

- Existió una limitación en la realización de las pruebas, a pesar de que se realizaron las mencionadas, existen otros casos de uso con escenarios distintos que faltaron ser probados.

Mejoras

- Se desarrollaran nuevos elementos o actualizara los existentes en las interfaces para que faciliten el registro de ventas a los usuarios.

Métricas de éxito del sprint

- **Cumplimiento de las tareas**
  - **Tareas planeadas:** 5
  - **Tareas completadas:** 4
  - **Cumplimiento:** 80%
- **Indicadores de calidad**
  - La reducción del stock automatizada al realizar ventas se encuentra totalmente implementada.
  - Las alertas de stock bajo se configuraron de manera apropiada.

4.5.5 Pruebas realizadas

Tabla 24.

Pruebas realizadas sprint 5

ID	Caso de prueba	Resultado	Observaciones
HU – 09	Ingreso de productos regulares.	Aprobado	El ingreso de productos realizado de manera general funciona de manera apropiada.
HU - 09	Ingreso de productos con bajo stock.	Aprobado	Al ingresar productos con un stock bajo, se presenta la alerta advirtiendo esto.
HU - 10	Ingreso de clientes nuevos.	Aprobado	El sistema permite ingresar clientes nuevos como proceso de registro de venta.
HU – 10	Ingreso de CI de clientes disponibles.	Aprobado	El sistema carga los datos de clientes ya registrados de manera correcta.
HU - 11	Búsqueda de productos específicos.	Aprobado	El sistema permite una búsqueda sencilla de los productos.
HU - 12	Visualización Kardex de producto que contiene entradas y salidas.	Aprobado	Se presenta la visualización en forma de registro Kardex de manera correcta permitiendo seleccionar distintos productos.
HU – 12	Visualización Kardex productos sin entradas ni salidas.	Aprobado	Al seleccionar un producto sin registros de entradas o salidas, se presenta un mensaje indicando esto inmediatamente.

4.6 Gestión de reportes gráficos

4.6.1 Introducción y planificación del sprint

El sprint 6 se centra específicamente en generar reportes gráficos que permitan visualizar de manera precisa el estado del inventario y ventas. Para esto de definen ciertos gráficos

predeterminados, y se permite también la creación de gráficos personalizados.

Planificación del sprint:

- **Duración:** Desde el 9 hasta el 22 de diciembre de 2024
- **Objetivo del sprint:** Desarrollar e implementar la funcionalidad de generación de reportes gráficos.
- **Tareas principales**
    - Diseñar la interfaz que permita la visualización de los reportes gráficos.
    - Implementar gráficos por defecto que muestren los datos de importancia general para la empresa.
    - Desarrollar la funcionalidad que permita al gerente personalizar los gráficos con los atributos que el desee.
    - Probar el módulo de generación de gráficos personalizados.

4.6.2 Sprint backlog

Tabla 25.

Sprint backlog sprint 6

Tarea	Descripción	Estado	Tiempo estimado
Diseño de la interfaz de gráficos	Diseñar la interfaz que permita la visualización de los gráficos estadísticos.	Completado	6 – 8 horas
Desarrollo e implementación de la funcionalidad de gráficos personalizados	Desarrollar la funcionalidad tanto en frontend como backend que permita generar gráficos personalizados.	Completado	20 horas
Pruebas del módulo	Validar el correcto	Completado	2 – 4 horas

	funcionamiento de las funcionalidades.	

4.6.3 Historias de usuario

Tabla 26.

Historias de usuario sprint 6

ID	Título	Descripción	Criterios de aceptación
HU – 13	Gráficos predeterminados.	Como gerente, quiero observar de manera gráfica los datos de relevancia para el negocio de manera continua, para poder entender el comportamiento del negocio.	- El sistema muestra gráficos predeterminados de manera automática al ingresar a la interfaz. - Los gráficos deben ser de utilidad para comprender el estado del negocio. - Los gráficos permiten una interacción básica.
HU – 14	Gráficos personalizados.	Como gerente, quiero poder realizar gráficos estadísticos personalizados, para poder analizar información específica según sea necesario.	- El sistema permite seleccionar distintos tipos de gráficos estadísticos. - El sistema permite seleccionar datos personalizados.

4.6.4 Resultados y retrospectiva del sprint

Tabla 27.

Resultados sprint 6

ID	Estado	Observaciones
HU – 13	Completado	El sistema carga de manera predefinida gráficos con información valiosa para la empresa.
HU – 14	Completado	El sistema permite de manera correcta el desarrollar gráficos personalizados.

Fortalezas

- El sistema cumple con los requisitos básicos de visualización y personalización establecidos.

Debilidades

- El apartado de visualización de gráficos funciona, sin embargo, existen limitantes en la generación de gráficos, permitiendo graficar solo atributos de la misma tabla de la base de datos.
- Actualmente, las listas de atributos presentan los nombres de la base de datos, que de manera visual no es tan cómoda para los usuarios.
- Actualmente, el gráfico generado se obtiene en forma de imagen, lo que limita mucho las interacciones con los mismos.

Mejoras

- Mejorar la visualización del gráfico personalizado evitando los problemas referentes a interacción humano computador mencionados en las debilidades.

Métricas de éxito del sprint

- **Cumplimiento de las tareas**

- **Tareas planeadas:** 3
- **Tareas completadas:** 2
- **Cumplimiento:** 66.67%

- **Indicadores de calidad**
  - Gráficos predeterminados totalmente implementados.

4.6.5 Pruebas realizadas

Tabla 28.

Pruebas realizadas sprint 6

ID	Caso de prueba	Resultado	Observaciones
HU – 13	Visualización de gráficos	Aprobado	La visualización de los gráficos personalizados es correcta.
HU - 14	Creación de gráficos personalizados	Aprobado	El sistema permite de manera correcta el desarrollar gráficos personalizados.

4.7 Gestión de stock y optimización de la interfaz de usuario

4.7.1 Introducción y planificación del sprint

El sprint 7 se desarrollara con el único fin de mejorar el sistema en el sentido de alertas y automatizaciones y en la mejora de la interfaz del sistema. Buscando de esta manera perfeccionar la experiencia del usuario al manejar el sistema.

Planificación del sprint:

- **Duración:** Desde el 23 hasta el 6 de diciembre de 2024
- **Objetivo del sprint:** Afinar las alertas y la gestión de stock automatizada para evitar que existan inconsistencias en el sistema y mejorar la experiencia del sistema.

- **Tareas principales**
    - Finalizar el desarrollo de alertas en las pestañas que sean necesarias para el control de stock adecuado.
    - Desarrollar la lógica que permita mostrar solo los productos disponibles dentro de la pestaña de venta.
    - Rediseñar componentes importantes para la interfaz que permitan mejorar la experiencia de los usuarios.
    - Implementar mejoras necesarias en los distintos módulos del sistema.

4.7.2 Sprint backlog

Tabla 29.

Sprint backlog sprint 7

Tarea	Descripción	Estado	Tiempo estimado
Mejoras en la automatización del stock.	Realizar los cambios y mejorar la lógica respecto a la automatización del stock.	Completado	8 – 10 horas
Desarrollo de alertas en los módulos necesarios.	Implementar junto a la anterior tarea las distintas alertas respecto al stock.	Completado	5 horas
Ajustes en el diseño de las interfaces.	Optimizar las interfaces para mejorar la navegación y la estética del sistema.	Completado	8 horas
Pruebas de usabilidad.	Realizar pruebas que permita validar los cambios y mejoras realizados en la interfaz.	Completado	3 horas

4.7.3 Historias de usuario

Tabla 30.

Historias de usuario sprint 7

ID	Título	Descripción	Criterios de aceptación
HU – 15	Alertas de stock	Como usuario, quiero recibir alerta que reflejen los productos con stock bajo, para evitar que existan productos agotados.	
HU – 16	Interfaz intuitiva	Como usuario, quiero poder navegar a través del sistema de manera sencilla, para realizar distintas tareas de manera sencilla.	

4.7.4 Resultados y retrospectiva del sprint

Tabla 31.

Resultados sprint 7

ID	Estado	Observaciones
HU – 15	Completado	El sistema muestra las alertas de stock bajo cuando los productos pasen el límite establecido al registrar los productos.
HU – 16	Completado	La interfaz permite navegar de una forma sencilla a través de todas las pestañas.

Fortalezas

- Las mejoras realizadas permitieron una significativa mejora en la interfaz que la hace más

comprensible y accesible.

Debilidades

- La experiencia humano computador, abarca una gran variedad de conceptos y detalles, lo que significa que significa que siempre existirá un margen de mejora, por lo que es probable que las pruebas realizadas en este sprint no sean suficientes.

Mejoras

- Finalizar de integrar todas las recomendaciones establecidas por los clientes.

Métricas de éxito del sprint

- **Cumplimiento de las tareas**
    - **Tareas planeadas:** 4
    - **Tareas completadas:** 4
    - **Cumplimiento:** 100%
- **Indicadores de calidad**
    - Mejoras en los componentes de IHC en la navegación del sistema.
    - Alertas implementadas y probadas.

4.7.5 Pruebas realizadas

Tabla 32.

Pruebas sprint 7

ID	Caso de prueba	Resultado	Observaciones
HU – 13	Venta de productos con bajo stock.	Aprobado	El sistema valida de manera correcta la existencia de stock de los productos.
HU - 14	Navegación del sistema.	Aprobado	El sistema es sencillo de usar e

	intuitivo en su navegación y funcionalidades.

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

El sistema prototipo desarrollado demostró un impacto positivo al cumplir claramente cumplir con las características planteadas, integrando una interfaz intuitiva, un sistema de almacenamiento apropiado para las necesidades específicas de ProdeOptica y funcionalidades que permiten la automatización, lo que facilita los procesos de la empresa. Estas características fundamentales han permitido mejorar significativamente la gestión tanto de inventario como de ventas, resolviendo con éxito problemáticas previas como la inconsistencia de los datos registrados en el inventario.

La implementación del sistema en relación con el diseño de la interfaz permitió reducir los tiempos de registro de productos y de ventas, así como disminuir los errores asociados a la gestión manual, mejorando en gran medida el flujo de trabajo. Además, la validación de campos asegura que los datos ingresados sean adecuados y consistentes.

En cuanto a la automatización de los procesos, se lograron implementar características como notificaciones automatizadas para productos con bajo stock y generación de reportes gráficos. Estás funcionalidades optimizaron el monitoreo del inventario y la gestión empresarial, lo que ha permitido un análisis más profundo basándose en datos reales.

Al haber utilizado los frameworks de Django y Next.js durante el desarrollo del sistema, aseguramos la creación de un sistema escalable debido a las características modulares de los mismos.

Durante el desarrollo del prototipo, se comprendió la relevancia de una adecuada planificación al

aplicar una metodología ágil como SCRUM, lo cual facilitó de gran manera la entrega de módulos funcionales, permitiendo iterar y mejorar continuamente según las necesidades y pedidos de los clientes. Además, se destacó la importancia de los conceptos de interacción humano-computador al desarrollar una interfaz, buscando priorizar la experiencia del usuario. Esto facilitó los procedimientos y aumentó la gestión de inventarios y ventas. También, se consiguió influir en la experiencia de los clientes, ofreciendo una mayor fiabilidad.

5.2 Recomendaciones

Una vez implementado el sistema, es necesario que se continúe evaluando la experiencia de los distintos usuarios al utilizar el sistema a través de pruebas de usabilidad. De esta manera, se busca garantizar que el sistema se mantenga intuitivo y eficiente al satisfacer las necesidades de la empresa. Además, se recomienda actualizar el sistema cuando sea necesario con nuevas tecnologías o métodos según sea necesario.

Además de los reportes gráficos generados de manera predeterminada y la funcionalidad que permite personalizar gráficos, se sugiere añadir funcionalidades y herramientas más precisas para la generación de reportes periódicos. Esta mejora permitirá fortalecer el análisis del negocio y, por lo tanto, mejorar la toma de decisiones estratégicas fundamentadas en datos.

Al ser un sistema personalizado para la empresa, es de suma importancia manejar un plan de mantenimiento periódico con el fin de asegurar el funcionamiento adecuado del sistema y resolver cualquier tipo de inconveniente que suceda mientras el sistema se encuentre en funcionamiento. De igual manera, es fundamental que la empresa capacite a todos los usuarios con acceso al sistema en cuanto al uso apropiado del sistema.

Por otro lado, es recomendable la implementación de herramientas como Hotjar para poder observar la forma en que los usuarios interactúan con el sistema. Esto facilitaría de gran manera la

recolección de datos para el reconocimiento de patrones de uso y también para descubrir inconvenientes en el flujo de trabajo. Es una herramienta de suma utilidad para mejorar el diseño y la funcionalidad del sistema basándose en datos.

Junto a la anterior recomendación, una buena práctica es realizar evaluaciones de usabilidad trimestrales que permitan recopilar opiniones respecto al diseño y las funcionalidades de los usuarios habituales del sistema. Lo cual asegurará que el sistema siga evolucionando según las necesidades reales del usuario, asegurando la sostenibilidad del sistema a largo plazo.

REFERENCIAS

Auth0. (2025). *Get Started with JSON Web Tokens*. Auth0. https://auth0.com/learn/json-web-tokens

auth0.com. (2024). *JSON Web Tokens - jwt.io*. Jwt.io; Auth0. https://jwt.io/

AWS. (2022). *¿En qué consiste Scrum? - Explicación sobre la metodología Scrum - AWS*. Amazon Web Services, Inc. https://aws.amazon.com/es/what-is/scrum/

AWS. (2023a). *¿Qué son los diagramas de arquitectura? - Explicación de los diagramas de arquitectura de software y sistemas - AWS*. Amazon Web Services, Inc. https://aws.amazon.com/es/what-is/architecture-diagramming/#:~:text=Los%20diagramas%20de%20arquitectura%20del,la%20arquitectura%20de%20un%20sistema.

AWS. (2023b). *Front End frente a back-end: diferencia entre el desarrollo de aplicaciones - AWS*. Amazon Web Services, Inc. https://aws.amazon.com/es/compare/the-difference-between-frontend-and-backend/#:~:text=El%20front%20end%20es%20aquello,permiten%20que%20la%20aplicaci%C3%B3n%20funcione.

AWS. (2023c). *PostgreSQL y MySQL: diferencia entre los sistemas de administración de bases de datos relacionales (RDBMS). AWS.* Amazon Web Services, Inc. https://aws.amazon.com/es/compare/the-difference-between-mysql-vs-postgresql/

B, G. (2018, June 8). *¿Qué es un hosting y cómo funciona?* Tutoriales Hostinger. https://www.hostinger.es/tutoriales/que-es-un-hosting#%C2%BFQue_es_un_hosting

Borovskoy, D. (2025). *LinkedIn*. Linkedin.com. https://www.linkedin.com/pulse/qu%C3%A9-es-react-caracter%C3%ADsticas-y-funcionamiento-2023-denis-borovskoy/

ClickUp. (2024). *ClickUp™ | Una aplicación para sustituirlas a todas*. Clickup.com. https://clickup.com/es-ES

Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer*, *34*(11), 131–133. https://doi.org/10.1109/2.963450

Comparación con otros frameworks — Vue.js. (2015). Vuejs.org. https://es.vuejs.org/v2/guide/comparison

Django. (2024). *Django*. Django Project. https://www.djangoproject.com/

Django Software Foundation. (2024). *Django 1.0 release notes | Django documentation*. Django Project. https://docs.djangoproject.com/en/5.1/releases/1.0/

Excalidraw. (2024). *Excalidraw — Collaborative whiteboarding made easy*. Excalidraw. https://excalidraw.com/

Facultad de Contaduría y Administración, Universidad Nacional Autónoma de México (UNAM). (2019). *Una cadena de suministro increíblemente exitosa: ¿Cómo lo hace Walmart?* https://cedigec.fca.unam.mx/materiales/administracion/5/anexos/1426/Unidad_2/u2_A_Walmart.pdf?utm_source=chatgpt.com

Galiana, J. M. (2024). ¿Qué tipos de stock existen y cómo se clasifican? *Toyota-Forklifts.es*.

https://doi.org/10162901831/1569925426632

GitHub. (2024). *Acerca de GitHub y Git - Documentación de GitHub*. GitHub Docs. https://docs.github.com/es/get-started/start-your-journey/about-github-and-git

Gómez Ramírez, N. M. (2010). Registros kardex. *Repositorio.konradlorenz.edu.co*. https://repositorio.konradlorenz.edu.co/handle/001/2917

Heroku. (2015, May 26). *What is Heroku | Heroku*. Heroku.com. https://www.heroku.com/what

HP. (2021). *¿Qué es un firewall de red y cómo funciona?* Hp.com. https://www.hp.com/mx-es/shop/tech-takes/que-es-un-firewall-de-red-y-como-funciona

Huet, P. (2022, January 21). *Qué es Tailwind CSS y por qué deberías usarlo*. OpenWebinars.net. https://openwebinars.net/blog/que-es-tailwind-css-y-por-que-deberias-usarlo/

IBM. (2023, March 3). *¿Qué es el desarrollo de software? | IBM*. Ibm.com. https://www.ibm.com/es-es/topics/software-development#:~:text=El%20desarrollo%20de%20software%20a%20medida%20consiste%20en%20dise%C3%B1ar%2C%20crear,de%20usuarios%2C%20funciones%20u%20organizaciones.

Jetbrains. (2024). *PyCharm: el IDE de Python para la ciencia de datos y el desarrollo web*. Jetbrains.com; JetBrains. https://www.jetbrains.com/es-es/pycharm/

Kanban University. (2021). *La guía oficial del método Kanban*. https://resources.kanban.university/wp-content/uploads/2021/08/The-Official-Kanban-Guide_Spanish_A4.pdf

LinkedIn. (2025). Linkedin.com. https://www.linkedin.com/pulse/instagram-scales-python-2-billion-daily-users-shrey-batra/

Manzanilla, V. H. (2023, February 14). *Qué es el registro de ventas + 3 ejemplos*. Emprendedor

Growth Model; EGM. https://metodoegm.com/emprendimiento/registro-de-ventas/

McKinsey Global Institute. (2017). A future that works: Automation, employment, and productivity. Executive summary. In *McKinsey & Company*. https://www.mckinsey.com/~/media/McKinsey/Featured%20Insights/Digital%20Disruption/Harnessing%20automation%20for%20a%20future%20that%20works/MGI-A-future-that-works_Executive-summary.pdf

Mena, P. (2024). *Gestión de inventarios*. Ediciones Paraninfo, S.A.

Microsoft. (2021, November 3). *Visual Studio Code*. Visualstudio.com; Microsoft. https://code.visualstudio.com/

mijacobs, LizCasey1, & EdKaim. (2023, October 5). *¿Qué es Git? - Azure DevOps*. Microsoft.com. https://learn.microsoft.com/es-es/devops/develop/git/what-is-git

Miro. (2024a). *Mockups: Qué son y cómo hacerlos para que funcionen | Miro*. Https://Miro.com/. https://miro.com/es/mockup/que-es-mockup/

Miro. (2024b). *Plantilla de diagrama de flujo para sitio web y ejemplo para equipos | Miro*. Https://Miro.com/. https://miro.com/es/plantillas/diagrama-flujo-para-sitio-web/

Monday.com. (2024, July 15). *¿Qué es monday.com?* Support. https://support.monday.com/hc/es/articles/115005310945--Qu%C3%A9-es-monday-com

Mozilla Developer Network. (2024, July 28). *Visión General Cliente-Servidor - Aprende desarrollo web | MDN*. MDN Web Docs. https://developer.mozilla.org/es/docs/Learn/Server-side/First_steps/Client-Server_overview

Netflix TechBlog. (2025, January 7). Netflix TechBlog. https://netflixtechblog.com/

NIC Argentina. (2018). *¿Qué es Internet?* NIC Argentina.

https://nic.ar/es/enterate/novedades/que-es-internet

Notion. (2024). *Tu espacio de trabajo que conecta tus wikis, documentos y proyectos | Notion*. Notion. https://www.notion.so/es

Okta. (2024). *¿Qué es la autenticación frente a la autorización? -Auth0*. Auth0. https://auth0.com/es/intro-to-iam/authentication-vs-authorization

Ortega, Á. (2024). *LinkedIn*. Linkedin.com. https://www.linkedin.com/pulse/qu%C3%A9-es-un-orm-y-c%C3%B3mo-puede-ayudarte-en-tu-proyecto-%C3%A1ngel-ortega-yz11e/

PostgreSQL. (2024). *PostgreSQL: About*. Postgresql.org. https://www.postgresql.org/about/

Postman. (2022). *What is Postman? Postman API Platform*. Postman API Platform. https://www.postman.com/product/what-is-postman/

React. (2015). React.dev. https://es.react.dev/

Red Hat. (2024). *¿Qué es la metodología ágil?* Redhat.com. https://www.redhat.com/es/topics/devops/what-is-agile-methodology

Schwaber, K., & Sutherland, J. (2020). *Scrum Guide*. Scrumguides.org. https://scrumguides.org/scrum-guide.html

Transeop. (2021, April 22). *Sistema de inventario perpetuo y periódico: ¿Qué es? Diferencias*. Transeop.com; Transeop. https://www.transeop.com/blog/sistema-de-inventario-perpetuo-y-periodico-que-es-diferencias/1038/

Uso de Fetch - Referencia de la API Web | MDN. (2024, July 28). MDN Web Docs. https://developer.mozilla.org/es/docs/Web/API/Fetch_API/Using_Fetch

Vercel. (2018). *Vercel: Build and deploy the best web experiences with the Frontend Cloud – Vercel*. Vercel. https://vercel.com/

Vercel. (2024). *Next.js by Vercel - The React Framework*. Nextjs.org. https://nextjs.org/

Vyas, R. (2022). Comparative Analysis on Front-End Frameworks for Web Applications. *International Journal for Research in Applied Science and Engineering Technology*, *10*(7), 298–307. https://doi.org/10.22214/ijraset.2022.45260

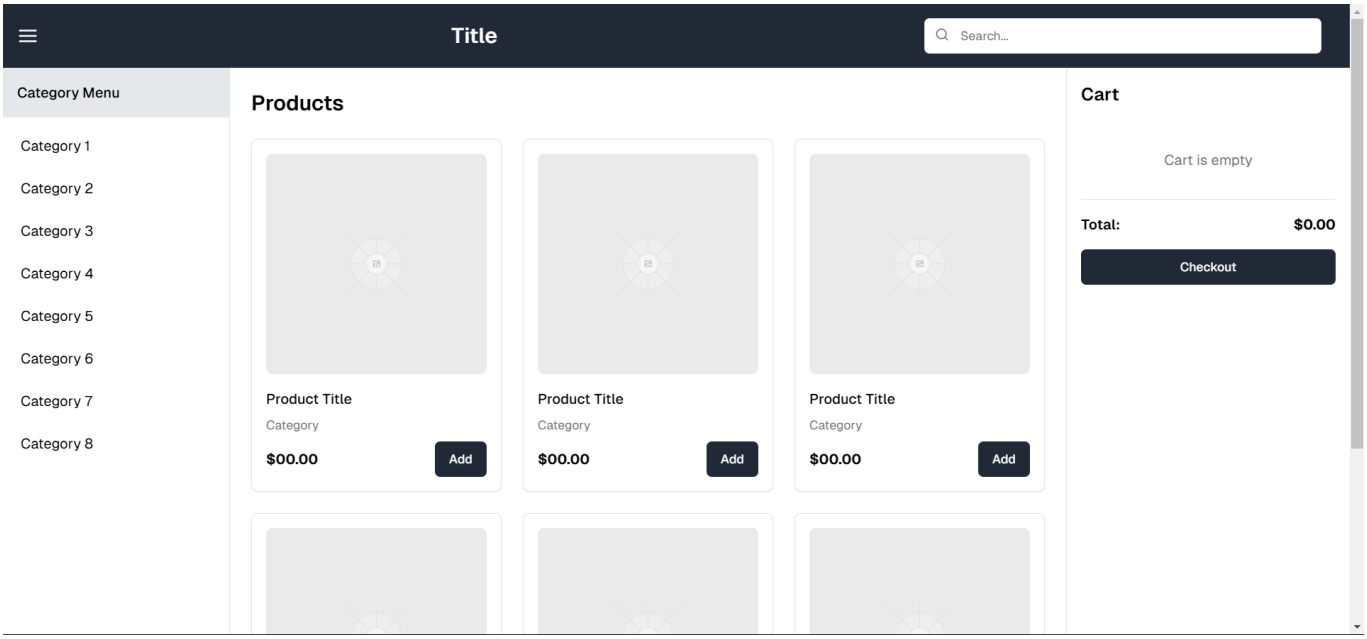
Yeeply. (2022, October 18). *Desarrollo De Software a Medida: ¿Cuándo Y Por Qué Lo Necesita Tu Empresa?* Linkedin.com. https://www.linkedin.com/pulse/desarrollo-de-software-medida-cu%C3%A1ndo-y-por-qu%C3%A9-lo-necesita-tu-empresa-/

Yohan, M., Nitin, S., & Mohammed, S. (2025a). *What are the best practices for secure data validation in back-end code?* Linkedin.com. https://www.linkedin.com/advice/1/what-best-practices-secure-data-validation-back-end-islff?lang=es&originalSubdomain=es

Yohan, M., Nitin, S., & Mohammed, S. (2025b). *What are the best practices for secure data validation in back-end code?* Linkedin.com. https://www.linkedin.com/advice/1/what-best-practices-secure-data-validation-back-end-islff?lang=es&originalSubdomain=es

Zoho Inventory. (2025). *Inventory Plans and Pricing | Zoho Inventory*. Zoho.com. https://www.zoho.com/inventory/pricing/

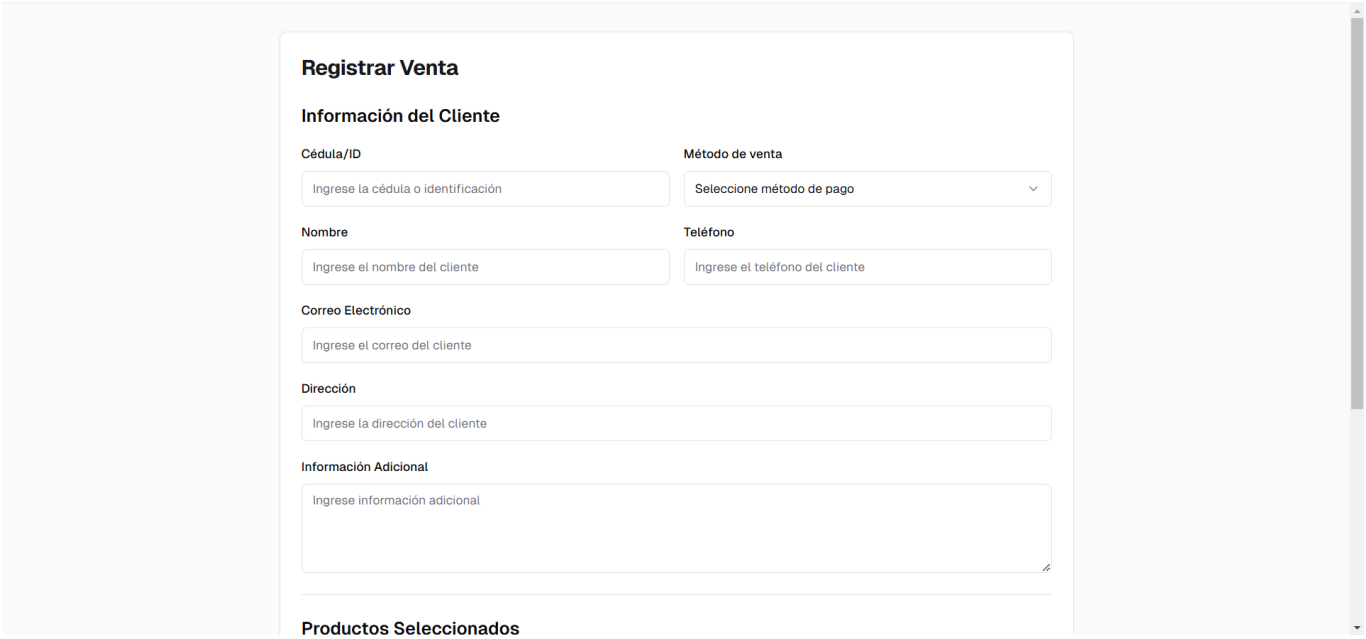
ANEXOS



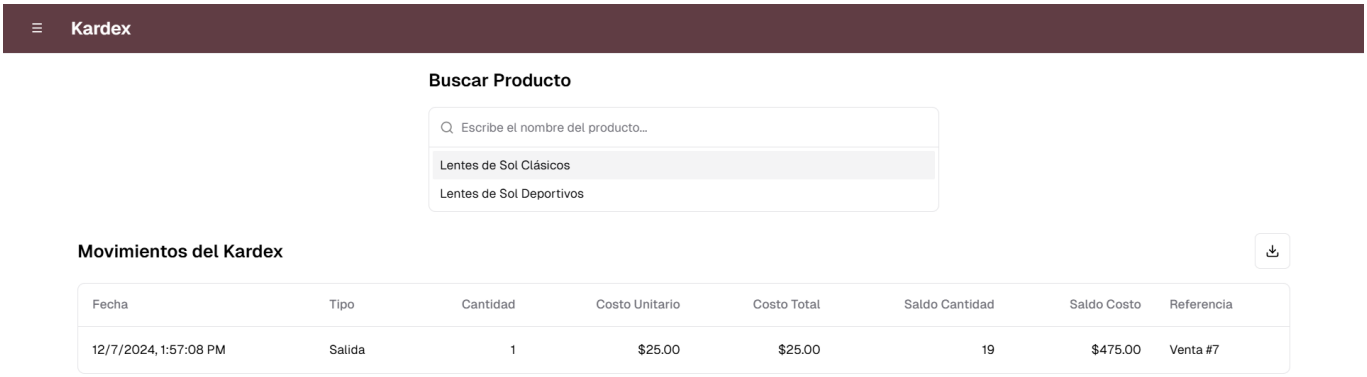
Anexo 1. Mockup pantalla de ventas



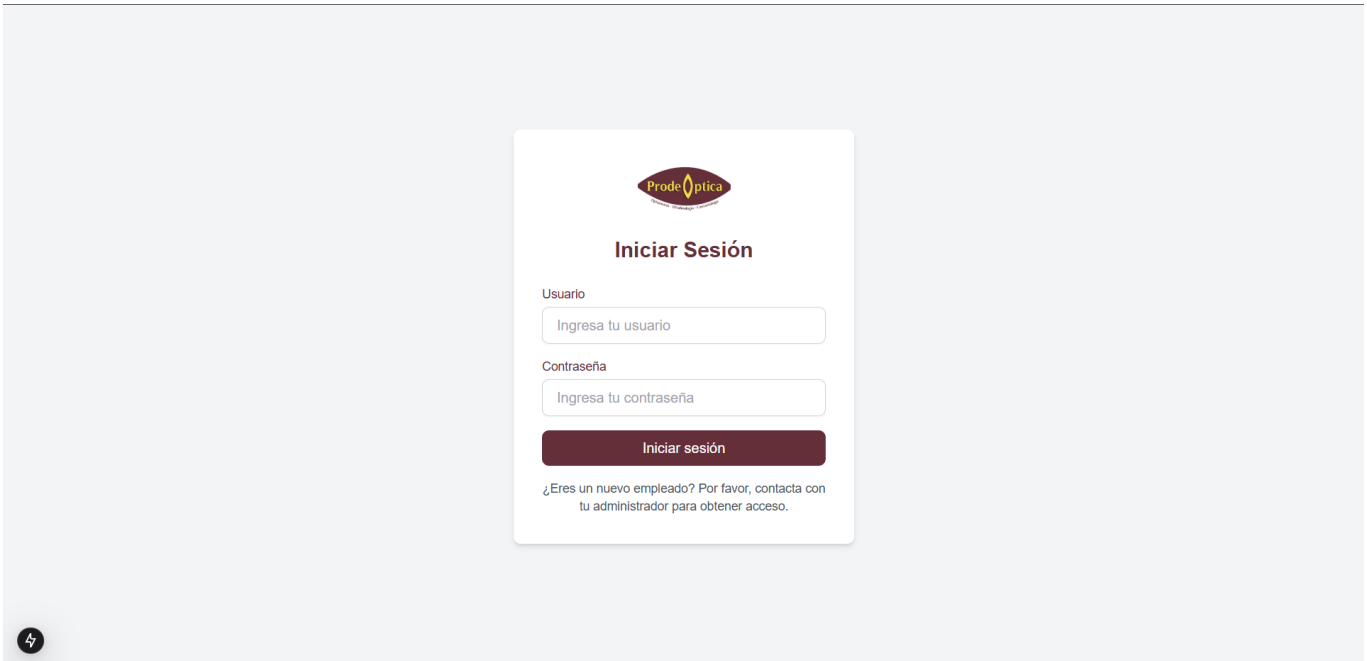
Anexo 2. Mockup pantalla de gestión de productos



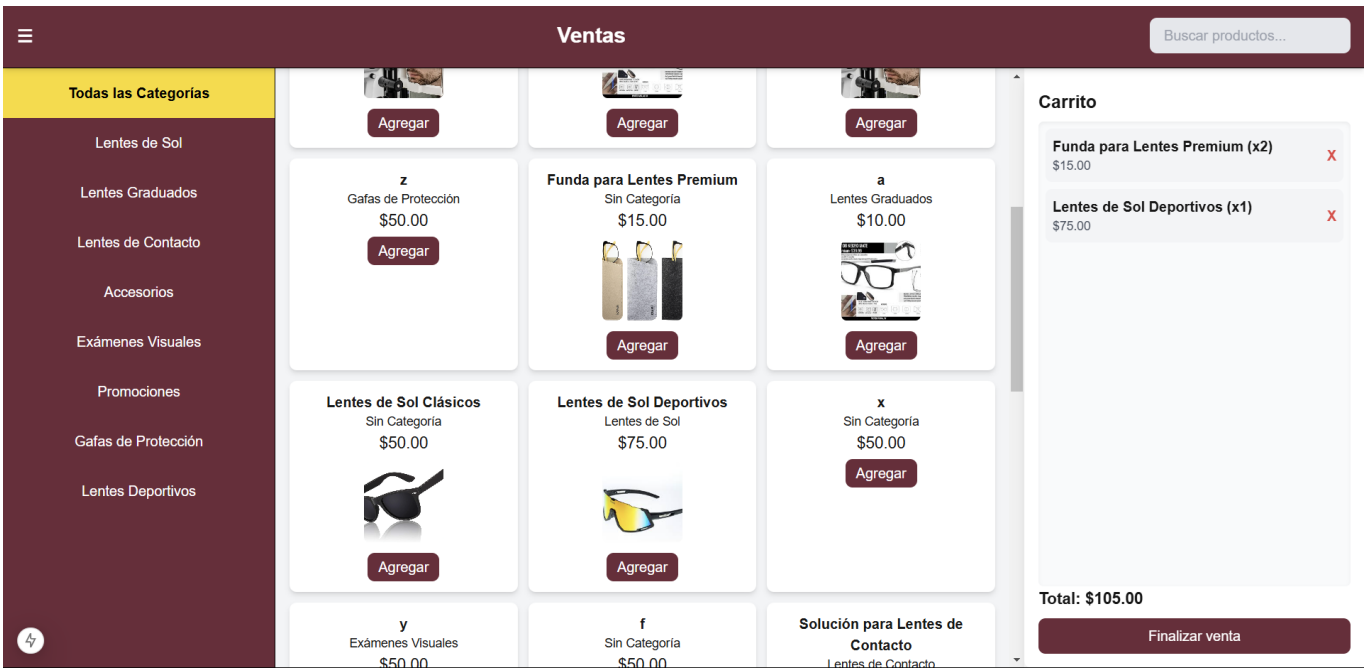
Anexo 3. Mockup de registro de ventas y formularios



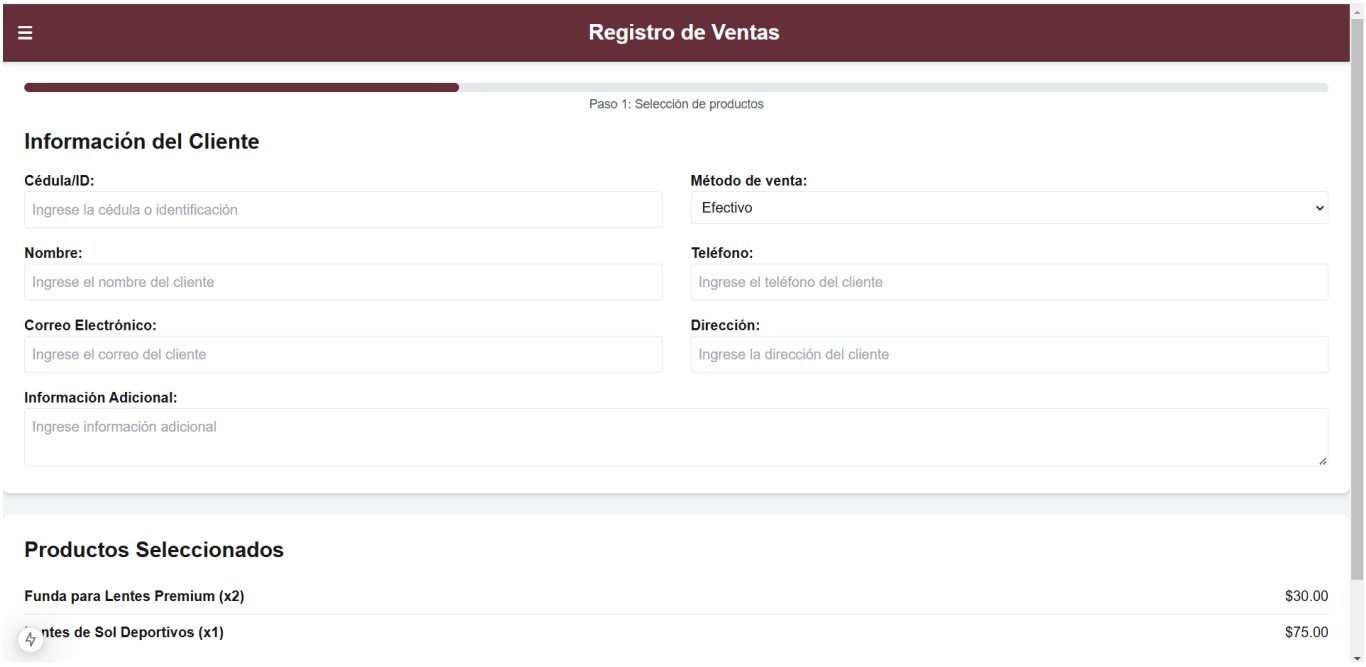
Anexo 4. Mockup registro Kardex



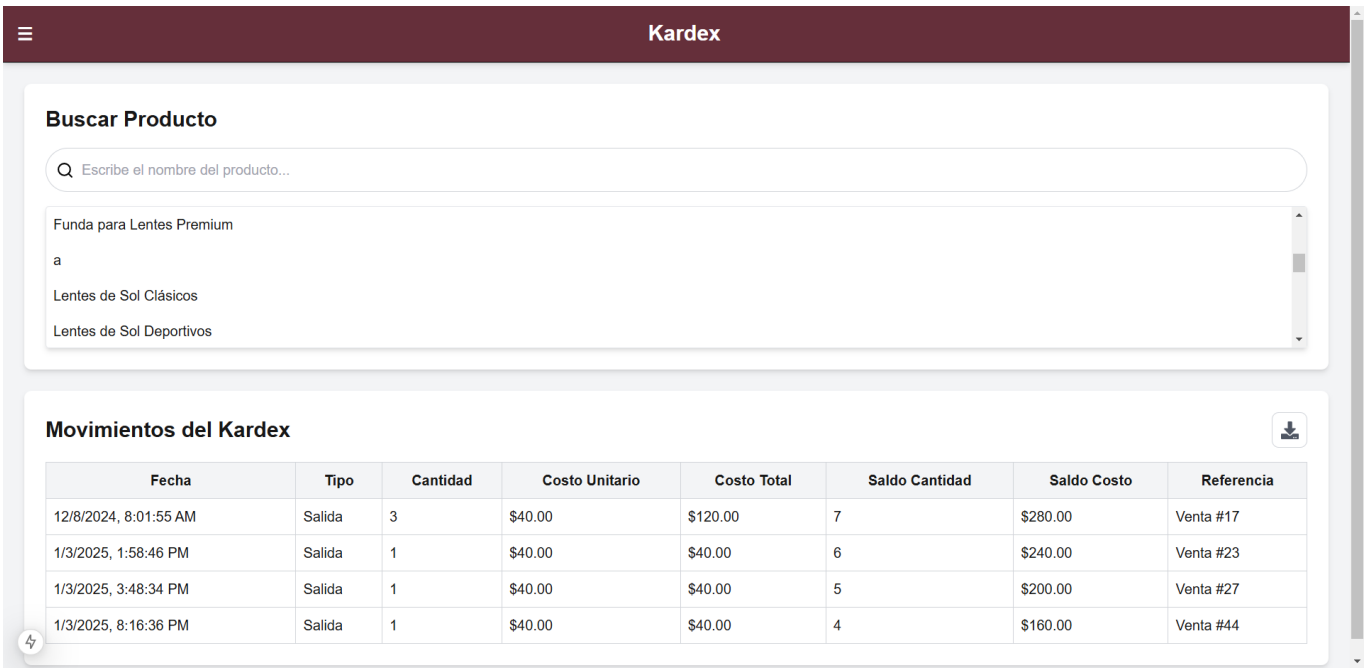
Anexo 5. Pantalla de inicio de sesión



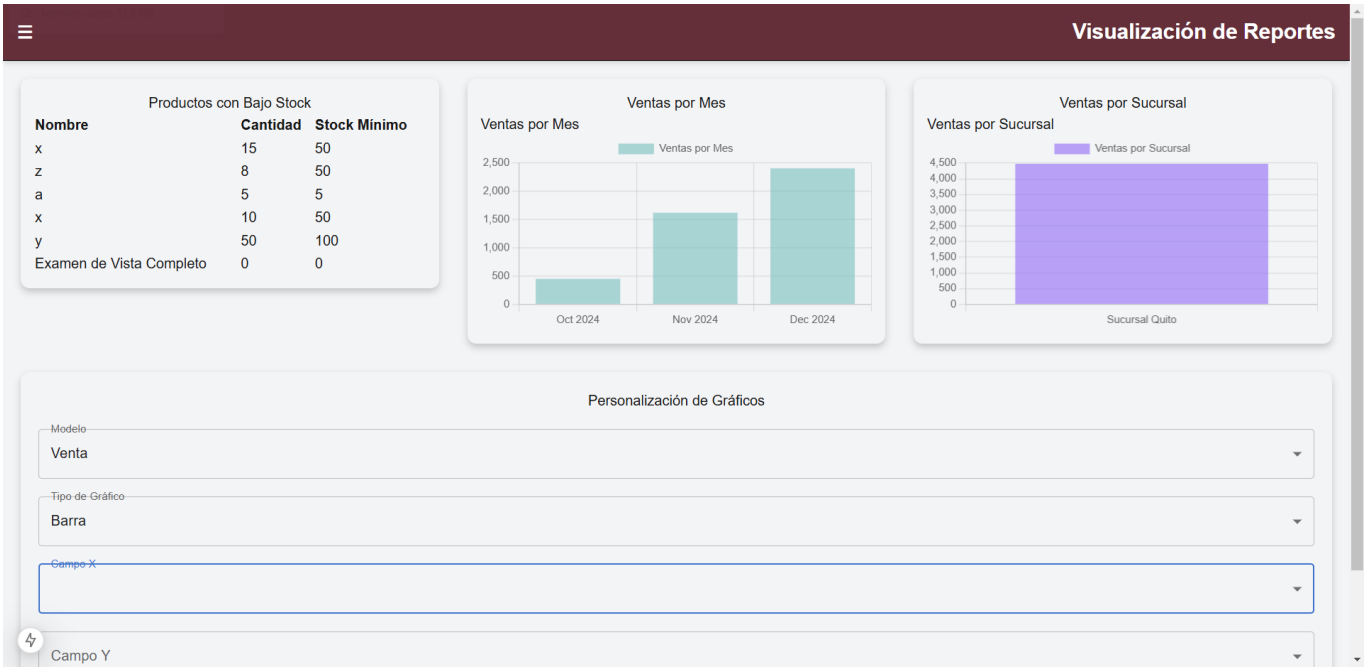
Anexo 6. Pantalla de gestión de ventas



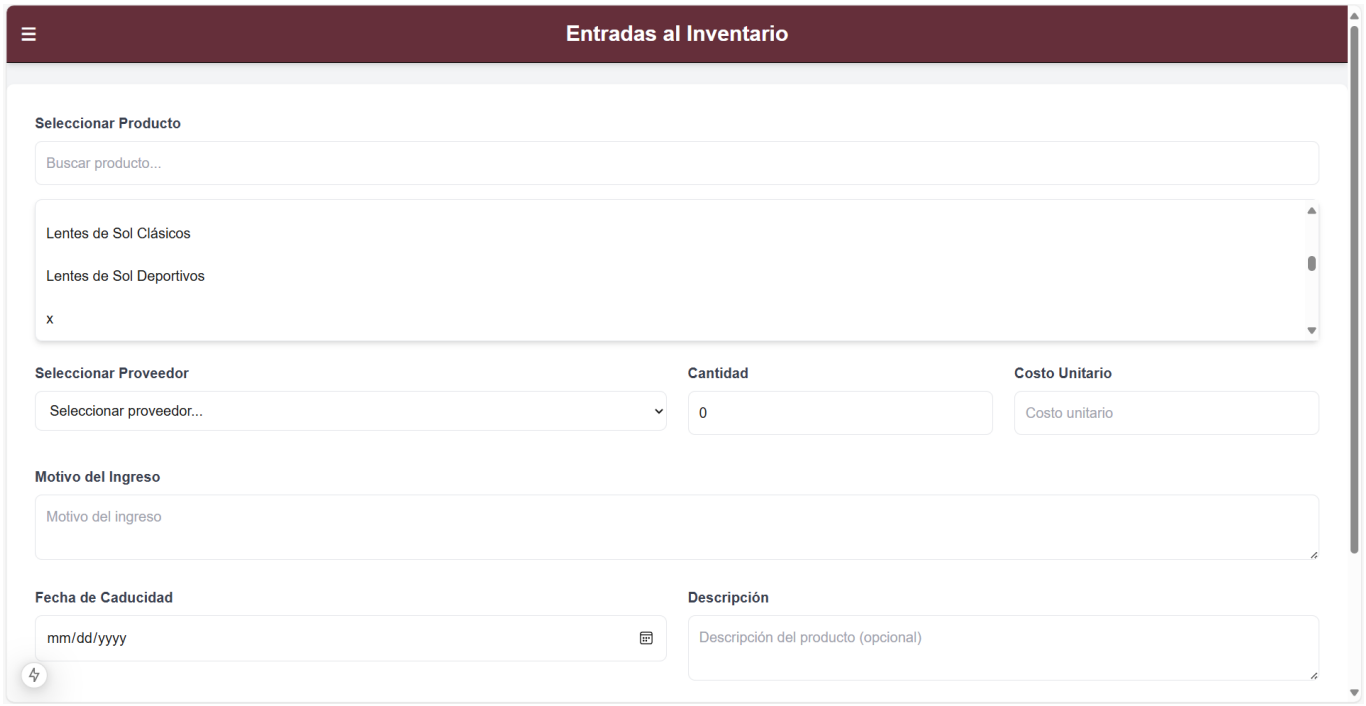
Anexo 7. Pantalla de registro de clientes



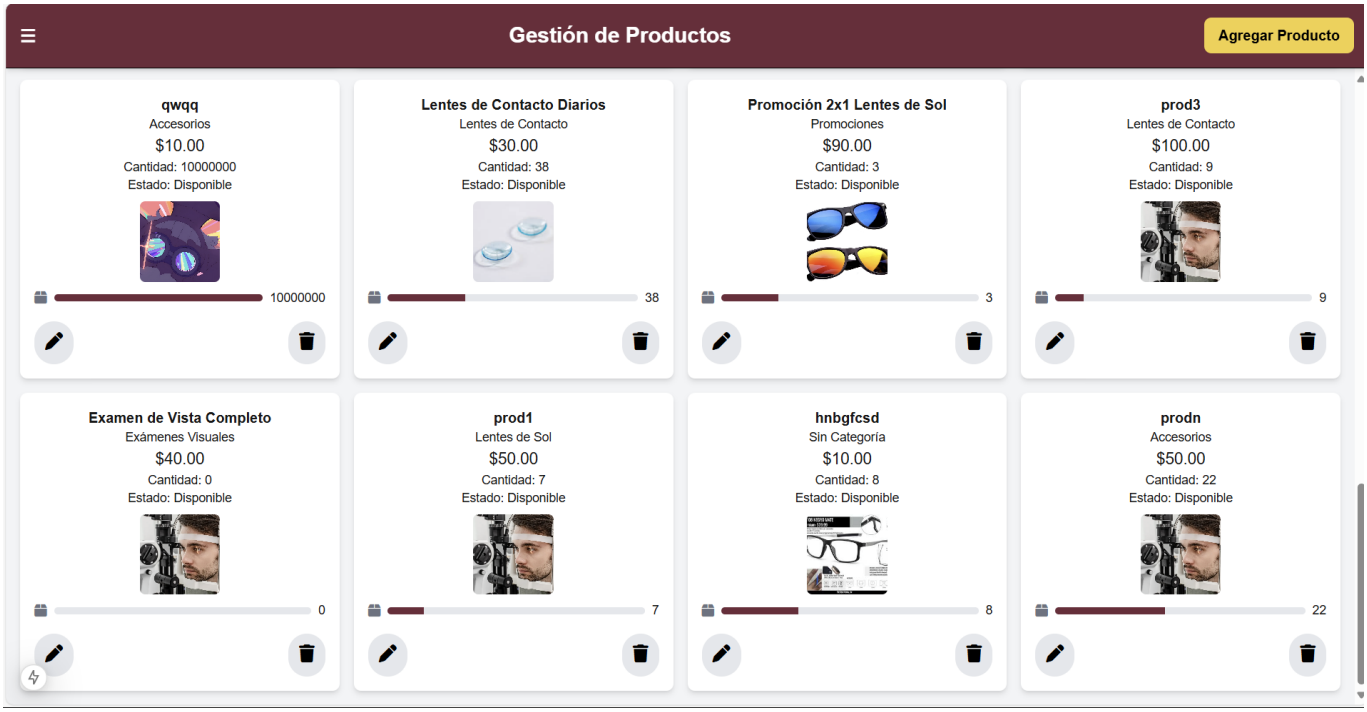
Anexo 8. Pantalla de registros Kardex



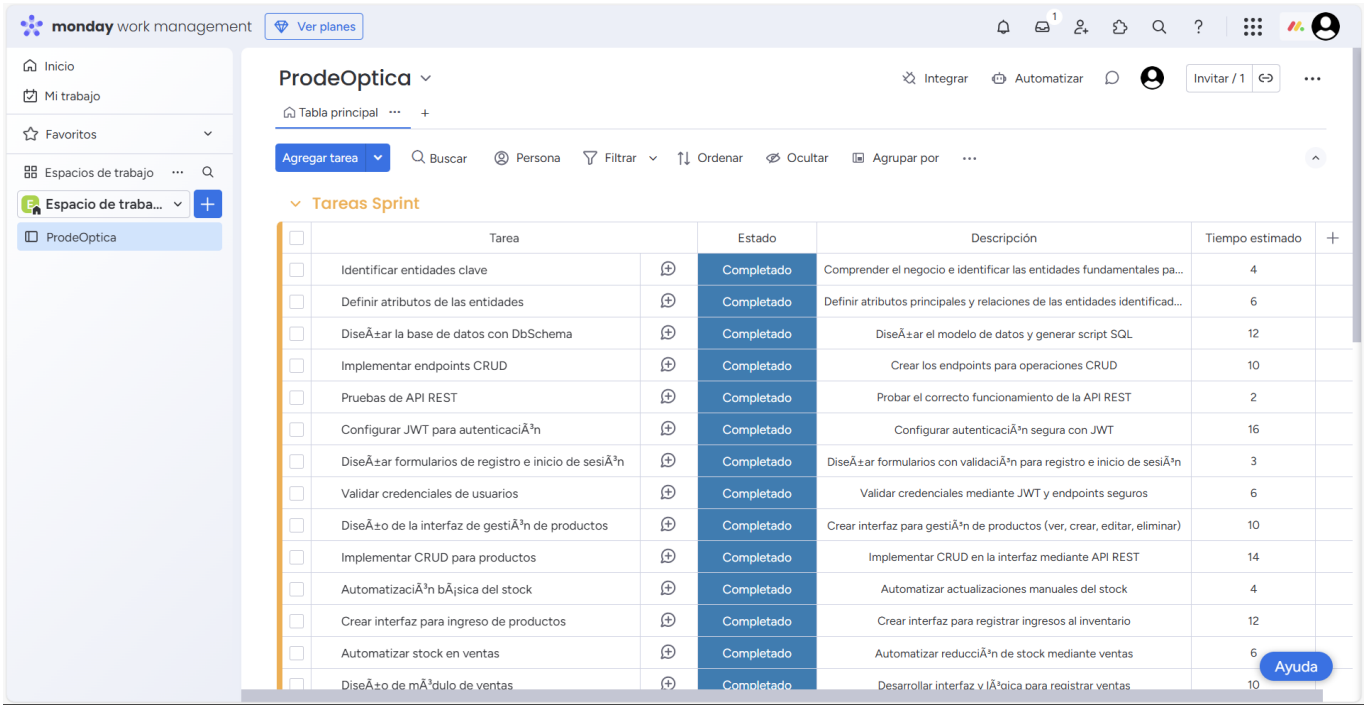
Anexo 9. Pantalla de generación de reportes gráficos



Anexo 10. Ingreso de productos al inventario



Anexo 11. Gestión de productos



Anexo 12. Gestión del proyecto en Monday.com

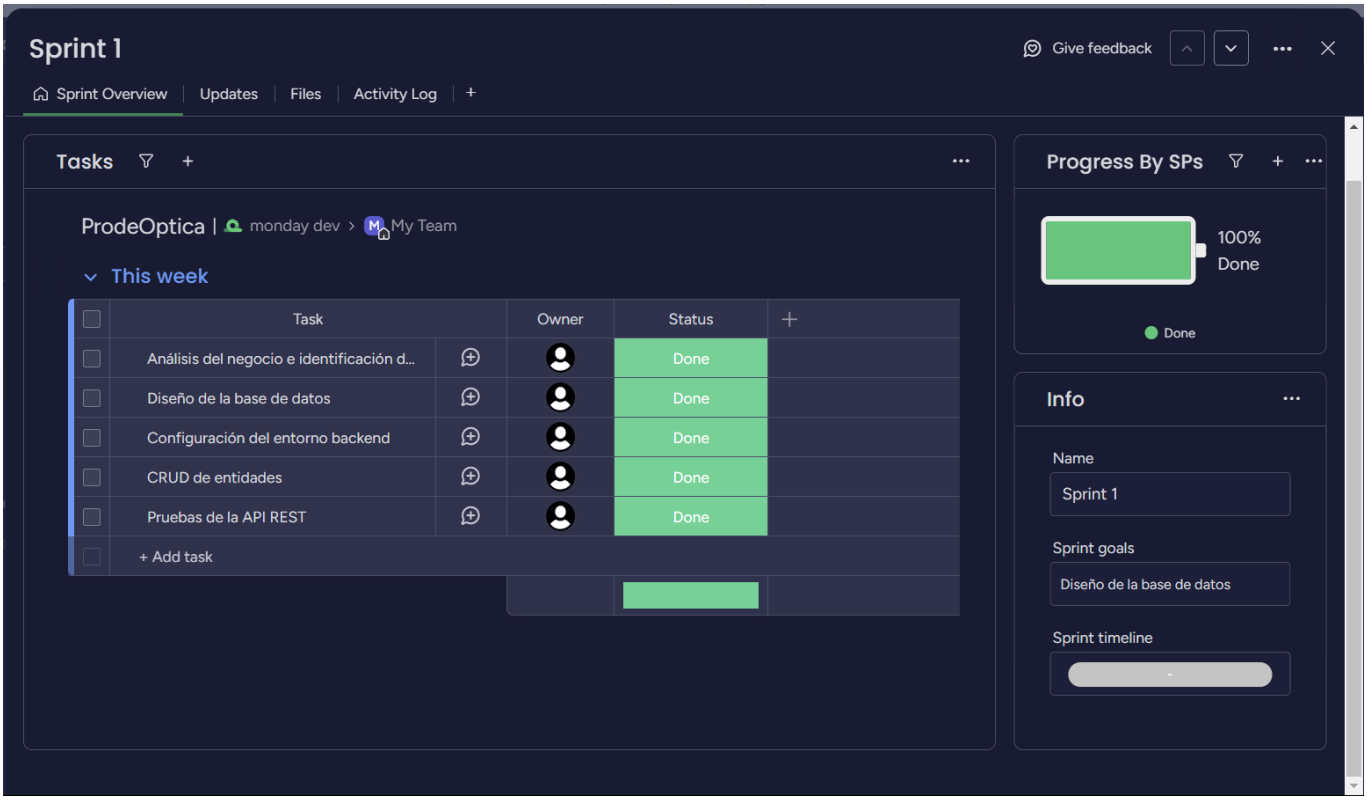


Ilustración 4. Tablero del sprint en Monday.com

```python
class RegisterView(APIView):
    permission_classes = [AllowAny]  # Permite acceso sin autenticación
    def post(self, request):
        print(f"Permisos actuales: {self.permission_classes}")  # Depuración
        serializer = UserSerializer(data = request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data, status=status.HTTP_201_CREATED)
        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)


class LogoutView(APIView):
    permission_classes = [IsAuthenticated]

    def post(self, request):
        print("Headers:", request.headers)
        print("Body:", request.data)

        refresh_token = request.data.get("refresh_token")
        if not refresh_token:
            return Response({"detail": "Refresh token is required"}, status=status.HTTP_400_BAD_REQUEST)

        try:
            token = RefreshToken(refresh_token)
            token.blacklist()
            return Response({"detail": "Logout successful"}, status=status.HTTP_200_OK)
        except Exception as e:
            return Response({"detail": str(e)}, status=status.HTTP_400_BAD_REQUEST)
```

Anexo 13. Implementación de funciones para el manejo de usuarios con JWT



```python
from rest_framework_simplejwt.views import TokenObtainPairView, TokenRefreshView
from django.urls import path
from inventario.API.vistas import RegisterView, LogoutView
from inventario.API.vistas import ProductoListView

urlpatterns = [
    # Login - obtener tokens (access y refresh)
    path('login/', TokenObtainPairView.as_view(), name='token_obtain_pair'),
    # Refrescar token
    path('token/refresh/', TokenRefreshView.as_view(), name='token_refresh'),
    # Logout - invalidar token
    path('logout/', LogoutView.as_view(), name='logout'),
    # Registro
    path('register/',RegisterView.as_view(), name= 'register'),

    path('productos/', ProductoListView.as_view(), name='productos-list'),

]

```

Anexo 14. Endpoints para el manejo del acceso mediante JWT