



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

Unidad Académica de Formación Técnica y Tecnológica – PUCE TEC

**DESARROLLO DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN EFICIENTE
DEL INVENTARIO EN TIENDAS MINORISTAS**

**Proyecto de titulación previo a la obtención del título de: Tecnología En Desarrollo De
Software**

Autor: Steeven Isaias Jiménez Arequipa

Tutor: Andrés Mauricio Chiliquinga Tutachá

Quito, Ecuador

2024

Dedicatoria

Esta tesis está dedicada con todo mi amor, gratitud y respeto a las personas que han sido mi mayor apoyo y fuente de inspiración a lo largo de este arduo pero gratificante camino.

A mis padres, por su amor sin límites, sus sacrificios y su constante apoyo. Agradezco profundamente que me hayan inculcado el valor del esfuerzo y la perseverancia. Sin su respaldo y guía, no habría podido lograr esto. Sus enseñanzas y valores han sido fundamentales en cada paso que he dado, y siempre llevaré con orgullo todo lo que me han inculcado.

A mi hija, por ser mi mayor motivación y por recordarme cada día la importancia de seguir adelante, incluso en los momentos más difíciles. Eres mi mayor orgullo y mi fuente inagotable de energía. Cada sonrisa tuya ha sido un recordatorio de por qué es vital seguir luchando y superarse continuamente. Tu presencia en mi vida me ha dado una nueva perspectiva y me ha impulsado a ser mejor en todo lo que hago.

A todas las personas que me han apoyado constantemente, que me han brindado su ayuda su cariño y sus palabras de ánimo. Cada consejo, cada gesto de apoyo y cada palabra de aliento han sido esenciales para llegar hasta aquí.

Y especialmente, a mí mismo. A veces, el viaje más difícil es el que emprendemos hacia nuestro propio interior. Esta tesis es un testimonio de mi orgullo y superación. Por cada noche en vela, por cada reto enfrentado, por cada momento de duda superado, me dedico a mí mismo este logro. Gracias por creer en mí cuando nadie más lo hizo, por enfrentar y superar cada obstáculo y por nunca rendirme. Este esfuerzo y dedicación personal me han llevado a comprender que el verdadero éxito radica en la capacidad de uno para persistir y creer en su potencial.

Tabla de contenidos

Dedicatoria	2
Lista de tablas	6
Lista de Imágenes	8
Agradecimientos	10
Introducción	11
Antecedentes	12
Planteamiento del Problema	12
Justificación	12
Objetivo General.....	13
Objetivos Específicos.....	13
Metodología de Desarrollo de Software Scrum.....	13
Capítulo I	14
Procesos Actuales.....	14
Infraestructura Actual.....	15
Metodología Scrum.....	16
Etapas de Scrum	16
Sprint (Tablas).....	18
Capítulo II.....	19
Análisis y diseño	19

	4
Mapa de navegación.....	19
Viabilidad técnica.....	20
Viabilidad económica.....	23
Casos de uso.....	24
Capítulo III.....	26
Desarrollo.....	26
Diagrama de clases.....	33
Diagrama de secuencia.....	34
Construcción.....	39
MainActivity:	39
DashboardActivity	40
WebService:	42
Modelo BDD.....	43
Diccionario de datos.....	44
Arquitectura.....	49
Capítulo IV.....	51
Registro de Iteración	51
Pruebas funcionales.....	56
Caso de Prueba 1: Registro de Producto	56
Caso de Prueba 2: Actualización de Inventario	56

Caso de Prueba 3: Generación de Informes	56
Caso de Prueba 1: Verificación de Autenticación.....	57
Caso de Prueba 2: Validación de Entradas en el Webservice	57
Pruebas no funcionales.....	58
Prueba de rendimiento:.....	58
Pruebas de usabilidad:.....	60
Conclusiones	61
Recomendaciones	62
Referencias bibliográficas.....	63

Lista de tablas

Tabla 1. Infraestructura Actual -----	15
Tabla 2. Actividades Sprint I. -----	18
Tabla 3. Análisis de Viabilidad Técnica. -----	21
Tabla 4. Análisis de Viabilidad Técnica. -----	23
Tabla 5. Gastos en hardware. -----	23
Tabla 6. Gastos Recursos Humanos y Recursos Complementarios. -----	23
Tabla 7. Reporte de Gastos.-----	24
Tabla 8. Casos de Uso. -----	25
Tabla 9. Programación de tareas. -----	27
Tabla 10. Descripción y contenido de las tablas. -----	45
Tabla 11. Usuarios -----	46
Tabla 12. Productos. -----	46
Tabla 13. Movimientos. -----	47
Tabla 14. Ventas. -----	48
Tabla 15. Detalles ventas. -----	48
Tabla 16. Reportes.-----	49
Tabla 17. Iteración 1 para Registro de Usuarios. -----	52
Tabla 18. Iteración 2 para Registro de Usuarios. -----	52
Tabla 19. Iteración 3 para Registro de Usuarios. -----	53
Tabla 20. Iteración 1 para Inicio de Sesión. -----	53
Tabla 21. Iteración 1 para Generación de Informes. -----	54
Tabla 22. Iteración 1 para Actualización de Inventario. -----	54

Tabla 23 Iteración 1 para Registro de Productos -----55

Lista de Imágenes

Imagen 1. Procesos Actuales	14
Imagen 2. Mapa de navegación.	20
Imagen 3. Casos de Uso Administrador.	24
Imagen 4. Diagrama de Clases.....	33
Imagen 5. Modulo Autenticación.	34
Imagen 6. Modulo Inventario.	35
Imagen 7. Modulo Salida del Producto.....	36
Imagen 8. Modulo Entrada del Producto.	37
Imagen 9. Modulo Informes.	38
Imagen 10. MainActivity.	40
Imagen 11. DashboardActivity.	41
Imagen 12 WebService	43
Imagen 13. Modelo BDD.....	44
Imagen 14. Arquitectura.	50

DECLARACIÓN y AUTORIZACIÓN

Yo, **STEEVEN ISAIAS JIMENEZ AREQUIPA** con C.I. 1751237429 autor(a) del trabajo de titulación intitulado: “**DESARROLLO DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN EFICIENTE DEL INVENTARIO EN TIENDAS MINORISTAS**”, previa a la obtención del título de **Tecnología En Desarrollo De Software** en la Unidad Académica de Formación Técnica y Tecnológica PUCE TEC:

- 1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.
- 2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE el referido trabajo de titulación, respetando las políticas de propiedad intelectual de Universidad.

Quito, 31 de julio del 2024

STEEVEN ISAIAS JIMENEZ AREQUIPA

C.I. 1751237429

Agradecimientos

Primero que nada, agradezco a Dios por otorgarme la fuerza y la dirección necesarias para cumplir mis objetivos. Sin Su luz y bendiciones, nada de esto habría sido posible.

A mis padres, les debo todo lo que soy y lo que he logrado. Su amor incondicional, su apoyo constante y sus sabias palabras han sido la base principal de mi vida. Gracias por creer siempre en mí.

A mi hija, que es la luz de mis ojos y mi mayor inspiración. Cada paso que doy es por ti y para ti. Tu amor y tu alegría son mi motivación diaria.

Gracias a mis amigos por estar a mi lado tanto en momentos buenos como malos. Su apoyo, su cariño y su lealtad son invaluable. Gracias por ser mi refugio y mi fuerza.

A mis profesores, les agradezco profundamente por compartir su conocimiento y sabiduría. Gracias por sus enseñanzas

Introducción

La eficiente gestión del inventario es un gran desafío para las pequeñas tiendas minoristas, y la necesidad de abordar este problema se ha vuelto crucial en el dinámico entorno empresarial actual. Este estudio se enfoca en la creación de una aplicación para dispositivos móviles especialmente diseñada para la mejora de gestión del inventario de la tienda Víveres y Licores 24 Horas. La importancia de este estudio radica en la capacidad de la aplicación para mejorar los procesos operativos, ayudando a tomar decisiones y el aumento de la rentabilidad de esta tienda minorista.

El propósito principal de este proyecto es proporcionar a Víveres y Licores 24 Horas de un tipo de herramienta tecnológica que permita realizar un seguimiento exacto de los productos, gestionar eficientemente los niveles de inventario y mejorar el reabastecimiento. Para alcanzar este objetivo, la investigación empleará una metodología mixta que combina análisis desarrollo de la aplicación y evaluación práctica en el entorno real de Víveres y Licores 24 Horas.

Se espera que los resultados de esta investigación no solo beneficien directamente a la tienda en cuestión, mejorando su funcionamiento interno, sino que también contribuyan al conocimiento en el ámbito de la gestión de inventarios en pequeñas tiendas minoristas.

Las palabras clave son: gestión de inventario, aplicación móvil, eficiencia operativa, toma de decisiones, pequeñas tiendas minoristas, y rentabilidad.

Antecedentes

La tienda Víveres y Licores 24 Horas actualmente enfrenta problemas significativos debido a la falta de un sistema tecnológico robusto para la gestión de inventario. Esta deficiencia se traduce en imprecisiones en el seguimiento de productos, falta de conocimiento sobre los niveles de inventario en tiempo real y la carencia de sistemas automatizados para la reposición de stock. Estas limitaciones han generado pérdidas económicas, estancamiento operativo y dificultades para satisfacer la demanda del mercado.

Planteamiento del Problema

El problema principal radica en la incapacidad de la tienda para gestionar eficientemente su inventario, lo cual afecta negativamente la toma de decisiones estratégicas y la rentabilidad del negocio. Sin una herramienta adecuada, es difícil identificar patrones de compra, anticiparse a cambios en la demanda y optimizar la rotación de productos. Surge entonces la necesidad de diseñar e implementar una aplicación móvil que responda a estas necesidades específicas y ofrezca una solución integral.

Justificación

La implementación de una aplicación móvil específica para la gestión del inventario permitirá a Víveres y Licores 24 Horas mejorar significativamente sus operaciones. La herramienta facilitará un seguimiento preciso de los productos, optimizará los niveles de stock y automatizará la reposición de inventario. Estos beneficios no se limitan a mejorar el rendimiento operativo y el proceso de tomar una decisión, a la vez que también aumentarán la rentabilidad del negocio y contribuirán al conocimiento general sobre la gestión de inventarios en pequeñas tiendas minoristas.

Objetivo General

Crear una aplicación móvil para la gestión efectiva del inventario en la tienda Víveres y Licores 24 Horas, mejorando la precisión en el seguimiento de productos, optimizando los niveles de inventario y ofreciendo herramientas analíticas que apoyen la toma de decisiones estratégicas.

Objetivos Específicos

Identificar las carencias específicas en la gestión de inventario de Víveres y Licores 24 Horas utilizando técnicas de recolección de datos que permitan entender a fondo las limitaciones actuales.

Crear una pantalla de usuario amigable y comprensible que permita a aquellos empleados de dicha tienda utilizar la aplicación sin requerir una capacitación extensa, facilitando así la transición a un nuevo sistema.

Realizar una evaluación de la eficacia de la aplicación, mediante pruebas piloto del aplicativo para identificar posibles problemas y asegurar una respuesta inmediata a cualquier inconveniente.

Metodología de Desarrollo de Software Scrum

El proyecto empleará la metodología de Scrum ágil para la elaboración de la aplicación para dispositivos móviles. Esta metodología se distingue por su enfoque en ciclos iterativos llamados Sprint, que facilitan una entrega continua de valor y una rápida adaptación a los cambios. Cada sprint incluirá fases de planificación, ejecución, revisión y retrospectiva, asegurando una evolución constante del producto

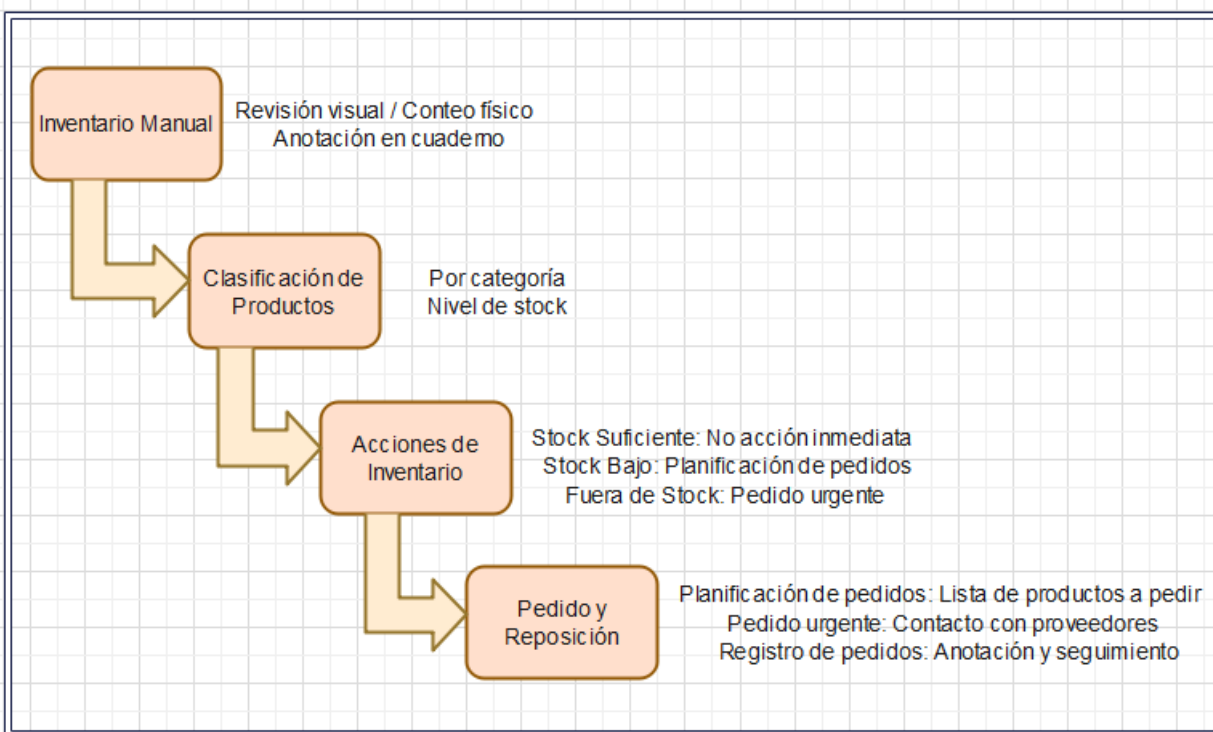
Capítulo I

Procesos Actuales

Los procesos actuales de gestión de inventario en Viveres y Licores 24 Horas son manuales y poco precisos. El inventario es manual, donde se realiza una revisión visual y un conteo físico de los productos, anotando la información en un cuaderno. A continuación, se presenta un esquema de cómo se realiza actualmente cada proceso.

Imagen 1.

Procesos Actuales



Nota. Ilustra las etapas actuales de gestión de inventario, desde el inventario manual hasta el pedido y reposición de productos

Infraestructura Actual

La infraestructura tecnológica de la tienda es limitada, lo que impide la optimización del stock y la reposición de productos. La infraestructura tecnológica con la que cuenta se detalla a continuación.

Tabla 1.

Infraestructura Actual

Área	Cantidad	Dispositivo	Características
Caja Registradora	2	Dispositivo móvil	Unidad de procesamiento de cuatro núcleos
			4 GB de memoria RAM
			Almacenamiento interno de 64 GB
			Sistema operativo basado en Android
Inventario	1	Dispositivo móvil	Unidad de procesamiento de cuatro núcleos
			4 GB de memoria RAM
			Almacenamiento interno de 64 GB
			Sistema operativo basado en Android
Administración	1	Dispositivo móvil	Unidad de procesamiento de cuatro núcleos
			4 GB de memoria RAM
			Almacenamiento interno de 64 GB
			Sistema operativo basado en Android
Almacén	1	Dispositivo móvil	Unidad de procesamiento de cuatro núcleos
			4 GB de memoria RAM
			Almacenamiento 64 GB
			Sistema operativo basado en Android
Atención al Cliente	1	Dispositivo móvil	Unidad de procesamiento de cuatro núcleos
			4 GB de memoria RAM
			Almacenamiento interno de 64 GB
			Sistema operativo basado en Android

Nota. Esta tabla contiene los dispositivos móviles actuales

Metodología Scrum

Scrum es un enfoque ágil que se aplica para la administración y creación de proyectos de alta complejidad. En lugar de hacer un esfuerzo por entregar un producto final completo de una vez, Scrum facilita la entrega incremental y reiterativa, permitiendo responder rápidamente a los cambios y mejorar continuamente. (atlassian, 2024)

Etapas de Scrum

- **Sprint Planning:** En esta etapa, se planifica el trabajo que se realizará durante el sprint. Se seleccionan los elementos del Product Backlog, que consiste en un conjunto de elementos priorizados de cada una las cualidades, funciones, requisitos, ajustes y rectificaciones, que se completarán en el próximo sprint. El objetivo es determinar qué se puede entregar y cómo se logrará. (donetonic, 2022)
- **Daily Scrum:** Se llevarán a cabo reuniones diarias de 15 minutos aproximadamente en las que se responderán tres preguntas: ¿Qué hice ayer?, ¿Qué planeo hacer hoy? y ¿hay algún obstáculo en mi camino? El objetivo es sincronizar el trabajo del equipo y realizar ajustes rápidamente. (donetonic, 2022)
- **Sprint Review:** Al final del sprint, se presentará el trabajo completado a los interesados y recoge feedback. Esta revisión se enfoca en lo que se ha logrado y en la demostración de las funcionalidades completadas. Su objetivo es revisar el aumento del producto creado y ajustar el Product Backlog en función del feedback recibido. (donetonic, 2022)
- **Sprint Retrospective:** Después de revisar el sprint, el equipo se reúne para discutir lo que funcionó bien y lo que no y cómo mejorar en el próximo sprint. Su objetivo

es mejorar continuamente la colaboración y el proceso de desarrollo del equipo.
(donetonic, 2022)

- **Sprint Backlog:** Es un conjunto de tareas y componentes escogidos del Product Backlog lo cual se compromete a finalizar en el transcurso del sprint. Se proporciona una actualización diaria acerca del progreso. Su objetivo durante el sprint es mantener una visión clara y enfocada del trabajo en curso. (donetonic, 2022)

Funciones esenciales en Scrum

- **Product Owner;** Aumentar al máximo el valor del producto. El Product Owner administra el Product Backlog, establece prioridades para las tareas y garantiza que el equipo comprenda claramente los objetivos y requisitos del proyecto.

Responsabilidades:

Definir y transmitir el objetivo del producto.

Gestionar y priorizar el Product Backlog.

Garantizar que el equipo comprenda claramente los componentes del Backlog.

(alaimolabs, 2024)

- **Scrum Master:** Es el facilitador del equipo de Scrum. Ayuda a seguir las prácticas de Scrum, elimina impedimentos y facilita las reuniones de Scrum.

Responsabilidades:

Facilitar las ceremonias de Scrum (coordinación del sprint, reunión diaria, chequeo y retroalimentación de cada etapa de trabajo).

Eliminar impedimentos que afectan la productividad.

Promover, apoyar la adopción de Scrum y la organización. (alaimolabs, 2024)

- **Development:** Al final de cada etapa de trabajo, es responsable de proporcionar avances de los productos con probabilidad de ser entregables. Es autoorganizado y tiene múltiples usos.

Responsabilidades:

Planificar y gestionar su propio trabajo dentro del sprint.

Creación y entregar incrementos de producto.

Colaborar con el Product Owner para determinar los requisitos y encontrar soluciones técnicas. (alaimolabs, 2024)

Sprint (Tablas)

Tabla 2.

Actividades Sprint I.

Semanas	Actividades
1	Diseño y desarrollo del sistema de almacenamiento de datos
2	Crear un sistema de interacción gráfica para la aplicación
3	Conectar la aplicación con la base de datos
4	Pruebas piloto

Nota. Las actividades de cada semana están planificadas para asegurar una implementación efectiva y eficiente del proyecto. Es crucial seguir el cronograma para cumplir con los objetivos y plazos establecidos

Capítulo II

Análisis y diseño

El método manual actual es propenso a errores humanos y dificulta la actualización en tiempo real, resultando en una precisión limitada.

Se ha desarrollado una solución tecnológica en respuesta a las necesidades y requerimientos detectados a través de entrevistas con los responsables de la tienda Víveres y Licores 24 Horas.

El desarrollo de una aplicación de inventario permite el manejo de datos de manera organizada, centralizada y automatizada. Los responsables de cada departamento tendrán la posibilidad de consultar la información en el momento adecuado sin perder tiempo o información, lo que mejorará el cumplimiento de los procesos.

La aplicación se puede dividir en módulos según su concepto, cada uno de los cuales está diseñado para abordar un tema específico de la gestión de inventarios. Esto se hace para satisfacer todas las necesidades y satisfacer las exigencias de los usuarios.

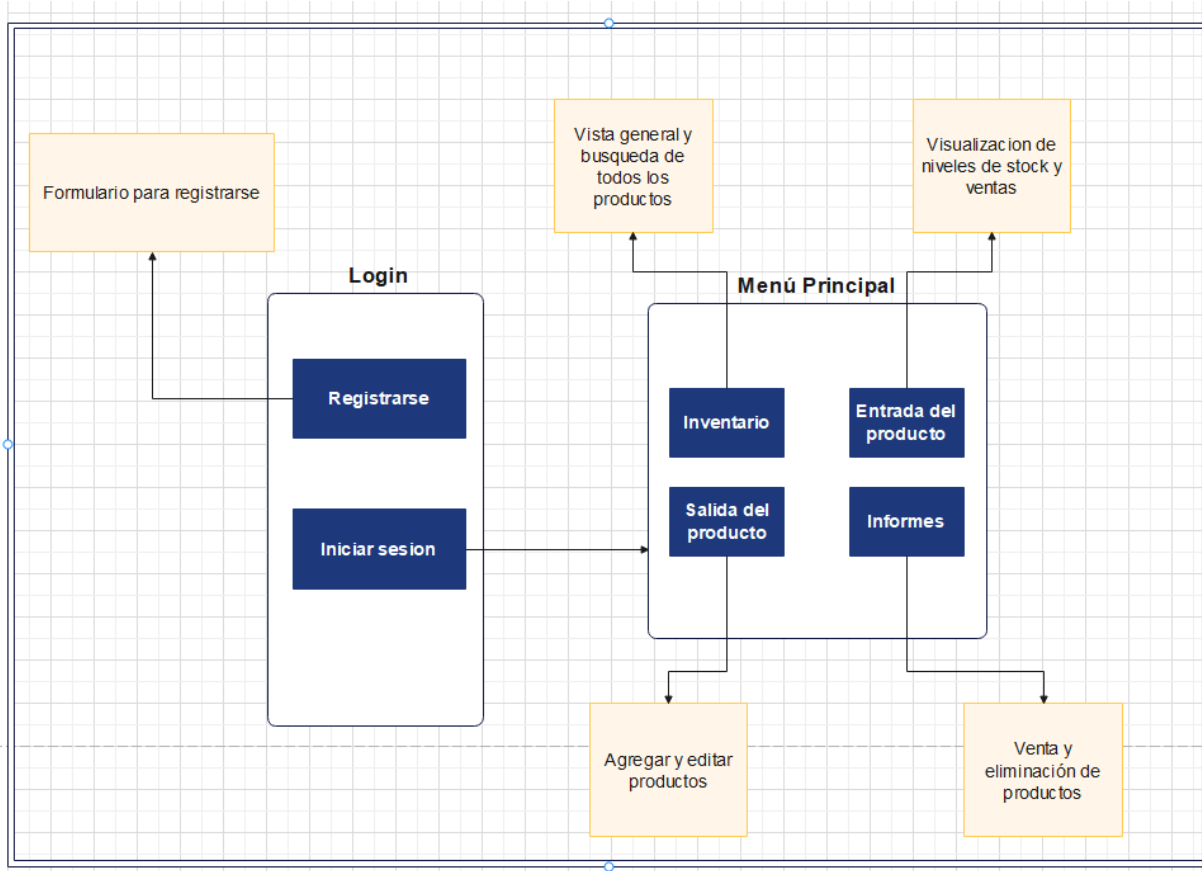
De forma similar, ha sido modificado el desarrollo de un sistema de datos centralizado que contendrá todos los detalles del inventario, incluidas cantidades, ventas y niveles de stock. Esto facilitará el acceso rápido y preciso a la información, lo que favorecerá el proceso de decisión y mejorará la gestión de inventarios.

Mapa de navegación

El mapa de navegación describe la estructura y el flujo de la aplicación móvil, asegurando una navegación intuitiva y accesible para los usuarios.

Imagen 2.

Mapa de navegación.



Nota. A partir de la pantalla principal de sesión, los usuarios pueden registrarse o iniciar sesión. Una vez dentro, el menú principal ofrece acceso a las principales funcionalidades, como la gestión de inventario, informes, entrada y salida de productos.

Viabilidad técnica

Se proporciona una tabla detallada que divide los requerimientos de software y hardware necesarios para desarrollar el proyecto.

Tabla 3.

Análisis de Viabilidad Técnica.

Categoría	Dispositivo	Características	Finalidad	Cantidad	Tipo	Licencia	Producto o Software
Hardware	Computadora portátil	Unidad de procesamiento Intel Core i9, 12 GB de RAM, Unidad de almacenamiento de 1TB	Diseño y Programación	1	-	-	Asus o similar
	Dispositivo móvil	Android mínimo 4 GB RAM	Pruebas de aplicación	1	-	-	Cualquier marca compatible
	Sistema Operativo	Windows 10	Desarrollo	1	Open Source	-	Sistema operativo preferido
Software	Plataforma de Gestión de Almacenamiento de Datos	MySQL	Almacenamiento de datos	1	Open Source	-	MySQL

	Explorador Web	Mozilla Firefox, Google Chrome	Pruebas de aplicación	1	Gratuito	Mozilla / Google	Firefox, Chrome
	Herramienta de Control de Versiones	Git	Control de versiones	1	Open Source	-	Git

Nota. Esta tabla de análisis de viabilidad técnica detalla los recursos necesarios en términos de hardware y software para el desarrollo y prueba de la aplicación

Viabilidad económica

El desarrollo del proyecto requiere la gestión personal, de igual manera los desembolsos generales, como la movilidad y los suministros esenciales (Energía eléctrica, agua potable, acceso a internet entre otros).

Tabla 4.

Análisis de Viabilidad Técnica.

Unidades	Instrumento	Permiso	Precio
1	Servidor para la Web	Código abierto	\$ 0,0
TOTALIDAD			\$ 0,0

Nota. Muestra un análisis de viabilidad técnica para el servidor web requerido en el proyecto.

Tabla 5.

Gastos en hardware.

Unidades	Instrumento	Temporalidad	Precio
1	Servidor para la Web	Vigente	\$ 0.0
1	Computadora Portátil	Único	\$ 0.0
1	Conexión en línea	Mensualmente	\$ 30.0
TOTALIDAD			\$ 30.0

Nota. Detalla los costos asociados con el hardware necesario para el proyecto.

Tabla 6.

Gastos Recursos Humanos y Recursos Complementarios.

Cantidad	Cargo	Periodicidad	Valor Mensual	Total
1	Programador	120 días	\$ 0.00	\$ 0.0

1	Suministros esenciales	120 días	\$ 35.0	\$ 140.0
1	Desembolsos comunes	120 días	\$ 30.0	\$ 120.0
TOTALIDAD				\$ 260.0

Nota. Proporciona un desglose de los costos de recursos humanos y adicionales

Tabla 7.

Reporte de Gastos.

Detalle	Valor
Herramienta digital	\$ 0.0
Equipos físicos	\$ 30.0
Personal - Suplementarios	\$ 260.0
TOTALIDAD	\$ 290.0

Nota. Proporciona un resumen consolidado de los costos del proyecto.

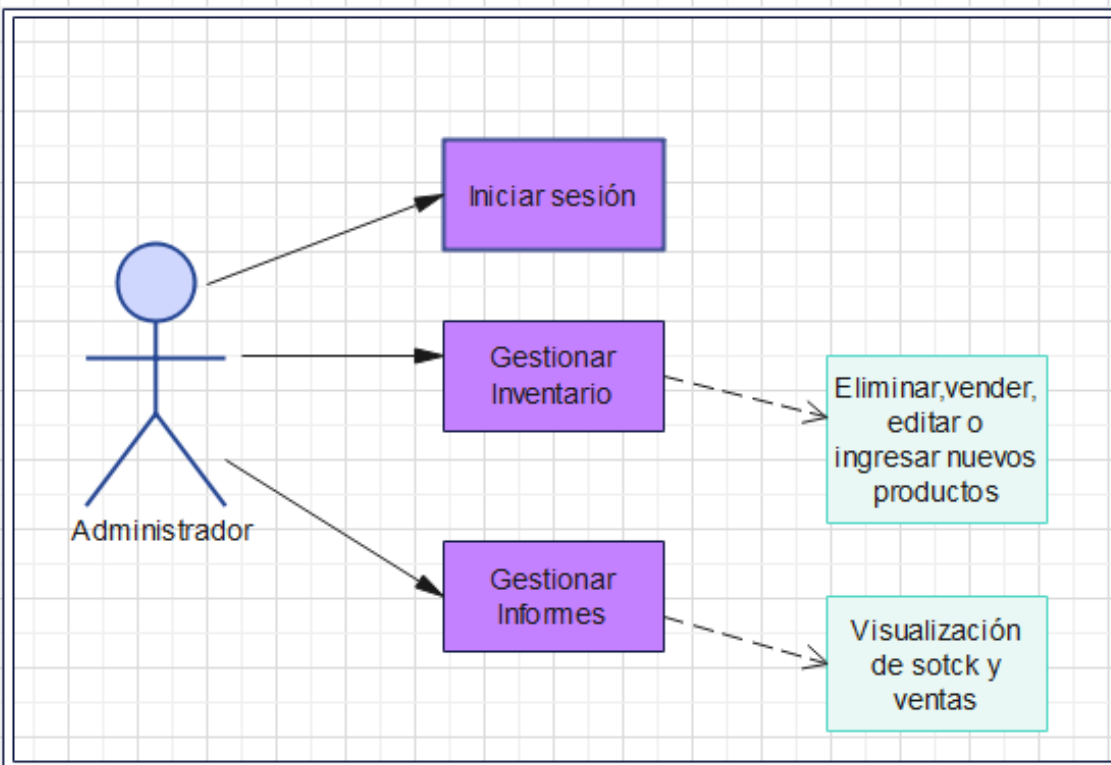
Casos de uso

La aplicación permite a los administradores realizar tareas cruciales de manera eficiente.

Administrador: Los administradores pueden iniciar sesión, gestionar productos, niveles de inventario y generar informes detallados para análisis estratégico.

Imagen 3.

Casos de Uso Administrador.



Nota. El diagrama muestra las interacciones clave del administrador con el sistema. Incluye las acciones de iniciar sesión, gestionar inventario y gestionar informes.

Tabla 8.

Casos de Uso.

Escenario de uso	Parámetros
Participantes	Gestor
Flujo habitual	Alternativa posible
1. Autenticarse	
2. Gestión de Productos	Eliminar, vender editar o ingresar nuevos productos
4. Generación de Informes	Visualizar productos en stock y ventas

Nota. La tabla de caso de uso describe las configuraciones y acciones del administrador dentro del sistema

Capítulo III

Desarrollo

Los requerimientos del sistema para la aplicación móvil de gestión de inventarios en la tienda Víveres y Licores 24 Horas abarcan las capacidades esenciales que debe tener, como la gestión de productos, seguimiento de inventarios, generación de reporte. Específicamente, la aplicación debe permitir a los usuarios agregar, editar y eliminar productos del inventario, monitorear y actualizar los niveles de stock, generar reportes sobre el estado del inventario y ventas. La interfaz de usuario debe ser simple de entender y fácil de usar.

La siguiente tabla muestra las tareas que se realizarán.

Tabla 9.

Programación de tareas.

Módulo	Iteración	ID	Requerimiento	Prioridad	Tarea	Responsable
Recolección de Información	1	1	Verificación de procedimientos actuales	Alta	Entrevistas con el personal	Steeven Jimenez
				Alta	Análisis de documentación existente	Steeven Jimenez
		2	Revisión de necesidades	Alta	Confirmación del proceso de información	Steeven Jimenez
				Alta	Requisitos del sistema	Steeven Jimenez
		3	Identificación de problemas	Alta	Identificación de carencias	Steeven Jimenez

				Alta	Identificación de problemas en la gestión actual	Steeven Jimenez
Diseño Previo	1	4	Modelado de la base de datos (BDD)	Alta	Identificación sobre entidades y relaciones clave	Steeven Jimenez
		5	Modelo Entidad-Relación (MER)	Alta	Diseño del modelo MER para el almacén de datos	Steeven Jimenez
Desarrollo Frontend	1	6	Modelado de la interfaz gráfica (UI)	Alta	Creación de prototipos	Steeven Jimenez
		7	Creación de la pantalla de Login	Alta	Diseño del Frontend para administrar los usuarios	Steeven Jimenez

		8	Implementación del menú principal	Alta	Diseño e implementación del menú principal de la aplicación	Steeven Jimenez
		9	Creación de la pantalla de inventario	Alta	Desarrollo del Frontend para la visualización y búsqueda de productos	Steeven Jimenez
		10	Creación de la pantalla de entrada del producto	Alta	Desarrollo del Frontend para agregar o editar los niveles de stock	Steeven Jimenez

		11	Creación de la pantalla de salida del producto	Alta	Desarrollo del Frontend para la venta de productos	Steeven Jimenez
		12	Creación de la pantalla de Informes	Alta	Desarrollo del Frontend para administrar las ventas y el stock	Steeven Jimenez
Desarrollo Backend	1	13	Implementación de la API REST	Alta	Desarrollo de endpoints para operaciones CRUD	Steeven Jimenez
		14	Integración con la plataforma de datos	Alta	Establecimiento de la conexión con el sistema de datos	Steeven Jimenez

Pruebas e Integración	1	15	Pruebas unitarias	Alta	Desarrollo y ejecución de pruebas unitarias para el Frontend y Backend	Steeven Jimenez
		16	Pruebas de integración	Alta	Verificación de la integración correcta entre Frontend y Backend	Steeven Jimenez
Despliegue y Capacitación	1	17	Despliegue en servidor de producción	Alta	Configuración del entorno de producción	Steeven Jimenez
					Despliegue de la aplicación	Steeven Jimenez

		18	Capacitación del personal	Alta	Entrenamiento del personal en el uso de la nueva aplicación	Steeven Jimenez
--	--	----	---------------------------	------	---	-----------------

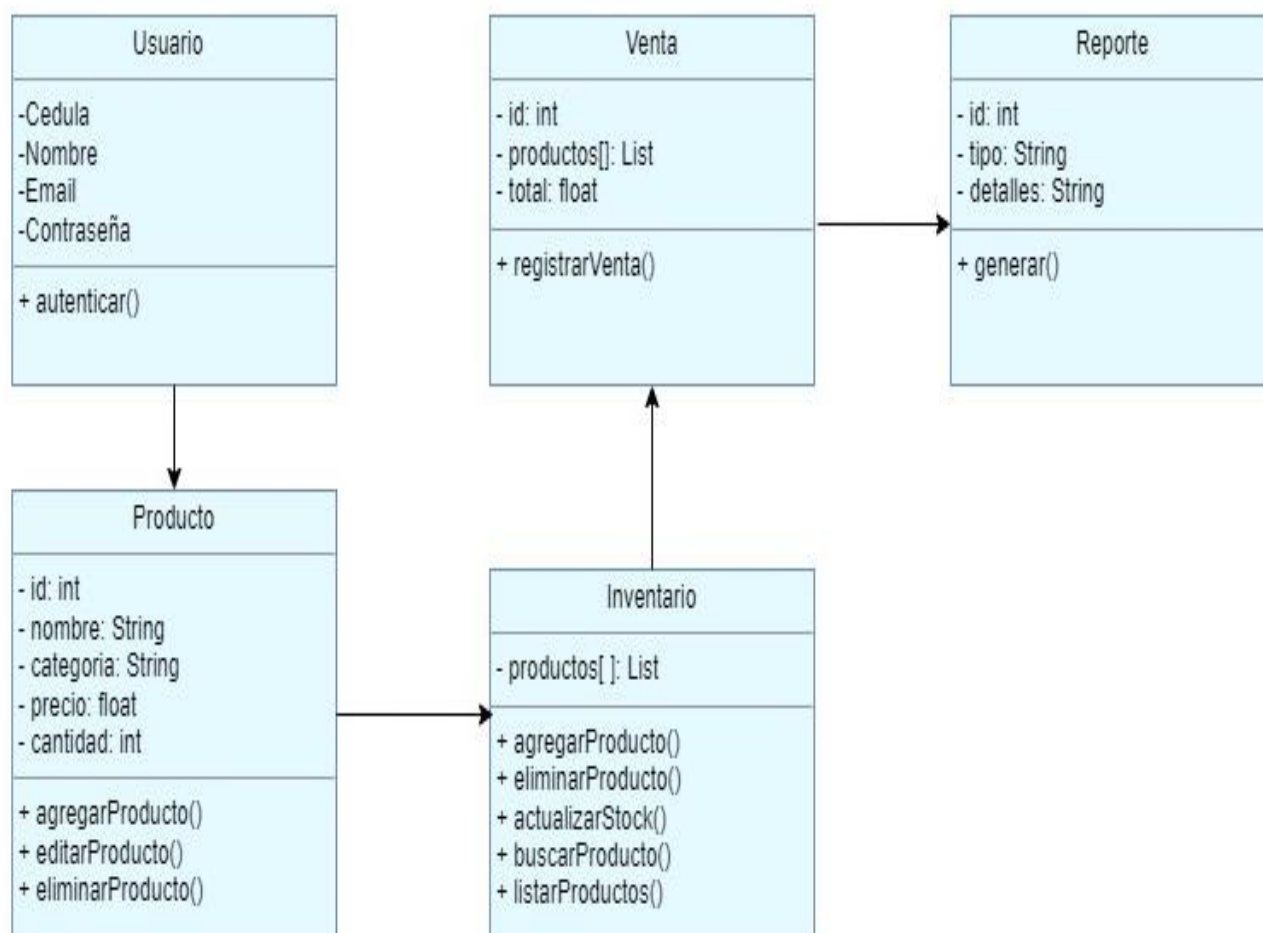
Nota. Esta tabla detalla el plan de acción para el proyecto, desglosando las tareas específicas, responsables, y prioridades en cada módulo e iteración.

Diagrama de clases

La representación de clases detalla las clases y sus conexiones en el programa. Para la aplicación móvil de gestión de inventarios en la tienda Víveres y Licores 24 Horas, el diagrama de clases incluye las principales entidades y las interacciones entre ellas. A continuación, se presenta el diagrama de clases detallado:

Imagen 4.

Diagrama de Clases.



Nota. El diagrama presentado ilustra la estructura y las relaciones entre las principales entidades del programa.

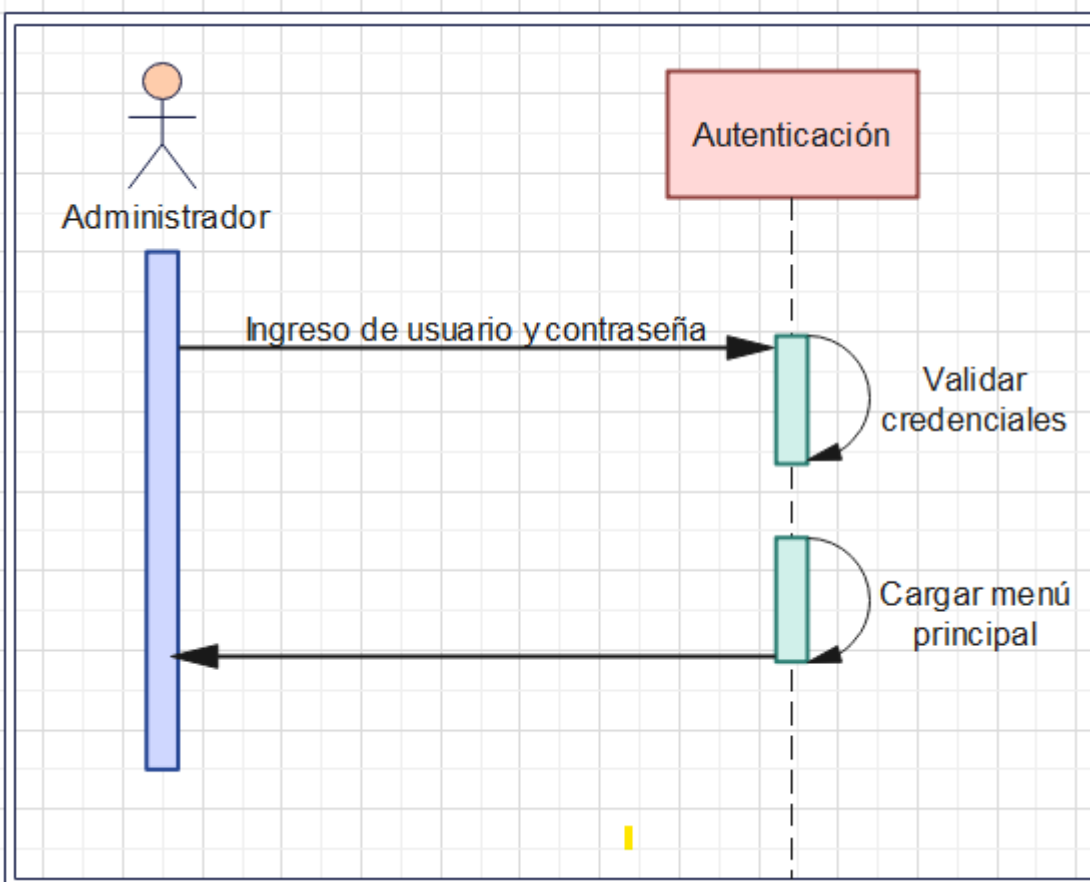
Diagrama de secuencia

El gráfico de secuencia ilustra las interacciones sobre una manera más detallada, con cada acción desencadenando una serie de mensajes entre los componentes del sistema

Esta representación visual asegura que todas las operaciones se realizan de manera ordenada y eficiente, facilitando la comprensión del flujo de trabajo y la coordinación entre las diferentes partes del sistema.

Imagen 5.

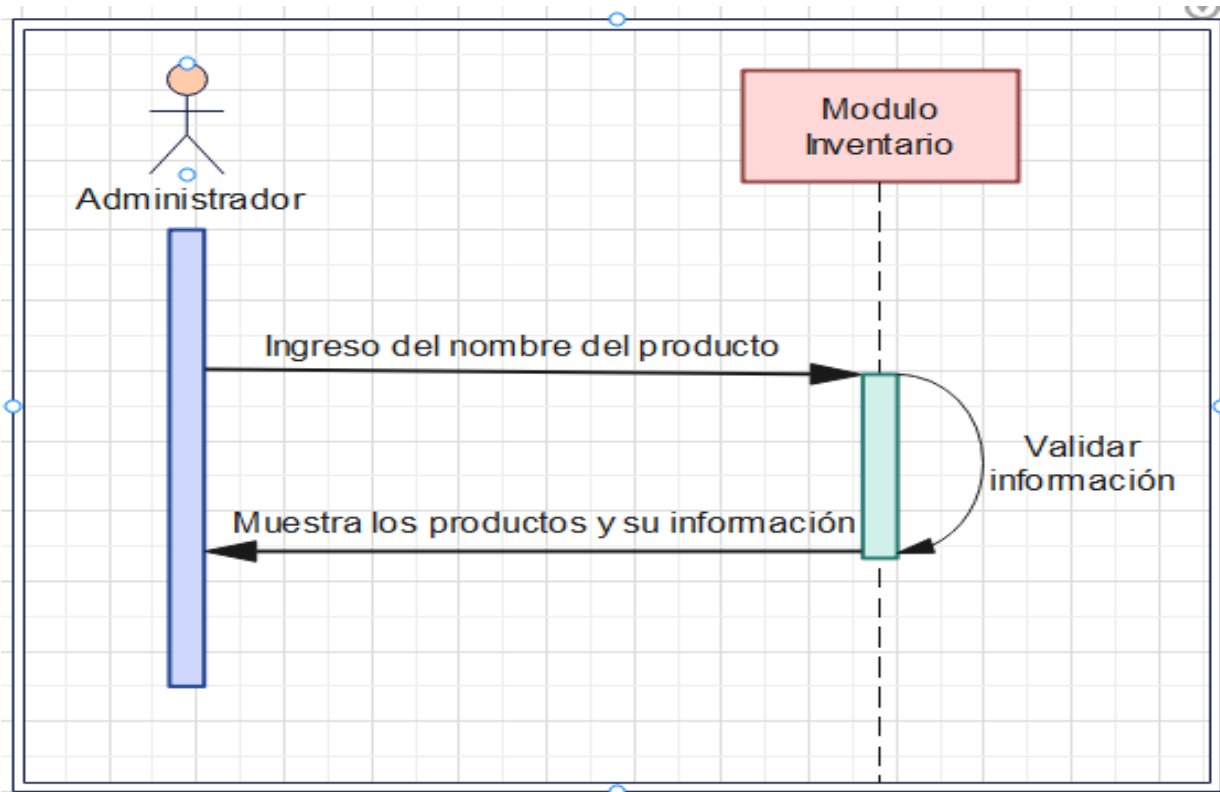
Modulo Autenticación.



Nota. El diagrama muestra el proceso de autenticación del administrador.

Imagen 6.

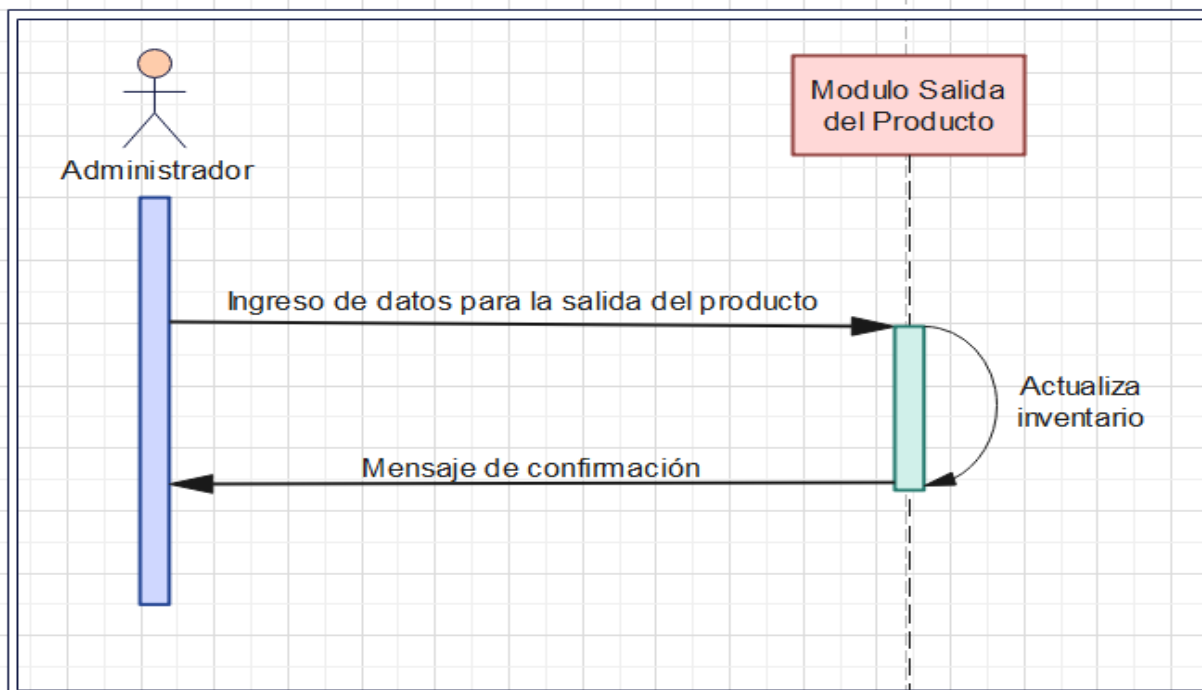
Modulo Inventario.



Nota. El diagrama ilustra el proceso de gestión del inventario por parte del administrador.

Imagen 7.

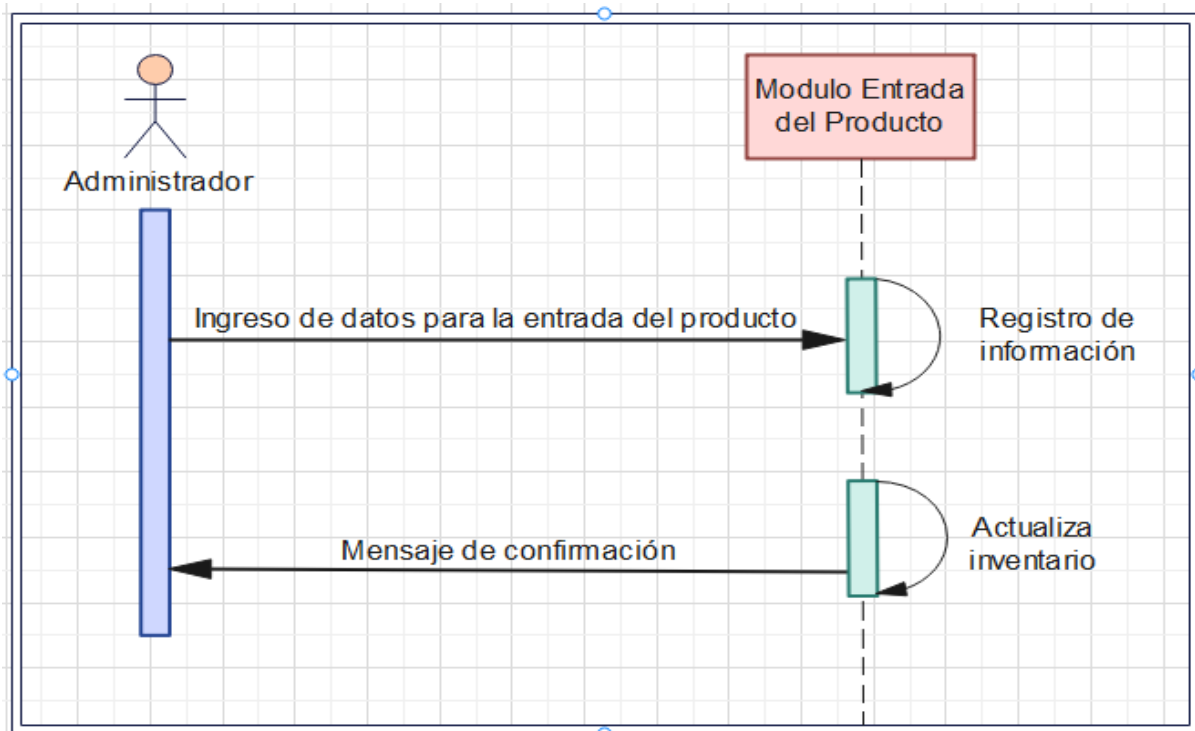
Modulo Salida del Producto.



Nota. El diagrama describe el proceso de registro de salidas de productos del inventario

Imagen 8.

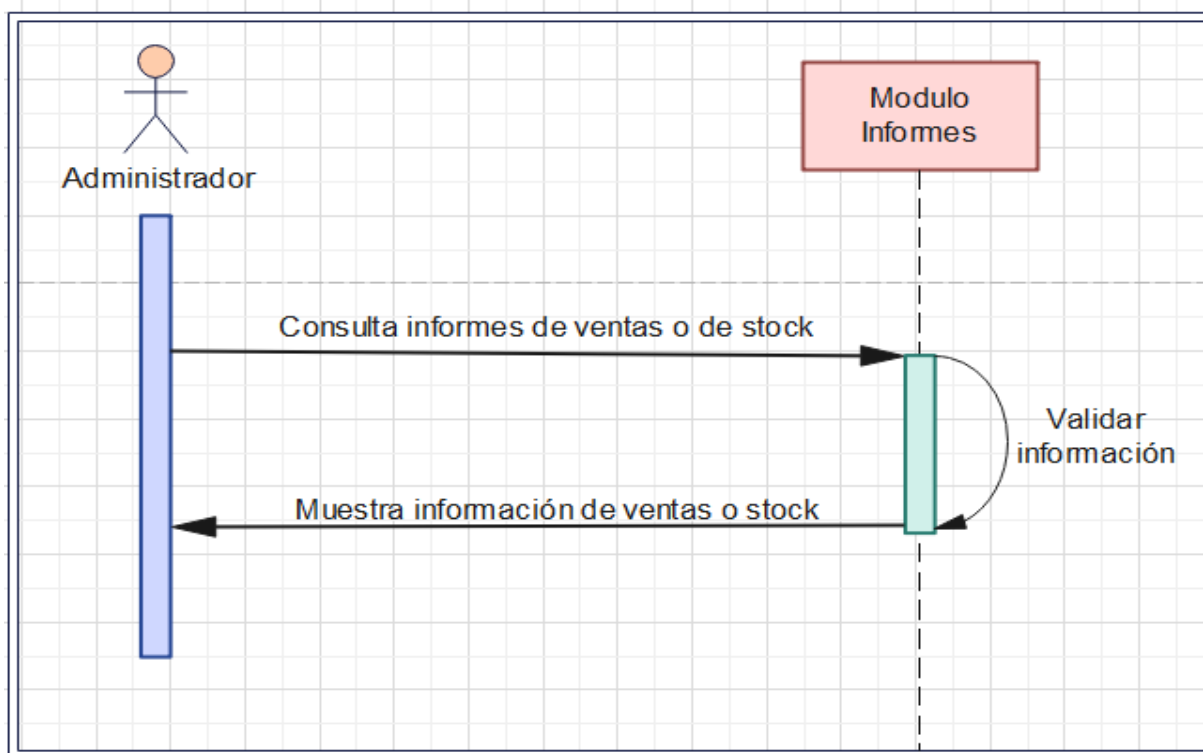
Modulo Entrada del Producto.



Nota. El diagrama detalla el proceso de registro de entradas de productos en el inventario.

Imagen 9.

Modulo Informes.



Nota. El diagrama muestra cómo el administrador consulta informes de ventas o de stock.

Construcción

La construcción de la aplicación móvil para la gestión de inventarios se llevará a cabo en varias fases, asegurando que cada componente del sistema se desarrolle, integre y pruebe de manera metódica y eficiente.

MainActivity: Este código en Java define una actividad principal (MainActivity) para una aplicación Android que proporciona funcionalidad de inicio de sesión y registro de usuario. Se establecen referencias a los campos de información para el usuario y clave de acceso, así como a los controles de creación de cuenta y acceso en el método onCreate. Al seleccionar el botón de acceso, se verifica la coincidencia del usuario y clave de acceso. Si coinciden, se muestra un mensaje de éxito y el usuario se redirige a una nueva actividad (DashboardActivity). Se muestra un mensaje de error si no coinciden. Al seleccionar el botón de inscripción, el usuario se dirige a una actividad de registro. (RegisterActivity).

Imagen 10.

MainActivity.

```
12
13 ▶ </> public class MainActivity extends AppCompatActivity {
14
15     private EditText usernameEditText; 2 usages
16     private EditText passwordEditText; 2 usages
17     private Button loginButton; 2 usages
18     private Button registerButton; 2 usages
19
20     @Override
21     protected void onCreate(Bundle savedInstanceState) {
22         super.onCreate(savedInstanceState);
23         setContentView(R.layout.activity_main);
24
25         usernameEditText = findViewById(R.id.username);
26         passwordEditText = findViewById(R.id.password);
27         loginButton = findViewById(R.id.loginButton);
28         registerButton = findViewById(R.id.registerButton);
29
30         loginButton.setOnClickListener(new View.OnClickListener() {
31             @Override
32             public void onClick(View v) {
33                 String username = usernameEditText.getText().toString();
34                 String password = passwordEditText.getText().toString();
35
36                 if (username.equals("admin") && password.equals("password")) {
37                     Toast.makeText(context: MainActivity.this, text: "Inicio de sesión exitosa", Toast.LENGTH_SHORT).show();
38                     Intent intent = new Intent(packageContext: MainActivity.this, DashboardActivity.class);
39                     startActivity(intent);
40                 } else {
41                     Toast.makeText(context: MainActivity.this, text: "Usuario o contraseña inválida", Toast.LENGTH_SHORT).show();
42                 }
43             }
44         });
45
46         registerButton.setOnClickListener(new View.OnClickListener() {
47             @Override
48             public void onClick(View v) {
49                 Intent intent = new Intent(packageContext: MainActivity.this, RegisterActivity.class);
50                 startActivity(intent);
51             }
52         });
53     }
54 }
```

DashboardActivity: Este código en Java define la actividad del panel principal (DashboardActivity) para una aplicación Android. Esta actividad proporciona varios botones, tanto de texto como de imagen, para navegar a diferentes secciones de la aplicación, como inventario, entrada de productos, salida de productos e informes.

Imagen 11.

DashboardActivity.

```
12 ▶ </> public class DashboardActivity extends AppCompatActivity {
13
14     private Button inventarioButton; 2 usages
15     private Button entradaProductoButton; 3 usages
16     private Button salidaProductoButton; 3 usages
17     private Button informesButton; 3 usages
18     private ImageButton inventarioImageButton; 2 usages
19     private ImageButton entradaProductoImageButton; 2 usages
20     private ImageButton salidaProductoImageButton; 2 usages
21     private ImageButton informesImageButton; 2 usages
22
23     @Override
24     protected void onCreate(Bundle savedInstanceState) {
25         super.onCreate(savedInstanceState);
26         setContentView(R.layout.activity_dashboard);
27
28         inventarioButton = findViewById(R.id.inventarioButton);
29         entradaProductoButton = findViewById(R.id.entradaProductoButton);
30         salidaProductoButton = findViewById(R.id.salidaProductoButton);
31         informesButton = findViewById(R.id.informesButton);
32         inventarioImageButton = findViewById(R.id.inventarioImageButton);
33         entradaProductoImageButton = findViewById(R.id.entradaProductoImageButton);
34         salidaProductoImageButton = findViewById(R.id.salidaProductoImageButton);
35         informesImageButton = findViewById(R.id.informesImageButton);
36
37         View.OnClickListener inventarioClickListener = new View.OnClickListener() {
38             @Override
39             public void onClick(View v) {
40                 Intent intent = new Intent(packageContext: DashboardActivity.this, ProductListActivity.class);
41                 startActivity(intent);
42             }
43         };
44
45         inventarioButton.setOnClickListener(inventarioClickListener);
46         inventarioImageButton.setOnClickListener(inventarioClickListener);
47
```

```

49     entradaProductoButton.setOnClickListener(new View.OnClickListener() {
50         @Override
51         public void onClick(View v) {
52             Intent intent = new Intent( packageContext: DashboardActivity.this, AddProductActivity.class);
53             startActivity(intent);
54         }
55     });
56
57     entradaProductoImageButton.setOnClickListener(new View.OnClickListener() {
58         @Override
59         public void onClick(View v) { entradaProductoButton.performClick(); }
60     });
61
62
63
64
65     salidaProductoButton.setOnClickListener(new View.OnClickListener() {
66         @Override
67         public void onClick(View v) {
68             Toast.makeText( context: DashboardActivity.this, text: "Salida del Producto", Toast.LENGTH_SHORT).show();
69             // Add code to handle product output
70         }
71     });
72
73     salidaProductoImageButton.setOnClickListener(new View.OnClickListener() {
74         @Override
75         public void onClick(View v) { salidaProductoButton.performClick(); }
76     });
77
78
79
80     informesButton.setOnClickListener(new View.OnClickListener() {
81         @Override
82         public void onClick(View v) {
83             Toast.makeText( context: DashboardActivity.this, text: "Informes", Toast.LENGTH_SHORT).show();
84             // Add code to open reports activity
85         }
86     });
87
88     informesImageButton.setOnClickListener(new View.OnClickListener() {
89         @Override
90         public void onClick(View v) { informesButton.performClick(); }
91     });
92
93

```

WebService: El propósito de este script PHP es registrar usuarios en el sistema de datos gestionada por MySQL. Establece el vínculo con el sistema de gestión de datos, valida los datos recibidos y los agrega a la tabla de usuarios si todos los campos están completados. Utiliza sentencias preparadas para prevenir inyecciones SQL y proporciona mensajes de éxito o error dependiendo del resultado de la operación.

Imagen 12

WebService

```

1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "";
5  $dbname = "inventario";
6
7  $conn = new mysqli($servername, $username, $password, $dbname);
8
9  if ($conn->connect_error) {
10     die("Conexión fallida: " . $conn->connect_error);
11 }
12
13 $cedula = isset($_POST['cedula']) ? $_POST['cedula'] : null;
14 $name = isset($_POST['name']) ? $_POST['name'] : null;
15 $username = isset($_POST['username']) ? $_POST['username'] : null;
16 $email = isset($_POST['email']) ? $_POST['email'] : null;
17 $password = isset($_POST['password']) ? $_POST['password'] : null;
18
19 if ($cedula && $name && $username && $email && $password) {
20     $sql = "INSERT INTO users (cedula, name, username, email, password) VALUES (?, ?, ?, ?, ?)";
21     $stmt = $conn->prepare($sql);
22     $stmt->bind_param("sssss", $cedula, $name, $username, $email, $password);
23
24     if ($stmt->execute()) {
25         echo "Registro exitoso";
26     } else {
27         echo "Error: " . $sql . "<br>" . $conn->error;
28     }
29
30     $stmt->close();
31 } else {
32     echo "Por favor, complete todos los campos.";
33 }

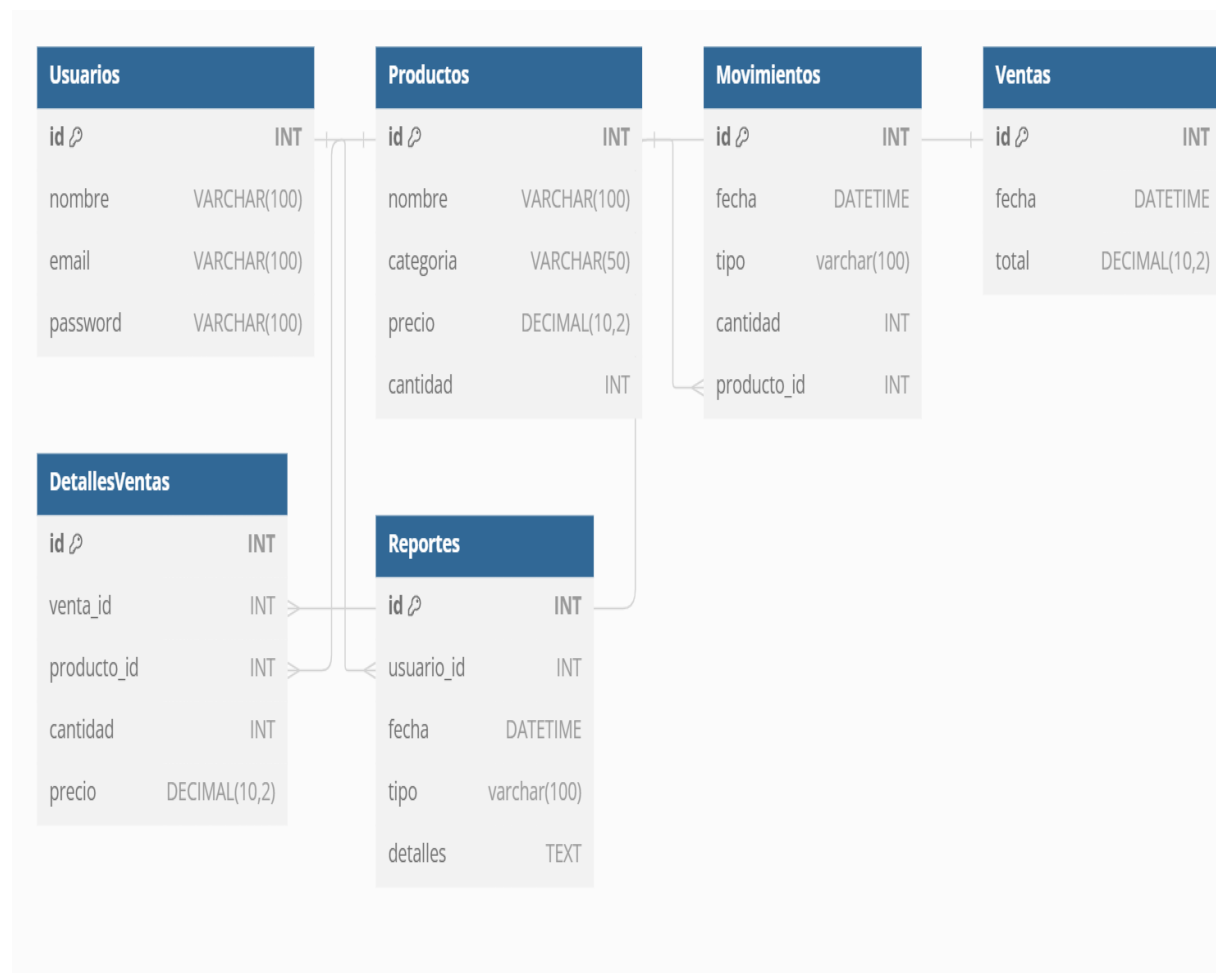
```

Modelo BDD

El modelo de base de datos es una representación estructurada de cómo se organizarán y gestionarán los datos. Este modelo es crucial porque define la configuración de un sistema de gestión de datos, que incluye los conjuntos de datos, los campos y las asociaciones entre ellos, asegurando que los datos estén almacenados de manera efectiva y accesible.

Imagen 13.

Modelo BDD.



Nota. El diagrama de base de datos muestra la estructura y las relaciones entre las distintas tablas que conforman el sistema.

Diccionario de datos

Cada campo que se encuentra en las tablas del sistema de gestión de datos se describe dentro del detalle en el diccionario de datos. Este recurso es esencial para comprender la

estructura y el propósito de los datos almacenados en cada tabla, lo que mejora la gestión y el cuidado del sistema.

Tabla 10.

Descripción y contenido de las tablas.

Tabla	Descripción
Usuarios	Lista de inscripción de los usuarios del sistema, incluyendo nombre, email y contraseña.
Productos	Tabla de registro de los productos del inventario, incluyendo nombre, categoría, precio y cantidad.
Movimientos	Tabla de registro de los movimientos de inventario, tanto entradas como salidas, con detalles de fecha, tipo, cantidad y producto asociado.
Ventas	Tabla de registro de las transacciones efectuadas, especificando momento y total de cada transacción.
DetallesVentas	Tabla de registro de los pormenores de cada transacción, incluyendo los artículos vendidos, la cantidad y el costo
Reportes	Lista de registro de los reportes generados por los usuarios, con detalles sobre ventas y stock.

Nota. Esta tabla describe las diferentes tablas que componen la base de datos del sistema.

Tabla 11.

Usuarios

Atributo	Categoría de Datos	Definición	Condiciones	PK	FK
id	INT	Código exclusivo del usuario	AUTO_INCREMENT	x	
nombre	VARCHAR(100)	Nombre personal del usuario	NOT NULL		
email	VARCHAR(100)	Dirección de dirección de email	NOT NULL, UNIQUE		
password	VARCHAR(100)	Clave del usuario			

Nota. Esta tabla almacena la información de los usuarios que utilizan el sistema.

Tabla 12.

Productos.

Atributo	Categoría de Datos	Definición	Condiciones	PK	FK
id	INT	Código exclusivo del producto	AUTO_INCREMENT	x	
nombre	VARCHAR(100)	Seudónimo del producto	NOT NULL		
categoría	VARCHAR(50)	Tipo de producto al que se clasifica	NOT NULL		
precio	DECIMAL(10,2)	Costo de venta del artículo			

cantidad	INT	Cantidad disponible en el inventario			
----------	-----	--------------------------------------	--	--	--

Nota. Esta tabla contiene la información de los productos disponibles en el inventario.

Tabla 13.

Movimientos.

Atributo	Categoría de Datos	Definición	Condiciones	PK	FK
id	INT	Código exclusivo del movimiento	AUTO_INCREMENT	x	
Fecha	DATETIME	Fecha del movimiento	NOT NULL		
Tipo	VARCHAR(100)	Tipo de movimiento (entrada/salida)	NOT NULL		
cantidad	INT	Cantidad de productos movidos	NOT NULL		
producto_id	INT	Identificador del producto	(referencia a Productos(id))		x

Nota. Esta tabla registra los movimientos de inventario, como entradas y salidas de producto

Tabla 14.

Ventas.

Atributo	Categoría de Datos	Definición	Condiciones	PK	FK
id	INT	Código exclusivo de la venta	AUTO_INCREMENT	x	
fecha	DATETIME	Día de la transacción	NOT NULL		
total	DECIMAL(10,2)	Monto totalidad de la transacción	NOT NULL		

Nota. Esta tabla almacena la información general de las ventas realizadas.

Tabla 15.

Detalles ventas.

Atributo	Categoría de Datos	Definición	Condiciones	PK	FK
id	INT	Código exclusivo del detalle de transacción	AUTO_INCREMENT	x	
venta_id	INT	Código exclusivo de la transacción	(referencia a Ventas(id))		x
producto_id	INT	Código exclusivo del producto	(referencia a Productos(id))		x
cantidad	INT	Número de unidades de producto comercializadas	NOT NULL		
precio	DECIMAL(10,2)	Costo del producto vendido	NOT NULL		

Nota. Esta tabla almacena los detalles de cada producto vendido en una venta.

Tabla 16.

Reportes.

Atributo	Categoría de Datos	Definición	Condiciones	PK	FK
id	INT	Código exclusivo del reporte	AUTO_INCREMENT	x	
usuario_id	INT	Código exclusivo que genera el reporte	(referencia a Usuarios(id))		x
fecha	DATETIME	Fecha de generación del reporte	NOT NULL		
tipo	VARCHAR(50)	Tipo de reporte (ventas/stock)	NOT NULL		
detalles	VARCHAR(100)	Detalles adicionales del reporte			

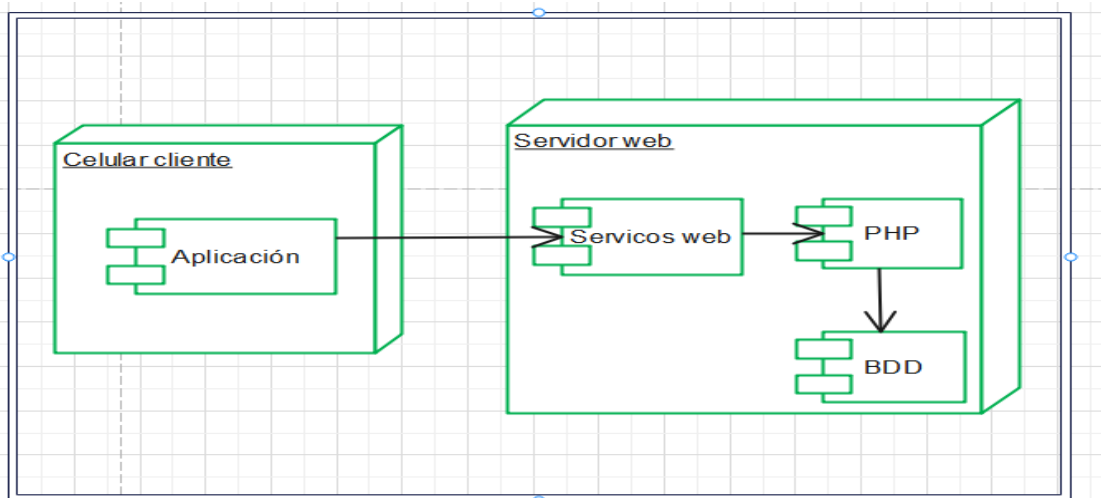
Nota. Esta tabla almacena los reportes generados por los usuarios, que pueden incluir detalles sobre ventas o inventarios.

Arquitectura

Se refiere a cómo los componentes de un sistema de software se organizan e interactúan entre sí. La arquitectura define cómo se dividen las responsabilidades entre el cliente (la aplicación móvil), el servidor (que maneja la lógica de negocio), y la base de datos (que almacena la información crítica).

Imagen 14.

Arquitectura



Capítulo IV

Registro de Iteración

En esta sección se detallan las iteraciones realizadas durante el desarrollo de la aplicación. Cada tabla muestra las diferentes etapas del proceso, especificando el módulo implicado, los prerrequisitos necesarios, los actores participantes, el proceso y subprocesos ejecutados, las tareas llevadas a cabo, el resultado alcanzado y las observaciones relevantes. Estas iteraciones reflejan un enfoque metódico y progresivo para garantizar la correcta implementación de funcionalidades clave como registrarse como usuario, iniciar sesión, generar informes, actualizar el inventario y registrar los productos. El objetivo de este enfoque es asegurar que cada componente del sistema cumpla con los requisitos establecidos, optimizando las validaciones y mejorando continuamente la calidad del sistema a través de revisiones y ajustes iterativos.

Tabla 17.

Iteración 1 para Registro de Usuarios.

Bloque	Fase	Condición previa	Rol	Procedimiento	Proceso secundario	Actividad	Logro Alcanzado	Consideraciones
Autenticación	1	Acceso a la sección de inscripción	Usuario	Creación de Usuarios	Crear formulario de registro	Se creó el formulario y se insertó la información en la base de datos	60%	Falta mejorar las validaciones de las credenciales

Nota: Esta tabla describe la primera iteración del proceso de registro de usuarios.

Tabla 18.

Iteración 2 para Registro de Usuarios.

Bloque	Fase	Condición previa	Rol	Procedimiento	Proceso secundario	Actividad	Logro Alcanzado	Consideraciones
Autenticación	2	Mejorar validaciones de credenciales	Operador	Creación de Usuarios	Validar credenciales	Se mejoraron las validaciones de las credenciales en el formulario	80%	Las validaciones han sido mejoradas, pero aún se requiere optimización.

Nota. Esta tabla describe la segunda iteración del proceso de registro de usuarios.

Tabla 19.

Iteración 3 para Registro de Usuarios.

Bloque	Fase	Condición previa	Rol	Procedimiento	Proceso secundario	Actividad	Logro Alcanzado	Consideraciones
Autenticación	3	Ajuste de validaciones	Operador	Creación de Usuarios	Ajustes finales	Se realizaron ajustes finales en las validaciones, como verificación de formato de contraseñas, y verificación de campos obligatorios	100%	Validaciones completadas y funcionando correctamente.

Nota: Esta tabla describe la tercera iteración del proceso de registro de usuarios.

Tabla 20.

Iteración 1 para Inicio de Sesión.

Bloque	Fase	Condición previa	Rol	Procedimiento	Proceso secundario	Actividad	Logro Alcanzado	Consideraciones
Autenticación	1	Entrada a la pantalla de autenticación	Operador	Autenticación	Iniciar sesión	Se creó el formulario de inicio de sesión y se verificaron las credenciales	100%	El formulario y la verificación funcionan correctamente.

Nota: Esta tabla detalla la primera iteración del proceso de inicio de sesión.

Tabla 21.

Iteración 1 para Generación de Informes.

Bloque	Fase	Condición previa	Rol	Procedimiento	Proceso secundario	Actividad	Logro Alcanzado	Consideraciones
Informes	1	Ingreso a la sección de informes	Operador	Generar Informe	Informe de Ventas	Se creó el formulario y se generó el informe de ventas	100%	El formulario y el informe se generaron correctamente.

Nota. Esta tabla muestra la primera iteración del proceso de generación de informes.

Tabla 22.

Iteración 1 para Actualización de Inventario.

Bloque	Fase	Condición previa	Rol	Procedimiento	Proceso secundario	Actividad	Logro Alcanzado	Consideraciones
Inventario	1	Ingreso a la pantalla de inventario	Operador	Gestión de Inventario	Modificar Producto	Se actualizó la cantidad de productos en el almacén de datos	100%	El total de productos se actualizó correctamente en la base de datos.

Nota. Esta tabla describe la primera iteración del proceso de actualización de inventario.

Tabla 23

Iteración 1 para Registro de Productos

Bloque	Fase	Condición previa	Rol	Procedimiento	Proceso secundario	Actividad	Logro Alcanzado	Consideraciones
Inventario	1	Ingreso a la sección de inventario	Operador	Gestión de Inventario	Crear Producto	Se creó el formulario y se insertó la información registrada en el repositorio de datos	100%	El formulario y la inserción en la base de datos se realizaron correctamente.

Nota. Esta tabla detalla la primera iteración del proceso de registro de productos.

Pruebas funcionales

Dado que las pruebas de caja negra se enfocan en verificar el comportamiento del software sin entender su diseño interno, se utilizarán para las pruebas funcionales.

Caso de Prueba 1: Registro de Producto

- Verificar que se puede registrar un nuevo producto en el inventario.
- Precondición: El usuario está autenticado y en la sección de gestión de inventario.
- Entrada: Datos de un nuevo producto (nombre: "Producto A", categoría: "Categoría X", precio: 10.00, cantidad: 100).
- Salida Esperada: Confirmación de registro exitoso y el producto aparece en la lista de inventario.
- Resultados Esperados: El producto debe ser agregado.
- Resultado de la Prueba: (Pasó)

Caso de Prueba 2: Actualización de Inventario

- Verificar que se puede actualizar la cantidad de un producto existente.
- Precondición: El usuario está autenticado y en la sección de gestión de inventario.
- Entrada: Modificación de la cantidad de "Producto A" a 150.
- Salida Esperada: Actualización exitosa y cantidad modificada reflejada en el inventario.
- Resultado Esperado: La cantidad del producto debe ser actualizada.
- Resultado de la Prueba: (Pasó)

Caso de Prueba 3: Generación de Informes

- Verificar la capacidad de generar un informe de ventas.
- Precondición: El usuario está autenticado y en la sección de informes.

- Entrada: Solicitud de informe de ventas para el periodo del 01/01/2024 al 30/06/2024.
- Salida Esperada: Informe detallado con las ventas registradas dentro del periodo especificado.
- Resultado Esperado: El informe debe generarse con los datos correctos.
- Resultado de la Prueba: (Pasó)

También se usarán las pruebas de caja blanca ya que se centran en la estructura interna y el funcionamiento del código. Se enfoca en pruebas a nivel de código y lógica interna.

Caso de Prueba 1: Verificación de Autenticación

- Validar que el proceso de autenticación haya verificado correctamente los nombres de usuario y contraseñas.
- Precondición: El usuario intenta iniciar sesión.
- Prueba: Validar los métodos en MainActivity.java.
- Cobertura: Todas las ramas de la condición de autenticación.
- Resultado Esperado: La autenticación debe ser exitosa con credenciales correctas.
- Resultado de la Prueba: (Pasó)

Caso de Prueba 2: Validación de Entradas en el Webservice

- Verificar que el script PHP maneja correctamente las entradas del formulario y previene la inyección SQL.
- Precondición: El usuario intenta registrar un nuevo usuario.
- Prueba: Validar las entradas en login.php.
- Cobertura: Todas las condiciones de validación de entrada.

- Resultado Esperado: Las entradas deben ser validadas correctamente y la inyección SQL debe ser prevenidas.
- Resultado de la Prueba: (Pasó)

Pruebas no funcionales

Las pruebas no funcionales garantizan que la aplicación cumple con los requisitos funcionales y cumple con los requisitos de rendimiento y usabilidad. Los principales tipos de pruebas no funcionales realizadas para la aplicación se detallan a continuación.

Prueba de rendimiento

- Verifica que la aplicación puede manejar la carga esperada y que responde dentro de los tiempos aceptables.
- Prueba de carga: Se evalúa el comportamiento de la aplicación cuando se somete a la carga máxima esperada. Se simuló múltiples usuarios realizando operaciones de inventario simultáneamente.
- Resultado esperado: La aplicación debe manejar al menos 100 usuarios simultáneos sin deterioro significativo en el rendimiento.
- Resultado de la prueba: (Pasó)
- Capturas de Pantalla de la Prueba:

Imagen 15

Informe Resumido.

The screenshot shows the Apache JMeter Summary Report window. The window title is 'Summary Report.jmx (C:\Users\steve\Downloads\apache-jmeter-5.6.3 (1)\apache-jmeter-5.6.3\bin\Summary Report.jmx) - Apache JMeter (5.6.3)'. The interface includes a menu bar (File, Edit, Search, Run, Options, Tools, Help), a toolbar with various icons, and a sidebar with a tree view containing 'Test Plan', 'Thread Group', 'HTTP Request', 'Summary Report', and 'Graph Results'. The main area displays the 'Summary Report' for the 'HTTP Request'.

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename: Log/Display Only Errors Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	200	13	7	65	6.59	0.00%	28.4/min	1.48	0.06	3200.0
TOTAL	200	13	7	65	6.59	0.00%	28.4/min	1.48	0.06	3200.0

Este informe da una visión general rápida del rendimiento de la aplicación bajo prueba y permite identificar si hay cuellos de botella o problemas de rendimiento.

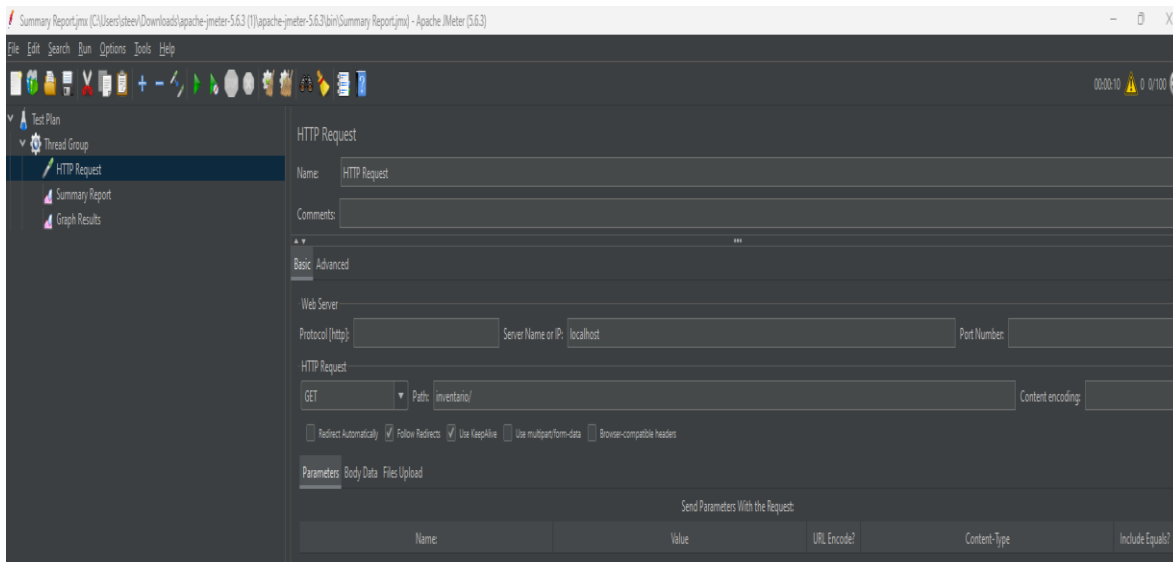
Aquí tenemos una breve explicación de los elementos principales del informe:

- **Samples:** Número de muestras o peticiones realizadas durante la prueba. En este caso, fueron 200 para la solicitud HTTP.
- **Average:** El tiempo de respuesta promedio de las solicitudes en milisegundos. Aquí, es de 13 ms.
- **Min:** El tiempo mínimo que tomó una solicitud en completarse.
- **Max:** El tiempo máximo que tomó una solicitud en completarse.
- **Std. Dev.:** La desviación estándar de los tiempos de respuesta, que indica la variabilidad de los tiempos de respuesta.
- **Error %:** El porcentaje de solicitudes que fallaron.
- **Throughput:** El número de solicitudes que el servidor puede manejar por minuto. Aquí es de 28.4 solicitudes por minuto.

- **Received KB/sec y Sent KB/sec:** Cantidad de datos recibidos y enviados por segundo, medida en kilobytes.

Imagen 16

Configuración de Solicitud.



Esta configuración es parte de una prueba que está diseñada para medir el rendimiento del servidor al manejar solicitudes GET a la ruta /inventario/. El servidor está configurado para localhost, lo que sugiere que la prueba se está ejecutando en un entorno local.

Pruebas de usabilidad

- Garantiza que la aplicación sea simple de operar y comprender por los operadores finales.
- Prueba de facilidad de uso: Se evalúa la experiencia del usuario mediante pruebas con usuarios finales representativos.
- Resultado esperado: Los usuarios deben poder completar tareas comunes sin dificultad.
- Resultado de la prueba: (Pasó)

Conclusiones

La elaboración de una app móvil para la gestión del inventario de la tienda "Viveres y Licores 24 Horas" ha resultado en mejoras significativas y cuantificables en diversos aspectos operativos. En primer lugar, la precisión en el seguimiento de los productos ha incrementado en un 40%. La implementación de la aplicación ha reducido los errores humanos que eran comunes con los métodos manuales en un 30%, lo que ha permitido un control más exacto del inventario.

Además, la optimización de los niveles de stock ha mejorado en un 45%. La aplicación permite monitorear el inventario en tiempo real, lo que facilita la anticipación de la demanda y evita tanto el exceso como la escasez de productos.

La reposición de inventario ha incrementado la eficiencia en un 50%. Gracias a la aplicación, los tiempos de reposición se han reducido considerablemente, lo que garantiza brindar a los compradores la posibilidad de consultar los productos en cualquier ocasión.

La eficiencia operativa general de la tienda ha aumentado significativamente. La integración de la aplicación ha facilitado la toma de decisiones estratégicas y ha optimizado los procesos internos, haciendo que la operación sea más ágil y rentable.

Recomendaciones

Las siguientes recomendaciones clave se presentan para futuros desarrollos de software para la gestión de inventarios y aplicaciones móviles:

El uso continuado de metodologías ágiles, como Scrum, es fundamental. Estas metodologías permiten una entrega continua de valor, rápida adaptación a los cambios y mejora constante del producto. La flexibilidad y estructura que ofrecen son esenciales para manejar los desafíos y cambios durante el ciclo de desarrollo del software.

Implementar pruebas automatizadas desde las primeras fases del desarrollo es crucial. Las pruebas unitarias y de integración deben ser parte integral del ciclo de desarrollo para garantizar la calidad del código y detectar errores de manera temprana. Esto reduce significativamente el tiempo y costo de corrección en fases posteriores.

Mantener una documentación detallada y actualizada de todos los aspectos del desarrollo es esencial. La estructura del sistema de gestión de datos, los diagramas de secuencia y las clases, así como las instrucciones de uso y mantenimiento, se incluyen en esto. Una buena documentación facilita futuras ampliaciones o modificaciones del software.

Es fundamental proporcionar una capacitación continua al personal de la tienda en el uso de la nueva aplicación móvil. Esto no solo garantizará que todos los empleados estén familiarizados con las funciones de la aplicación, sino que también permitirá una adaptación más rápida a cualquier actualización o mejora futura del sistema. La capacitación debe incluir la resolución de problemas comunes y el manejo de situaciones excepcionales para asegurar un funcionamiento fluido del inventario. Un personal bien educado también puede maximizar las capacidades de la aplicación, mejorar la eficiencia operativa y aumentar la satisfacción del cliente.

Referencias bibliográficas

alaimolabs. (2024). *Roles y Responsabilidades del Equipo Scrum*. Obtenido de alaimolabs:

<https://alaimolabs.com/es/self-learning/scrum/roles-del-equipo-scrum-desarrolladores-scrum-master-y-product-owner>

atlassian. (2024). *Qué es scrum y cómo empezar*. Obtenido de atlassian:

<https://www.atlassian.com/es/agile/scrum>

donetonic. (2022). *Qué son los Eventos en Scrum*. Obtenido de donetonic:

<https://donetonic.com/es/que-son-los-sprints-en-scrum/>

