

**Facultad de Tecnologías Libres**

Sistema de Gestión y Control de Ventas para la Tienda Ricardo & Neyde

Trabajo de diploma para optar por el título de   
Ingeniero en Ciencias Informáticas

**Autor:** Ricardo Alejandro D’Escoubet Montes de Oca

**Tutores:** MSc. Aneyty Martín García

MSc. Álvaro Alejandro Acosta Ruiz

**I**ng. Adrián Santiago Nuevo Príncipe

La Habana, noviembre de 2024

Año 66 de la Revolución

**DECLARACIÓN DE AUTORÍA**

Declaro por este medio que yo: Ricardo Alejandro D´Escoubet Montes de Oca con carné de identidad 01071372487, soy el autor principal del trabajo final de tesis de pregrado que se titula: “**Sistema de Gestión y Control de Ventas para la Tienda Ricardo & Neyde**”, el cual ha sido desarrollado como parte del trabajo en la Facultad de Tecnologías Libres. Autorizo a la Universidad de las Ciencias Informáticas a hacer uso de este en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, se firma la presente declaración jurada de autoría en La Habana, a los \_ días del mes de \_\_ del año 2024.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Est. Ricardo Alejandro D´Escoubet Montes de Oca

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

MSc. Aneyty Martín García MSc. Álvaro Alejandro Acosta Ruiz

**DATOS DE CONTACTO**

1. Aneyty Martín García graduada de Ingeniería en Ciencias Informáticas en el 2007. Máster en Educación a distancia 2016. Con experiencia en proyectos productivos, como jefa de proyecto, analista principal y probadora. Se ha desempeñado como jefa de Departamento de Ingeniería y Gestión de software, jefa de asignatura de Ingeniería de Software, jefa de Departamento de Informática. Y actualmente Vicedecana de Formación de la Facultad de Tecnologías Libres.
2. Ing. Adrián Santiago Nuevo Príncipe. Graduado en diciembre del año 2023 de la carrera Ingeniería en Ciencias Informáticas. Profesor del Departamento de Informática de la Facultad de Tecnologías Libres en la Universidad de las Ciencias Informáticas. Coordinador de cuarto año y Auxiliar Técnico de la Docencia de nivel superior.

**AGRADECIMIENTOS**

Quiero expresar nuestra profunda gratitud a todas las personas que han desempeñado un papel fundamental en la realización de este trabajo. Sus apoyos, consejos y contribuciones han sido invaluables. A mis padres quienes me han brindado su amor incondicional, apoyo emocional y sacrificio constante a lo largo de mi vida. Sus palabras de aliento y confianza han sido nuestro faro de esperanza. A mis respetados tutores Aneyty y Álvaro, quienes han compartido su sabiduría, paciencia y orientación experta. Sus valiosas sugerencias y comentarios críticos han enriquecido este trabajo. A mis profesores agradezco su dedicación a mi formación académica. A mis amigos cercanos, quienes me han apoyado en cada paso del camino. Sus palabras de ánimo y momentos compartidos han aligerado el peso de esta tarea. Gracias por estar a nuestro lado en los momentos más desafiantes. A mis compañeros de clase, quienes han compartido conocimientos, ideas y experiencias a lo largo de estos años académicos. A todas las personas que de una u otra manera contribuyeron a este proyecto, ya sea proporcionando datos, recursos o apoyo logístico, gracias, sus contribuciones han sido esenciales y altamente valoradas.

Este fragmento refleja nuestra profunda gratitud a todos los que han sido parte de este viaje. Sin su apoyo y amabilidad, este logro no sería posible. ¡Gracias de todo corazón!

**DEDICATORIA**

**A mis familiares, Por su amor inquebrantable, su apoyo incondicional y su constante aliento a lo largo de nuestra vida y principalmente en este viaje académico. Su sacrificio y dedicación han sido mi fuente de inspiración constante. A mis padres, abuela y demás familiares, por su amor y comprensión a lo largo de nuestras vidas. A ustedes, dedico este trabajo de diploma.**

A mis padres: Zuleika Montes de Oca Quintero y Ricardo D’Escoubet Nerey

A mi abuela: Loyda Quintero Rodríguez

A mi novia: Idianelis Luna Despaigne

**RESUMEN**

Este trabajo de diploma desarrolla un sistema de comercio electrónico y gestión de ventas para la Tienda Ricardo & Neyde, con el propósito de automatizar y mejorar los procesos de ventas actuales y optimizar la administración de la tienda. La solución está diseñada para simplificar la gestión del flujo de información y mejorar la eficiencia de los procesos comerciales mediante herramientas de gestión. Se utilizó la metodología AUP (Adaptative Unified Process) en su versión UCI para guiar el desarrollo del software, así como UML para el modelado y Visual Paradigm como herramienta CASE, lo que facilitó la comunicación entre el cliente y el equipo de desarrollo. Para el backend, se empleó el framework Express, mientras que React se usó en el frontend, junto con otras herramientas de código abierto. Este enfoque permitió construir una aplicación adaptable y escalable, ajustada a las necesidades de la tienda. Además, se llevaron a cabo pruebas exhaustivas para validar la funcionalidad y el rendimiento del sistema, asegurando su eficacia y fiabilidad. Como resultado, se logró un sistema que no solo agiliza los procesos de venta, sino que también mejora la gestión administrativa, ofreciendo una experiencia eficiente tanto para los usuarios internos como para los clientes de la tienda, contribuyendo a una operación más efectiva y organizada.

**PALABRAS CLAVE:** Sistema de gestión, Tienda Ricardo & Neyde, Comercio Electrónico.

***ABSTRACT***

This diploma work develops an e-commerce and sales management system for Ricardo & Neyde Store, with the aim of automating and improving current sales processes and optimizing store management. The solution is designed to simplify the management of information flow and enhance the efficiency of business processes through management tools. The AUP (Adaptive Unified Process) methodology in its UCI version was used to guide the software development, as well as UML for modeling and Visual Paradigm as a CASE tool, which facilitated communication between the client and the development team. The backend employed the Express framework, while React was used for the frontend, along with other open-source tools. This approach allowed the creation of an adaptable and scalable application tailored to the store's needs. Additionally, extensive testing was conducted to validate the system's functionality and performance, ensuring its effectiveness and reliability. As a result, a system was achieved that not only streamlines sales processes but also improves administrative management, offering efficient experience for both internal users and store customers, contributing to a more effective and organized operation.

**KEYWORDS**  
Management System, Ricardo & Neyde Store, E-commerce.

**TABLA DE CONTENIDOS**

[INTRODUCCIÓN 13](#_Toc183009480)

[CAPÍTULO I: Fundamentos teóricos de la investigación 20](#_Toc183009485)

[I.1 Definición de conceptos 20](#_Toc183009486)

[I.2 Análisis de mercado 21](#_Toc183009487)

[I.3 Metodología de Desarrollo 27](#_Toc183009488)

[1.3.1 Metodología de ingeniería de software 27](#_Toc183009490)

[I.4 Herramientas y tecnologías 29](#_Toc183009491)

[1.4.1 Herramienta CASE 29](#_Toc183009492)

[1.4.2 Lenguaje de modelado 30](#_Toc183009493)

[1.4.3 Lenguaje de programación 30](#_Toc183009496)

[1.4.4 Marco de trabajo para el desarrollo de la solución informática 31](#_Toc183009497)

[1.4.5 Entorno de desarrollo integrado 36](#_Toc183009498)

[1.4.6 Gestor de base de datos 37](#_Toc183009499)

[Conclusiones del capítulo 40](#_Toc183009500)

[CAPÍTULO II: Características y diseño del sistema 41](#_Toc183009501)

[II.1 Propuesta de solución 41](#_Toc183009502)

[II.2 Mapa Conceptual 41](#_Toc183009503)

[II.3 Requisitos de software 44](#_Toc183009504)

[2.2.1 Requisitos funcionales 44](#_Toc183009505)

[2.2.2 Requisitos No Funcionales 51](#_Toc183009506)

[Actores del Sistema 53](#_Toc183009507)

[Diagrama de Caso de Uso del Sistema 54](#_Toc183009508)

[II.4.1 Patrones de Arquitectura 55](#_Toc183009509)

[II.4.1 Patrones de Diseño 57](#_Toc183009510)

[II.4.1.1 Patrones GRASP 57](#_Toc183009511)

[II.4.1.2 Patrones GOF 60](#_Toc183009512)

[II.4.2 Diagrama de clase del diseño 62](#_Toc183009513)

[II.4.3 Diagrama de Secuencia 64](#_Toc183009514)

[II.4.4 Modelo de Datos 66](#_Toc183009515)

[II.4.5 Diagrama de Despliegue 67](#_Toc183009516)

[Conclusiones del capítulo 68](#_Toc183009517)

[CAPÍTULO III: Implementación y prueba del sistema 69](#_Toc183009518)

[III.1 Modelo de Implementación 69](#_Toc183009520)

[Diagrama de Componentes 69](#_Toc183009522)

[III.2 Pruebas de Software 70](#_Toc183009523)

[III.2.1 Pruebas unitarias 71](#_Toc183009524)

[Camino Básico 75](#_Toc183009525)

[**III.2.2 Pruebas de aceptación** 77](#_Toc183009526)

[III.3 Casos de pruebas 78](#_Toc183009527)

[Pruebas de Rendimiento 82](#_Toc183009528)

[Resumen de los Resultados 84](#_Toc183009529)

[Prueba de Usabilidad 86](#_Toc183009530)

[Prueba de Seguridad 87](#_Toc183009531)

[Conclusiones del capítulo 87](#_Toc183009532)

[CONCLUSIONES 89](#_Toc183009533)

[RECOMENDACIONES 90](#_Toc183009534)

[REFERENCIAS BIBLIOGRÁFICAS 91](#_Toc183009535)

[ANEXOS 94](#_Toc183009536)

**ÍNDICE DE TABLAS**

Tabla 1: Comparación de las funcionalidades de las aplicaciones homólogas. Fuente: elaboración propia ..................................................................................................................5

Tabla 2: Requisitos Funcionales. Fuente: Elaboración propia ……………………………...……26

Tabla 3: Actores del sistema. Fuente: Elaboración propia………………………………………35

Tabla 4: Prueba unitaria: Técnica de camino básico Fuente: Elaboración propia ……………58

Tabla 5: Diseño de caso de prueba: RF25 Crear Producto Fuente: Elaboración propia..........60

Tabla 6: Diseño de prueba de validación RF29 Crear Producto Fuente: Elaboración propia …62

**ÍNDICE DE FIGURAS**

Figura 1: Mapa conceptual…. 24

Figura 2: Diagrama de Caso de uso del sistema… 36

Figura 3: Arquitectura Modelo-Vista-Controlador… 37

Figura 4: Patrón Experto………… 39

Figura 5: Patrón Creador………39

Figura 6: Patrón Alta Cohesión……40

Figura 7: Patrón Bajo Acoplamiento…... 40

Figura 8: Patrón controlador……. 41

Figura 9: Patrón Singleton……. 42

Figura 10: Patrón Factory Method……43

Figura 11: Diagrama de Clase de Diseño del CU "Comprar Productos"…… 44

Figura 12: Diagrama de Clase de Diseño del CU "Gestionar Productos” ……45

Figura 13: Diagrama de Secuencia del caso de uso “Crear Producto” ……46

Figura 14: Diagrama de Secuencia del caso de uso “Editar Producto” ……46

Figura 15: Diagrama de Secuencia del caso de uso “Eliminar Producto” ……47

Figura 16: Modelo de Datos del Sistema……48

Figura 17: Diagrama de Despliegue……49

Figura 18: Diagrama de Componentes del CU “Gestionar Producto” ……51

Figura 19: Función de prueba……53

Figura 20: Prueba unitaria 3 fallidas y 1 pasada……53

Figura 21: Prueba Unitaria Satisfactoria……54

Figura 22: Prueba Unitaria……55

Figura 23: Representación del grafo de flujo: Técnica de camino básico……56

Figura 24: Resultado de las pruebas de aceptación……62

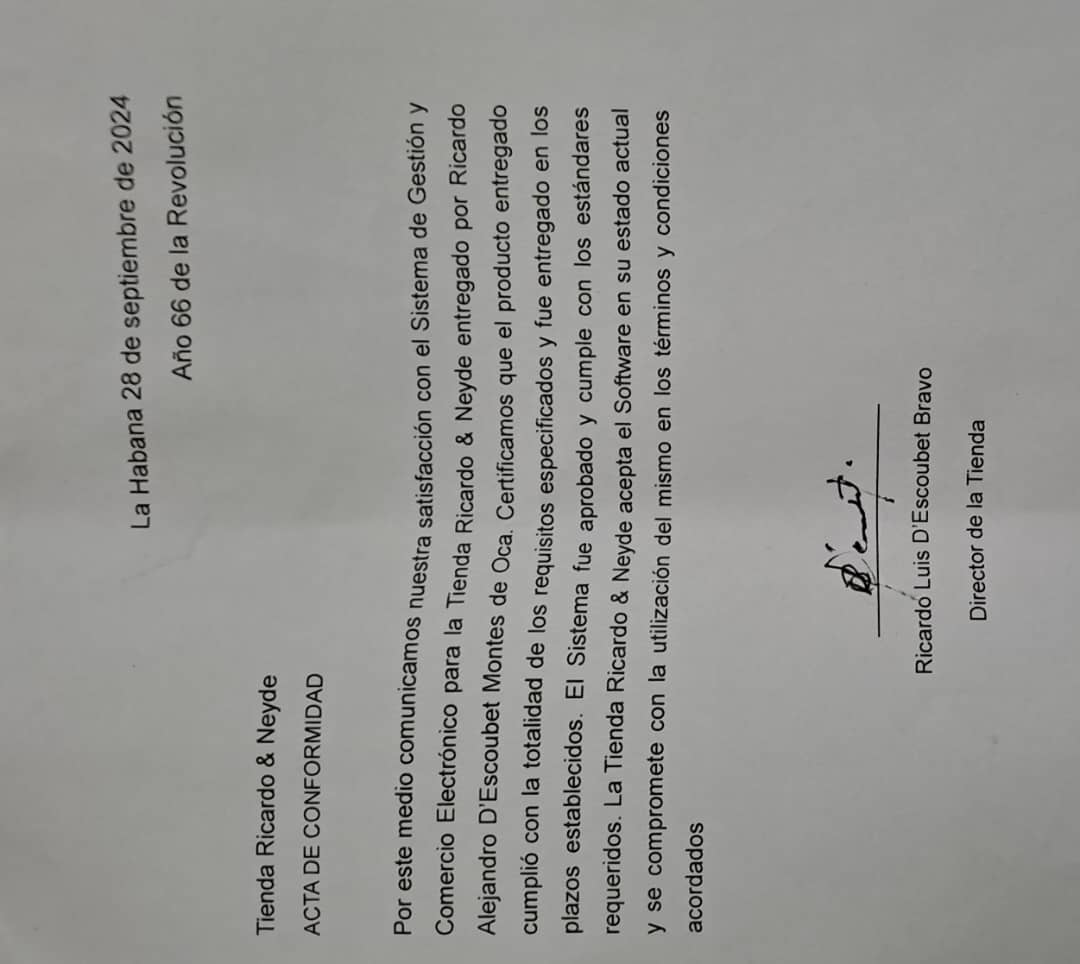
Figura 25: Resultado de las pruebas de tiempo de respuesta Apache Jmeter……64

Figura 26: Resultado de las pruebas de tiempo de respuesta Apache Jmeter……65

**OPINIÓN DEL(OS) TUTOR(ES)**

<Contenido de la opinión de los tutores>

**AVAL DEL CLIENTE**



# INTRODUCCIÓN

Las tecnologías de la información y la comunicación (TIC) han transformado radicalmente el panorama de la comunicación, el trabajo y las actividades diarias. Estas tecnologías abarcan un amplio espectro de herramientas y recursos que facilitan la creación, la difusión, el almacenamiento y la gestión de la información. En el ámbito empresarial, las TIC son fundamentales para desarrollar e implementar sistemas de gestión, que mejoran la eficiencia y la eficacia de la administración y los procesos de los recursos organizacionales.

El papel en la gestión empresarial: los sistemas de gestión basados en las TIC automatizan las tareas, mejoran la precisión de los datos y facilitan la toma de decisiones y, en última instancia, optimizan el rendimiento general de la empresa. Esta evolución tecnológica también ha tenido un impacto significativo en los entornos minoristas, donde las tiendas se han adaptado para incorporar diversas soluciones de TIC a fin de mejorar la participación de los clientes y agilizar las operaciones.

Evolución del comercio minorista: Con el tiempo, los establecimientos comerciales, como almacenes, supermercados, tiendas de descuento, centros comerciales y tiendas en línea, han evolucionado para satisfacer las necesidades cambiantes de los consumidores. Estas adaptaciones han brindado numerosos beneficios tanto a los clientes como a los propietarios de negocios, como la mejora de las experiencias de compra y la eficiencia operativa. Por ejemplo, la integración de la realidad aumentada y la inteligencia artificial en los entornos minoristas permite interacciones únicas con los clientes y experiencias de compra personalizadas, como se destaca en estudios recientes sobre las aplicaciones de las TIC en la comunicación empresarial (Kim, 2019)

Un sistema de gestión describe la forma en que las empresas se organizan en sus estructuras y procesos para actuar de forma sistemática, garantizar la fluidez de los procesos y lograr los resultados previstos Los sistemas de gestión modernos suelen seguir el ciclo PDCA de planificación, ejecución, revisión y mejora (Planificar-Hacer-Verificar-Actuar).

Un sistema de gestión eficaz se basa en procesos estructurados y optimizados y los controla. Así, establece la mejora sistemática y continua de la organización a través de reglas, roles y procesos claros.

Los sistemas de gestión pueden utilizarse en todos los ámbitos, dependiendo de dónde opere su empresa y de los objetivos que deba alcanzar. Puede ser en un sector específico, como el transporte y la logística, la industria del automóvil o la sanidad, o incluso en todos los sectores(DQS, 2024).

Implementar un **sistema de gestión de compras** puede tener una multitud de beneficios. Algunos de los beneficios más significativos incluyen:

### Aumento de la eficiencia

Un **sistema de gestión de compras** puede **simplificar** tu **proceso** de compra al **automatizar** muchas tareas que normalmente se hacen de forma manual. Esta automatización puede conducir a un **aumento de la eficiencia y la productividad**, ya que los empleados pueden dedicar más tiempo a tareas de valor añadido y menos tiempo a tareas administrativas.

### Mejora de la precisión

Los procesos manuales de compra son propensos a errores, lo que puede llevar a pérdidas financieras y otros problemas. Un **sistema de gestión de compras** puede ayudar a **mejorar** la **precisión** al **automatizar** muchas tareas y proporcionar **datos** en tiempo real para ayudar a[**identificar y corregir errores.**](https://qflowbpm.com/es/ia/automatizacion-de-procesos/)

### Mejor control y visibilidad

Un **sistema de gestión de compras** proporciona una **plataforma centralizada** para **gestionar** las actividades de **compra**, lo que puede ayudar a las empresas a obtener un **mejor control y visibilidad** sobre su proceso de compra. Esta visibilidad puede ayudar a identificar áreas en las que se pueden hacer mejoras y proporcionar información valiosa sobre el [rendimiento de los proveedores](https://qflowbpm.com/es/compras/como-elegir-el-proveedor-perfecto-de-un-sistema-de-gestion-de-compras/)**y las tendencias de compra.**

### Ahorro de costos

Implementar un **sistema de gestión de compras** puede conducir a un **ahorro de costos al reducir errores,** eliminar procesos manuales y mejorar la **gestión de proveedores**. Estos ahorros pueden acumularse con el tiempo y contribuir a la salud financiera general del negocio(Brañas, 2023).

En los últimos años, y especialmente a partir de la [pandemia de COVID-19](https://es.statista.com/temas/8697/pandemia-colapso-reseteo-y-nuevo-orden-economico-mundial/), el comercio electrónico se ha convertido en una parte indispensable del mercado minorista global. Y es que, durante meses, Internet fue el único medio a través del que muchas empresas pudieron seguir generando ingresos. Asimismo, fue la forma que los consumidores tuvieron para acceder a determinados artículos que bien por no ser considerados de primera necesidad, bien por su amplia demanda, no estaban disponibles en los establecimientos que se mantuvieron abiertos. No en vano, cerca del 90% de la población mundial admitió haber comprado en Internet en 2020, razón por la que no sorprende que los [ingresos procedentes de las ventas online](https://es.statista.com/estadisticas/1242096/facturacion-del-comercio-electronico-mundial/) se situaran en alrededor de 4,2 billones de dólares estadounidenses en dicho año. Esta cifra fue aún mayor en los tres años siguientes pese a la apertura de los comercios, lo que no hace sino dejar constancia de que este cambio en los hábitos de compra es, con casi toda seguridad, permanente(Abigail Orús & Texto, 2024).

Ricardo & Neyde, es una tienda ubicada en La Habana que se dedica a la venta presencial de varios productos como alimentos, combos de comida, entre otros**.** La gestión del inventario, tanto en la tienda como en el almacén, se realiza de manera manual.Esto implica que el administrador y los dependientes deben llevar listas físicas o anotaciones en papel para registrar los productos disponibles, vendidos y necesarios, lo que lleva a errores humanos provocando inconsistencias en los registros y falta de actualización en tiempo real, imposibilitando una visión inmediata del estado de las ventas y el inventario. Esto dificulta la toma de decisiones, además de que el proceso consume mucho tiempo, afectando la productividad del personal y la eficiencia operativa de la tienda**.**

La creación de listas de productos necesarios para abastecer la tienda se realiza manualmente después de verificar el inventario, lo que puede resultar en desabastecimientos o sobreabastecimientos, afectando la supervisión y la transparencia en las operaciones de compra y abastecimiento**.** La tienda solo es conocida por clientes locales y frecuentes debido a la ubicación intrincada que tiene, perdiendo la oportunidad de llegar a una audiencia más amplia, lo que puede llevar a la pérdida de ventas potenciales**.** En un mercado donde las ventas online son cada vez más comunes, la tienda queda en desventaja frente a competidores que ofrecen esta facilidad.

Para obtener la información necesaria para el levantamiento de requisitos, se aplicó una entrevista al cliente dueño de la tienda (Ver Anexo.1). El objetivo de la entrevista fue conocer la necesidad de la propuesta de solución y definir los requisitos funcionales y no funcionales. La entrevista fue realizada por el estudiante Ricardo Alejandro D´Escoubet Montes de Oca de la Universidad de las Ciencias Informáticas (UCI) a los siguientes empleados de la tienda Ricardo & Neyde: Ricardo Luis D´Escoubet Bravo (Administrador y jefe), Neyde Yeste Rosales (Administradora). Se arribó a las siguientes conclusiones:

* describieron que los procesos de venta, abastecimiento y control de productos son manuales, basados en listas físicas y anotaciones en papel.
* Identificaron como principales deficiencias la posibilidad de errores humanos en el registro manual, la falta de actualización en tiempo real del inventario y la imposibilidad de realizar análisis de ventas eficientes.
* Señalaron que los productos están organizados en estanterías y el inventario del almacén se gestiona de manera similar, pero sin un sistema automatizado.
* Consideraron que la creación de listas de productos necesarios para abastecer la tienda es ineficiente y puede resultar en desabastecimientos o sobre abastecimientos.
* Indicaron que los principales objetivos con la página web son mejorar la eficiencia operativa, reducir errores humanos y realizar un mejor análisis del mercado y las ventas.

De acuerdo con lo planteado anteriormente, el problema de investigación existente se definiría de la siguiente manera: ¿cómo contribuir con la gestión de los procesos de la tienda de productos alimenticios Ricardo & Neyde?

El **objeto de estudio** comprendería el proceso de gestión de tiendas.

El **campo de acción** la gestión de los productos de la tienda.

Se propone como **objetivo general** desarrollar un sistema informático con tecnología web que contribuya con la gestión de los procesos de gestión y control de ventas en la tienda Ricardo & Neyde.

De este se derivan las siguientes **preguntas científicas**:

1. ¿cuáles son los aspectos teóricos que que debe cumplir el sistema de gestión para satisfacer las exigencias del cliente?
2. ¿Qué herramientas y metodologías a utilizar para el desarrollo del sistema de gestión?
3. ¿Cómo materializar, en términos de componentes y código fuente, los elementos especificados para el sistema?
4. ¿Cómo validar el sistema de gestión para la Tienda Ricardo & Neyde?

A estas preguntas se les irá dando solución a medida que se vayan realizando las diferentes **tareas de investigación**:

1. el análisis de bibliografía referentes a soluciones anteriores dadas a casos de estudio similares al actual
2. La entrevista con el cliente para determinar los requisitos funcionales principales del sistema.
3. La selección de las herramientas y metodologías necesarias para el desarrollo de la página web.
4. El desarrollo del sistema de gestión para la venta de productos alimenticios y combos variados.
5. La planificación y ejecución de las pruebas necesarias para validar que se cumplan los requisitos funcionales y no funcionales de la página web para la tienda Ricardo & Neyde.

Los métodos teóricos que se emplearán en este trabajo son:

* histórico-lógico para estudiar la trayectoria y el uso actual de las páginas web con el fin de seleccionar la opción más apropiada.
* Analítico-sintético para realizar el estudio de documentos científicos y extraer los elementos que posibiliten la elaboración de una solución factible para el desarrollo de la página web y la modelación para materializar la representación abstracta de determinadas características del negocio y el sistema mediante la construcción de diagramas y modelos.
* Modelación consiste en la creación, mediante abstracciones, de un objeto modelado para simular procesos, visualizar datos complejos o explicar conceptos detallados de manera más comprensible.

Los métodos empíricos que se emplearán en este trabajo son:

* la entrevista, esta se le realizará al cliente con el objetivo de obtener la información necesaria para el levantamiento de requisitos (Ver Anexo.1), la observación para poder conocer cuáles son las dificultades existentes en la tienda a las que se necesita dar solución con el sistema de gestión y el análisis documental para el estudio de las propuestas a revisar durante el proceso investigativo y generar conocimiento a partir del aprendizaje reflexivo de las mismas.

El presente trabajo de investigación se encuentra estructurado en 3 capítulos:

**Capítulo 1: Fundamentos teóricos de la investigación.**

Este capítulo se centra en los aspectos teóricos y conceptos fundamentales relacionados con el **objeto de estudio** y el **campo de acción** de la investigación. Se analizan las soluciones existentes tanto en el ámbito internacional como en el nacional y se describe el entorno de desarrollo a partir de la fundamentación del uso de la metodología, tecnologías y herramientas propuestas para el desarrollo de la propuesta de solución.

**Capítulo 2: Características y diseño del sistema.**

En este capítulo se llevará a cabo la caracterización de la propuesta de solución. Se especifican los requisitos funcionales y no funcionales a partir de la metodología seleccionada, y se realiza el diseño ingenieril donde se describen los patrones de diseño definidos como buenas prácticas durante el ciclo de desarrollo del software, la realización del modelado de diagramas, los elementos fundamentales del diseño y de la arquitectura que se deben tener en cuenta para llegar propuesta solución.

**Capítulo 3: Implementación y prueba del sistema.**

En este capítulo se describe el proceso de implementación y diseño de pruebas del sistema de gestión de trabajos de diploma propuesto. Se revelan los objetivos de las pruebas, así como los criterios de aceptación y rechazo. Finalmente, se ofrece el plan de ejecución de pruebas y el análisis de los resultados obtenidos.

# CAPÍTULO I: Fundamentos teóricos de la investigación

En este capítulo, se presentan los conceptos fundamentales, teorías y modelos que sirven como base para el estudio del sistema de trabajo de diplomas. Además, se analizarán soluciones ya existentes relacionadas con el tema de investigación, sus enfoques, métodos y resultados. Su objetivo principal es proporcionar una comprensión clara y completa del estado actual del conocimiento en el área de investigación, identificar las brechas y limitaciones en los estudios previos y justificar la necesidad de realizar la investigación propuesta. También se abordarán las herramientas y metodologías a utilizar en el desarrollo de la aplicación.

## I.1 Definición de conceptos

Las tecnologías de la información y la comunicación (TIC) abarcan una amplia gama de tecnologías que facilitan la creación, el almacenamiento, el intercambio y la gestión de la información. Esto incluye herramientas como las computadoras, Internet, las telecomunicaciones y las tecnologías de radiodifusión, que en conjunto permiten una comunicación y un intercambio de información efectivos en varias plataformas(NTUU " et al., 2022).

**Sistema de Gestión:** un **sistema de gestión** es una herramienta que permite controlar, planificar, organizar y, hasta cierto punto, automatizar las tareas de una empresa. Su objetivo es unificar en un único *software* todas las operaciones de la compañía con el fin de facilitar la toma de decisiones y el análisis de los datos(Ekon, 2001).

**El Comercio Electrónico**: es el método digital para realizar actividades de marketing, que permite comprar, vender y transferir bienes y servicios en línea, ofreciendo ventajas de rentabilidad y accesibilidad(Pallavi Gladwin & Sebastian T. Joseph, 2023).

Tienda: una tienda es un lugar físico donde los productos y servicios se ofrecen a la venta a los consumidores. Sirve como un espacio tangible donde los clientes pueden experimentar y percibir las marcas(Karanikas & Öhman, 2007)**.**

Inventario: son bienes tangibles que se tienen para la venta en el curso ordinario del negocio o para ser consumidos en la producción de bienes o servicios para su posterior comercialización. Los inventarios comprenden, además de las materias primas, productos en proceso y productos terminados o mercancías para la venta, los materiales, repuestos y accesorios para ser consumidos en la producción de bienes fabricados para la venta o en la prestación de servicios; empaques y envases y los inventarios en tránsito(Sagrero, 2024).

Automatización: la automatización se refiere al uso de la tecnología para realizar tareas con una mínima intervención humana, lo que mejora la eficiencia y la precisión en varios procesos. Abarca una amplia gama de aplicaciones, desde simples dispositivos mecánicos hasta complejos sistemas de software que pueden gestionar flujos de trabajo completos(Pallavi Gladwin & Sebastian T. Joseph, 2023).

## I.2 Análisis de mercado

Antes de desarrollar un sistema de gestión, se realizó un estudio homólogo que consistió en analizar las características y funcionalidades de cinco aplicaciones web que ofrecen productos similares a los que se quieren vender.

Estos sistemas son:

Nacionales:

1. [PymesBulevar.com](https://pymesbulevar.com/?form=MG0AV3):
   * Dedicación: es un marketplace online para Cuba.
   * Quiénes son: plataforma de los emprendedores cubanos que permite a las empresas cubanas exportar sus productos y servicios tanto a nivel nacional como internacional.
2. [MarinasMarlin.com](https://marinasmarlin.com/?form=MG0AV3):
   * Dedicación: ofrece servicios de marina, excursiones marítimas, pesca deportiva, buceo y actividades náuticas en Cuba.
   * Quiénes son: empresa cubana que brinda servicios de atraques, agua, electricidad, combustible, y más.
3. [MallHabana.com](https://mallhabana.com/?form=MG0AV3):
   * Dedicación: supermercado y tienda en línea que ofrece una amplia gama de productos para el hogar, tecnología, moda, y más, con envíos a toda Cuba.
   * Quiénes son: empresa que facilita la compra y envío de productos a Cuba, además de ofrecer servicios adicionales como reservas de hoteles y cenas.

Internacionales:

1. [Amazon.com](https://amazon.com/?form=MG0AV3):
   * Dedicación: plataforma de comercio electrónico que vende una amplia variedad de productos, incluyendo moda, hogar, belleza, electrónica, juguetes, libros, y más.
   * Quiénes son: Amazon es una de las mayores empresas de comercio electrónico del mundo, fundada por Jeff Bezos en 1994.
2. [Alibaba.com](https://alibaba.com/?form=MG0AV3):
   * Dedicación: mercado global para empresas que permite a los compradores y vendedores conectarse y hacer transacciones.
   * Quiénes son: Alibaba Group, fundada por Jack Ma en 1999, es una de las mayores plataformas de comercio electrónico y comercio al por mayor en el mundo.
3. [Best.aliexpress.com](https://best.aliexpress.com/?form=MG0AV3):
   * Dedicación: subsidiaria de Alibaba que ofrece productos a precios competitivos a nivel mundial.
   * Quiénes son: AliExpress es una plataforma de comercio electrónico que permite a los compradores de todo el mundo comprar productos directamente de los fabricantes chinos.

El propósito de este estudio fue identificar las mejores prácticas, las ventajas y desventajas, y las oportunidades de mejora de cada tienda, para así poder diseñar una página web que sea competitiva, atractiva y funcional para los clientes potenciales. El estudio se basó en la observación directa de las páginas web de las tiendas.

Tabla : Comparación de las funcionalidades de las aplicaciones homólogas. Fuente: Creación propia

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Características / Funcionalidades | pymesbulevar.com | marinasmarlin.com | www.alibaba.com | mallhabana.com | amazon.com | best.aliexpress.com |
| Diseño moderno y atractivo | Sí | No | Sí | No | Sí | Sí |
| Adaptabilidad a dispositivos móviles | Sí | No | Sí | No | Sí | Sí |
| Menú superior intuitivo | Sí | No | Sí | No | Sí | Sí |
| Búsqueda avanzada de productos | No | No | Sí | No | Sí | Sí |
| Carro de compra y pago en línea | No | No | Sí | No | Sí | Sí |
| Filtrado de productos por categorías | No | No | Sí | No | Sí | Sí |
| Registro e inicio de sesión | Sí | Sí | Sí | Sí | Sí | Sí |
| Procesamiento de pagos integrados | No | No | Sí | No | Sí | Sí |
| Integración con pasarelas de pago | No | No | Sí | No | Sí | Sí |
| Monitoreo y control de inventario | No | No | Sí | Sí | Sí | Sí |
| Gestión de productos (altas/bajas) | Sí | Sí | Sí | Sí | Sí | Sí |
| Gestión de usuarios (clientes y roles) | Sí | No | Sí | Sí | Sí | Sí |
| Descuentos y promociones personalizadas | No | No | Sí | No | Sí | Sí |
| Soporte en tiempo real (chat en línea) | No | No | Sí | No | Sí | Sí |
| Multilenguaje y multimoneda | No | No | Sí | No | Sí | Sí |
| Seguimiento de envíos | No | No | Sí | No | Sí | Sí |
| Reseñas y calificaciones de productos | No | No | Sí | No | Sí | Sí |
| Integración con redes sociales | No | No | Sí | No | Sí | Sí |
| Optimización SEO para productos | No | No | Sí | No | Sí | Sí |
| Segmentación de clientes | No | No | Sí | No | Sí | Sí |
| Recomendación de productos | No | No | Sí | No | Sí | Sí |

Tras evaluar los sistemas homólogos disponibles en el mercado, se determinó que ninguno de ellos cumplía con todos los requisitos específicos necesarios para el proyecto de la Tienda Ricardo & Neyde. Algunos sistemas presentan limitaciones significativas en términos de flexibilidad, personalización y funcionalidades integradas requeridas para una gestión integral de tienda y ventas online.

La decisión de desarrollar una solución personalizada se basa en la necesidad de adaptar el sistema a las particularidades del negocio, asegurando una experiencia óptima para los usuarios y una gestión eficiente de productos, pedidos y clientes. Al construir desde cero, se pueden incorporar características específicas como el procesamiento de pedidos, el monitoreo de inventario y la gestión de productos, elementos clave que no estaban completamente cubiertos por los sistemas evaluados.

**1. Diseño Moderno y Atractivo**

* **Implementación**: en sitios como Alibaba, Amazon y AliExpress, el diseño moderno se caracteriza por el uso de imágenes de alta resolución, colores llamativos pero sobrios, y una interfaz limpia. Estos sitios suelen emplear técnicas de diseño responsivo con HTML5 y CSS3, asegurando que el diseño se mantenga consistente en diferentes dispositivos. La simplicidad en el diseño visual, combinada con elementos interactivos como sliders o banners dinámicos, contribuye a una experiencia visual agradable.

**2. Adaptabilidad a Dispositivos Móviles**

• **Implementación**: los sitios mencionados utilizan frameworks como **Bootstrap** o tecnologías nativas como **CSS Grid** y **Flexbox** para lograr una adaptabilidad total en dispositivos móviles. El uso de **media queries** y un enfoque de diseño **"mobile-first"** asegura que el contenido se reorganice y redimensione adecuadamente para pantallas pequeñas sin perder funcionalidad. En estas plataformas, los menús y botones se adaptan a formatos más simplificados en móviles para facilitar la navegación.

**3. Menú Superior Intuitivo**

* **Implementación**: el menú superior en sitios como Amazon y AliExpress está diseñado para ofrecer un acceso rápido a todas las categorías principales del sitio. Esto se logra con un menú desplegable o mega menús que agrupan categorías de manera lógica. En muchos casos, el menú incluye íconos que representan visualmente las categorías, mejorando la navegación. Se utiliza JavaScript o librerías como React para crear menús interactivos que respondan de manera eficiente.

**4. Búsqueda Avanzada de Productos**

* **Implementación**: en plataformas como Alibaba y Amazon, la búsqueda avanzada está integrada con motores de búsqueda especializados como ElasticSearch o Solr. Estos motores permiten el filtrado por diversos criterios, como precio, categoría o calificación. También se observa el uso de sugerencias automáticas y autocompletado mientras el usuario escribe, mejorando la precisión de la búsqueda. Los filtros de búsqueda son dinámicos y se actualizan en tiempo real sin necesidad de recargar la página.

**5. Carro de Compras y Pago en Línea**

* **Implementación**: en sitios como Amazon y AliExpress, el carrito de compras está siempre visible en la barra superior. Estos sitios usan una estructura de datos que permite a los usuarios agregar productos sin necesidad de recargar la página, probablemente implementado con AJAX o tecnologías de React. Las pasarelas de pago están integradas con sistemas como Stripe, PayPal o soluciones propias, garantizando la seguridad de las transacciones mediante encriptación y estándares como PCI-DSS.

**6. Filtrado de Productos por Categorías**

* **Implementación**: los sitios que implementan esta funcionalidad, como Alibaba y Amazon, usan filtros desplegables a la izquierda o en la parte superior de los resultados de búsqueda. Estos filtros se actualizan en tiempo real sin refrescar la página, utilizando JavaScript asíncrono o frameworks como Angular y React. Los filtros permiten a los usuarios seleccionar criterios como precio, marca, o disponibilidad, y los resultados se ajustan automáticamente.

**7. Soporte en Tiempo Real (Chat en Línea)**

* **Implementación**: sitios como Amazon y AliExpress utilizan sistemas de chat en línea que están integrados con IA para ofrecer respuestas automáticas a las consultas más frecuentes. También permiten la interacción directa con agentes de soporte cuando es necesario. Estos chats están integrados a través de plataformas como Zendesk, Intercom o sistemas propios de atención al cliente.

**8. Optimización SEO**

* **Implementación**: sitios como Alibaba y Amazon emplean estrategias avanzadas de SEO, donde se observa el uso adecuado de meta tags, URL amigables, y contenido bien estructurado para cada producto. Se implementan técnicas como el uso de datos estructurados (schema.org) para mejorar la visibilidad de los productos en motores de búsqueda. También se optimizan las imágenes con etiquetas alt descriptivas y se prioriza el tiempo de carga rápido del sitio para mejorar el ranking en los motores de búsqueda.

## I.3 Metodología de Desarrollo

## Las metodologías de desarrollo de software son un conjunto de técnicas y métodos organizativos que se aplican para diseñar soluciones de software informático. El objetivo de las distintas metodologías es el de intentar organizar los equipos de trabajo para que estos desarrollen las funciones de un programa de la mejor manera posible(Santander, 2020).

### 1.3.1 Metodología de ingeniería de software

Se seleccionó la Metodología AUP versión UCI para desarrollar este proyecto.

Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Se decide hacer una variación de la metodología AUP, de forma tal que se adapte al **ciclo** de vida definido para la actividad productiva de la UCI**.**

**Descripción de las fases.**

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de esta, se unifican las restantes 3 fases de AUP en una sola, a la que llamaremos Ejecución y se agrega la fase de Cierre.

**Inicio.**

Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.

**Ejecución.**

En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.

**Cierre.**

En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

**Descripción de las disciplinas.**

AUP propone 7 disciplinas (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno), se decide para el ciclo de vida de los proyectos de la UCI tener 7 disciplinas también, pero a un nivel más atómico que el definido en AUP. Los flujos de trabajos: modelado de negocio, Requisitos y Análisis y diseño en AUP están unidos en la disciplina Modelo, en la variación para la UCI se consideran a cada uno de ellos disciplinas. Se mantiene la disciplina Implementación, en el caso de Prueba se desagrega en 3 disciplinas: Pruebas Internas, de Liberación y Aceptación. Las restantes 3 disciplinas de AUP asociadas a la parte de gestión para la variación UCI se cubren con las áreas de procesos que define CMMI DEV v1.3 para el nivel 2, serían CM (Gestión de la configuración), **PP (Planeación de proyecto) y PMC (Monitoreo y control de proyecto)**(García, 2017)**.**

Se utiliza el escenario número 2 de la metodología AUP versión UCI debido a su enfoque integral y detallado en la creación de artefactos como diagramas de modelo conceptual y casos de uso del sistema. Esta metodología permite una comprensión profunda y clara de la estructura y funcionalidad del proyecto, garantizando que todos los aspectos críticos sean documentados y visualizados adecuadamente. Esto facilita la comunicación entre los miembros del equipo de desarrollo y otros interesados, asegurando que el producto final cumpla con los requisitos establecidos y optimizando el proceso de desarrollo para una implementación exitosa y eficiente.

## I.4 Herramientas y tecnologías

La elección y uso de herramientas y tecnologías adecuadas es crucial para el éxito del desarrollo de sistemas de gestión. Seleccionar cuidadosamente estas herramientas permite trabajar de manera eficiente y producir un producto final de alta calidad. Es importante investigar las opciones disponibles para encontrar las que mejor se adapten a las necesidades del proyecto, asegurando que el equipo tenga acceso a las herramientas necesarias para cumplir con los objetivos del proyecto.

### 1.4.1 Herramienta CASE

**Visual Paradigm** Visual Paradigm versión 17.0 es una herramienta CASE: ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta privada disponible en varias ediciones, cada una destinada a satisfacer diferentes necesidades: enterprise, Professional, Community, Standard, Modeler y Personal. Existe una alternativa libre y gratuita de este software, la versión Visual Paradigm UML 6.4 Community Edition (Community Edition, ya que existe la Enterprise, Professional, etc). Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos. Esta herramienta permite aumentar la calidad del software, a través de la mejora de la productividad en el desarrollo y mantenimiento del software. Aumenta el conocimiento informático de una empresa ayudando así a la búsqueda de soluciones para los requisitos. También permite la reutilización del software, portabilidad y estandarización de la documentación, además del uso de las distintas metodologías propias de la Ingeniería de Software(Pressman, 2002) .

### 1.4.2 Lenguaje de modelado

### UML versión 2.0, por sus siglas en [inglés](https://www.ecured.cu/Ingl%C3%A9s), Unified Modeling Language: es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

### Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional), pero no especifica en sí mismo qué metodología o proceso usar(Fowler & Scott, 1997).

### 1.4.3 Lenguaje de programación

HTML5 es la última versión de HyperText Markup Language (Lenguaje de Marcado de Hipertexto). Combina tres tecnologías web: HTML, para construir la estructura de la página; CSS, para mejorar la presentación; y JavaScript, para agregar funcionalidad a los elementos HTML. Sus etiquetas o marcas hacen que textos, imágenes y elementos multimedia se visualicen correctamente en la pantalla, y que las páginas web tengan un formato unificado, estandarizado y sean compatibles con distintos navegadores, como Chrome, Edge, Firefox y Safari(Saavedra, 2023) .

Mientras que HTML nos permite definir la estructura una página web, las hojas de estilo en cascada (Cascading Style Sheets o CSS) son las que nos ofrecen la posibilidad de definir las reglas y estilos de representación en diferentes dispositivos, ya sean pantallas de equipos de escritorio, portátiles, móviles, impresoras u otros dispositivos capaces de mostrar contenidos web.

Las hojas de estilo nos permiten definir de manera eficiente la representación de nuestras páginas y es uno de los conocimientos fundamentales que todo diseñador web debe manejar a la perfección para realizar su trabajo.

La primera versión de CSS fue publicada a fines del año 1996 y fue logrando popularidad y aceptación hasta llegar a la versión 2.1, estándar actual que ofrece gran compatibilidad con la mayoría de los navegadores del mercado.

A partir del año 2005 se comenzó a definir el sucesor de esta versión, al cual se lo conoce como CSS3 o Cascading Style Sheets Level 3. Actualmente en definición, esta versión nos ofrece una gran variedad de opciones muy importantes para las necesidades del diseño web actual. Desde opciones de sombreado y redondeado, hasta funciones avanzadas de movimiento y transformación, CSS3 es el estándar que dominará la web por los siguientes años(Luca, 2010).

JavaScript es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas. Desde actualizar fuentes de redes sociales a mostrar animaciones y mapas interactivos, las funciones de JavaScript pueden mejorar la experiencia del usuario de un sitio web. Como lenguaje de scripting del lado del servidor, se trata de una de las principales tecnologías de la World Wide Web. Por ejemplo, al navegar por Internet, en cualquier momento en el que vea un carrusel de imágenes, un menú desplegable “click-to-show” (clic para mostrar), o cambien de manera dinámica los elementos de color en una página web, estará viendo los efectos de JavaScript(AWS, 2023a).

### 1.4.4 Marco de trabajo para el desarrollo de la solución informática

ReactJS versión 18.3.1 es una de las librerías más populares de [JavaScript](https://www.hostinger.es/tutoriales/que-es-javascript-introduccion-basica/) para el desarrollo de aplicaciones móviles y web. Creada por Facebook, React contiene una colección de fragmentos de código JavaScript reutilizables utilizados para crear interfaces de usuario (UI) llamadas componentes.

Es importante señalar que ReactJS no es un framework de JavaScript. Esto porque sólo es responsable de renderizar los componentes de la capa de vista de una aplicación. React es una alternativa a frameworks como [Angular](https://www.hostinger.es/tutoriales/que-es-angular) y [Vue](https://vuejs.org/), que permiten crear funciones complejas.

Virtual DOM

El Modelo de Objetos del Documento (DOM) presenta una página web en una estructura de árbol de datos. ReactJS almacena árboles DOM virtuales en la memoria. Al hacerlo, React puede aplicar actualizaciones a partes específicas del árbol de datos, lo que es más rápido que volver a renderizar la totalidad del árbol DOM.

Cada vez que haya un cambio en los datos, ReactJS generará un nuevo árbol de DOM virtual y lo comparará con el anterior para encontrar la forma más rápida de implementar los cambios en el DOM real. Este proceso se conoce como diffing.

Al asegurarse de que la manipulación de la interfaz de usuario sólo afecta a secciones específicas del árbol real del DOM, renderizar la versión actualizada tarda menos tiempo y utiliza menos recursos. Esta práctica beneficia enormemente a los proyectos grandes con una interacción intensa con el usuario(A, 2020).

**Ventajas:**

**1. eficiencia y desempeño**

El DOM virtual de React y el enlace de datos unidireccional contribuyen a Rendimiento mejorado al reducir el tiempo necesario para actualizar el interfaz de usuario, un aspecto crucial para las soluciones empresariales de TI y empresas de desarrollo de software. Esta eficiencia es esencial para empresas que trabajan con aplicaciones complejas y grandes conjuntos de datos, garantizando una experiencia de usuario perfecta incluso en entornos que consumen muchos recursos escenarios.

**2. Componentes reutilizables**

La arquitectura basada en componentes de React permite el desarrollo de componentes de interfaz de usuario reutilizables, acelerando el desarrollo y garantizando coherencia en toda la aplicación. Este modularidad es particularmente ventajoso para empresas que buscan contratar ReactJS desarrolladores y empresas de TI en el extranjero que buscan una solución escalable y proceso de desarrollo eficiente.

**3. Fácil integración**

React está diseñado para integrarse fácilmente en proyectos existentes, convirtiéndolo en una excelente opción para las empresas que buscan adoptar un marco Front-end más moderno y eficiente sin la necesidad de una revisión completa del sistema. React se puede introducir gradualmente en una aplicación que permite a las empresas actualizar y mejorar sus interfaces de usuario de forma incremental, un enfoque práctico para empresas de tecnologías de la información.

**4. Escalabilidad**

A medida que las empresas crecen, también lo hacen sus requisitos de aplicación. Reaccionar Arquitectura modular y la capacidad de construir escalables. Las aplicaciones lo convierten en la opción preferida para las empresas. anticipar el crecimiento futuro, una consideración crucial para el extranjero Empresas de TI. Ya sea un pequeño startup o una gran empresa, La escalabilidad de React garantiza que las aplicaciones puedan evolucionar junto con el negocio.

**5. Apoyo y recursos comunitarios**

ReactJS cuenta con una comunidad vibrante y activa de desarrolladores, Proporcionar a las empresas una gran cantidad de recursos, incluyendo documentación, tutoriales y bibliotecas de terceros. el extenso El apoyo de la comunidad garantiza que las empresas puedan encontrar soluciones a sus problemas. desafíos rápidamente, facilitando un proceso de desarrollo más fluido para empresas de desarrollo de software(Sunil, 2024).

Node.js versión 20.14.0, es un entorno en tiempo de ejecución multiplataforma para la capa del servidor (en el lado del servidor) basado en [JavaScript](https://www.itdo.com/blog/tag/javascript/).

Node.js es un entorno *controlado por eventos* diseñado para crear aplicaciones escalables, permitiéndote establecer y gestionar múltiples conexiones al mismo tiempo. Gracias a esta característica, no tienes que preocuparte con el bloqueo de procesos, pues no hay [bloqueos](https://nodejs.org/en/docs/guides/blocking-vs-non-blocking/).

El diseño de Node.js está inspirado en sistemas como el [*Event Machine*](https://github.com/eventmachine/eventmachine) de [Ruby](https://www.ruby-lang.org/es/) o el [*Twisted*](https://twistedmatrix.com/trac/) de [Python](https://www.python.org/). Sin embargo, Node.js presenta un bucle de eventos como una construcción en tiempo de ejecución en lugar de una biblioteca. Este bucle de eventos es invisible para el usuario.

Otra característica especial de Node.js es que está diseñado para simplificar la [comunicación](https://www.itdo.com/blog/tag/comunicacion/). No tiene subprocesos, pero te permite aprovechar múltiples núcleos en su entorno y compartir [*sockets*](https://www.itdo.com/blog/3-bibliotecas-de-websocket-para-node-js-que-deberias-probar/) entre procesos(Simões, 2021).

Ventajas:

amplio ecosistema de paquetes. **Node.js cuenta con un administrador de paquetes llamado NPM (Node Package Manager), que les permite a los desarrolladores acceder a una amplia gama de módulos y bibliotecas de código abierto.**Estos paquetes pueden acelerar el proceso de desarrollo al proporcionar soluciones preconstruidas para tareas comunes. Además, la comunidad de Node.js es activa y contribuye constantemente con nuevas herramientas y actualizaciones.

**Manejo eficiente de múltiples conexiones simultáneas**

Otra de las ventajas de usar Node.js es su capacidad para manejar múltiples conexiones simultáneas de forma eficiente. **A diferencia de los servidores tradicionales basados en hilos, donde cada conexión se maneja con un hilo separado, Node.js utiliza un enfoque sin bloqueo y basado en eventos.**Esto significa que las solicitudes entrantes no bloquean el hilo principal y se manejan de manera asincrónica. Como resultado, Node.js puede manejar una gran cantidad de conexiones simultáneas con una sobrecarga mínima, lo que lo convierte en una excelente opción para aplicaciones web en tiempo real, como chats en vivo, juegos en línea o transmisiones en directo.

**Desarrollo en tiempo real**

Node.js es particularmente adecuado para el desarrollo de aplicaciones en tiempo real, como API y servicios de comunicación.**Gracias a su arquitectura basada en eventos, Node.js permite la construcción de aplicaciones altamente interactivas y receptivas**; esta es una de las ventajas de usar Node.js más destacadas. Puedes crear aplicaciones de chat, notificaciones y actualizaciones, todo ello en tiempo real y sin problemas. Esto se debe a la capacidad de Node.js para establecer conexiones bidireccionales y transmitir datos de forma eficiente entre el servidor y el cliente en tiempo real.

**Utilización de JavaScript en todo el stack**

Si ya estás familiarizado con JavaScript para el desarrollo *frontend*, una de las ventajas de usar Node.js es que **te permite aprovechar tus conocimientos existentes en el *backend*.**Al utilizar JavaScript en todo el *stack*, puedes compartir código y lógica entre el *frontend* y el *backend*, lo que facilita el desarrollo y el mantenimiento del proyecto. Esta sincronía en el lenguaje también reduce la curva de aprendizaje para los desarrolladores que ya están familiarizados con JavaScript, lo que agiliza el proceso de desarrollo y mejora la productividad del equipo.

**Flexibilidad en el alojamiento del sitio**

Otra de las ventajas de usar Node.js es que ofrece flexibilidad en el alojamiento de sitios web y aplicaciones. **Puedes optar por alojar tus aplicaciones Node.js en plataformas en la nube, como AWS (Amazon Web Services) o Heroku, o puedes configurar tu propio servidor utilizando herramientas como Nginx o Apache.** La elección del alojamiento dependerá de tus necesidades específicas y preferencias, pero Node.js te brinda la libertad de seleccionar la opción que mejor se adapte a tus proyectos(Casero, 2023).

### 1.4.5 Entorno de desarrollo integrado

Visual Studio Code versión 19.0, al que conocemos también como VSCode, es un editor de código para programadores gratuito, de [código abierto](https://www.arsys.es/blog/que-es-open-source-o-codigo-abierto) y multiplataforma. Está desarrollado por Microsoft, una compañía con una dilatada experiencia en la creación de IDEs (entornos de desarrollo integrados), que ha conseguido plasmar su larga tradición en el sector para ofrecer una herramienta ligera y práctica que la comunidad ha adoptado en masa.

Aunque su nombre viene de otra de las herramientas creadas por la propia Microsoft, IDE Visual Studio, VSCode es en realidad una aplicación independiente y desarrollada bajo una base de código distinta y en tecnologías totalmente dispares. Además, mientras que Visual Studio está preparado para desarrollar especialmente con los lenguajes y tecnologías exclusivos de Microsoft, VSCode es capaz de adaptarse a cualquier [lenguaje de programación](https://www.arsys.es/blog/lenguaje-programacion) que nos podamos imaginar.

1. **Ventajas clave de utilizar Visual Studio Code**

Al explorar las funcionalidades de Visual Studio Code, hemos dejado entrever algunas de sus ventajas, pero queremos insistir en algunos puntos:

**1. Eficiencia y agilidad en la programación**

Ante todo, VSCode es un editor ágil que funciona muy bien en incluso en equipos con recursos limitados. Además, los desarrolladores lo aprecian porque su interfaz de usuario es muy intuitiva y permite comenzar a trabajar, incluso sin conocerlo de antemano, prácticamente sin necesidad de ninguna explicación.

**2. Amplia compatibilidad con lenguajes y frameworks**

Otro de los factores de su rápida propagación es su capacidad para admitir una amplia variedad de lenguajes y frameworks, por lo que muchas comunidades lo usan en áreas tan dispares como el front-end o back-end.

En este punto también cabe destacar que es un editor multiplataforma que podemos instalar libremente en sistemas operativos Windows, Linux y MacOS.

**3. Potentes herramientas de depuración y pruebas**

Aunque no todo el mundo las usa o las necesita, VSCode viene con un conjunto de herramientas de depuración y pruebas muy potentes. Esto ayuda a debuggear los programas de manera ágil sin salirse del editor o correr pruebas unitarias de manera integrada.

**4. Integración nativa con Git y control de versiones**

Quienes están acostumbrados a usar Git por la línea de comandos quizás no lo aprecian tanto, pero lo cierto es que integración nativa con Git ha permitido que muchos desarrolladores comiencen a usar de manera sencilla este sistema de control de versiones. Además, agiliza el flujo de trabajo, ya que nos permite hacer cosas como la sincronización del código, el trabajo con ramas o la resolución de conflictos de una manera muy rápida.

**5. Personalización y extensibilidad para adaptarse a tus necesidades**

Por último, hay que destacar una vez más la cantidad de extensiones que podemos encontrar para poder ampliar las funcionalidades de VSCode allá donde podamos imaginar. De hecho, sería difícil encontrar un lenguaje o framework popular que no tenga una o decenas de extensiones para poder mejorar la integración con VSCode(García de Zúñiga, 2024).

### 1.4.6 Gestor de base de datos

[MongoDB](https://www.mongodb.com/what-is-mongodb) versión 7.0.7 es un **sistema de gestión de bases de datos** (DBMS) no relacional de código abierto que utiliza documentos flexibles en lugar de tablas y filas para procesar y almacenar diversas formas de datos.

Como solución de [base de datos NoSQL](https://www.ibm.com/mx-es/topics/nosql-databases), MongoDB **no requiere un sistema de gestión de bases de datos relacionales** (RDBMS), por lo que proporciona un modelo de almacenamiento de datos flexible que permite a los usuarios almacenar y consultar tipos de datos multivariados con facilidad. Esto no únicamente simplifica la gestión de bases de datos para los desarrolladores, sino que también crea un entorno **altamente escalable** para aplicaciones y servicios multiplataforma.

Los documentos o colecciones de documentos de MongoDB son las unidades básicas de datos. Con un formato JSON binario (notación de objetos de secuencia de comandos de Java), estos documentos pueden almacenar **varios tipos de datos y distribuirse en diferentes sistemas**. Dado que MongoDB está diseñado bajo un esquema dinámico, los usuarios tienen una flexibilidad de, al crear registros de datos, consultar colecciones de documentos a través de la agregación de MongoDB y analizar grandes cantidades de información.

MongoDB frente a MySQL

[MySQL (enlace externo a IBM)](https://www.mysql.com/) utiliza un **lenguaje de consulta estructurado** para acceder a los datos almacenados. En este formato, los esquemas se utilizan para crear estructuras de bases de datos, utilizando tablas como una forma de estandarizar los tipos de datos para que los valores se puedan buscar y consultar correctamente. **MySQL**, una solución madura, es útil para una variedad de situaciones, incluidas bases de datos de sitios web, aplicaciones y manejo de productos comerciales.

Debido a su naturaleza rígida, MySQL supera a MongoDB en relación con mantener la integridad y el aislamiento de los datos son esenciales, como cuando se gestionan datos transaccionales. Sin embargo, el **formato menos restrictivo**de MongoDB y su mayor rendimiento lo convierten en una mejor opción, especialmente cuando la disponibilidad y la velocidad son las preocupaciones principales(IBM, 2023).

1. **Ventajas:**
2. **Flexibilidad en el esquema de datos**:
3. MongoDB es una base de datos NoSQL que utiliza un modelo de documentos flexibles en lugar de un esquema rígido como en las bases de datos relacionales. En un entorno empresarial dinámico como una tienda en línea, donde los productos pueden tener atributos variables o nuevas categorías pueden surgir, la flexibilidad en el esquema de datos es crucial. Con MongoDB, podemos agregar campos a los documentos de manera dinámica sin tener que alterar la estructura de la base de datos, lo que simplifica la gestión de datos en un entorno cambiante.
4. **Escalabilidad horizontal**:
   1. mongoDB está diseñado para escalar horizontalmente de manera eficiente, lo que significa que puede manejar grandes volúmenes de datos y un aumento en el número de usuarios y transacciones sin sacrificar el rendimiento. En una tienda en línea que puede experimentar un crecimiento rápido, la capacidad de escalar horizontalmente es fundamental para garantizar un rendimiento óptimo y una experiencia de usuario sin problemas.
5. **Velocidad y rendimiento**:
   1. mongoDB ofrece un rendimiento excepcional al realizar consultas sobre grandes conjuntos de datos. Utiliza un modelo de almacenamiento en caché en memoria que acelera el acceso a los datos y es especialmente efectivo para operaciones de lectura intensivas como las consultas de productos en una tienda en línea. Además, la indexación eficiente y las consultas flexibles permiten obtener resultados rápidos incluso en conjuntos de datos complejos.
6. **Integración con Node.js y Express.js**:
   1. mongoDB se integra perfectamente con Node.js y Express.js a través de la librería Mongoose, lo que facilita el desarrollo de aplicaciones web modernas y escalables. Mongoose proporciona una capa de abstracción sobre MongoDB que simplifica la interacción con la base de datos y ofrece características adicionales como validación de datos, esquemas de documentos y relaciones entre colecciones. Esta integración sin problemas permite a los desarrolladores aprovechar las ventajas de MongoDB mientras trabajan en un entorno de desarrollo JavaScript unificado.
7. **Facilidad de uso y administración**:

mongoDB es conocido por su facilidad de uso y administración. Su modelo de documentos y su sintaxis intuitiva de consulta hacen que sea fácil para los desarrolladores trabajar con la base de datos y realizar operaciones CRUD. Además, MongoDB ofrece herramientas de administración robustas que simplifican tareas como la configuración, monitorización y escalado de la base de datos, lo que reduce la carga operativa para el equipo de desarrollo(MongoDB Documentation, 2024).

## Conclusiones del capítulo

En este capítulo se analizaron varios conceptos relacionados con el desarrollo web, el comercio electrónico y la ingeniería de software. Además, se vieron otras soluciones informáticas al problema planteado realizadas en el país y las herramientas utilizadas en cada caso. Esto permitió a los autores arribar a las siguientes conclusiones:

* A pesar del notable crecimiento del comercio electrónico en Cuba, el país aún presenta una cultura limitada en este ámbito. Por ello, iniciativas como la propuesta en este trabajo son fundamentales para que la sociedad se adapte progresivamente a las nuevas tecnologías y avance tecnológicamente.
* Antes de desarrollar la Tienda Ricardo & Neyde, se llevó a cabo un estudio comparativo de seis aplicaciones web similares en el mercado: pymesbulevar.com, marinasmarlin.com, mallhabana.com, [amazon.com](http://amazon.com)**,** alibaba.com, best.aliexpress.com
* Cada una ofrece diversas funcionalidades, desde la venta de productos en moneda libremente convertible hasta soluciones integrales para la gestión de tiendas virtuales. El análisis se centró en identificar las mejores prácticas, así como las ventajas y desventajas, para guiar el diseño de una página web competitiva y funcional.
* La elección de herramientas y tecnologías para el desarrollo de la Tienda Ricardo & Neyde se basó en una investigación exhaustiva. Se optó por React.js para una interfaz de usuario dinámica y Node.js con Express.js para el Backend, asegurando un rendimiento óptimo y escalabilidad. Además, se utilizará Visual Paradigm para el modelado y UML para la documentación. Estas decisiones garantizan un desarrollo eficiente y la capacidad de adaptarse a las necesidades específicas del negocio y del mercado.

# CAPÍTULO II: Características y diseño del sistema

En este capítulo se presentará el diseño de la solución propuesta al problema científico planteado en la introducción. Para ello se registrarán las reglas del negocio, se explicarán los principales elementos, los modelos conceptuales, así como los requisitos funcionales y no funcionales de la solución propuesta; además de los posibles resultados esperados. Partiendo de lo anteriormente descrito, se modela el mecanismo quedando representado en un diagrama de casos de uso del sistema y sus descripciones textuales. Además, se modelan los artefactos definidos por la metodología AUP versión UCI y las posibles mejoras de la solución propuesta.

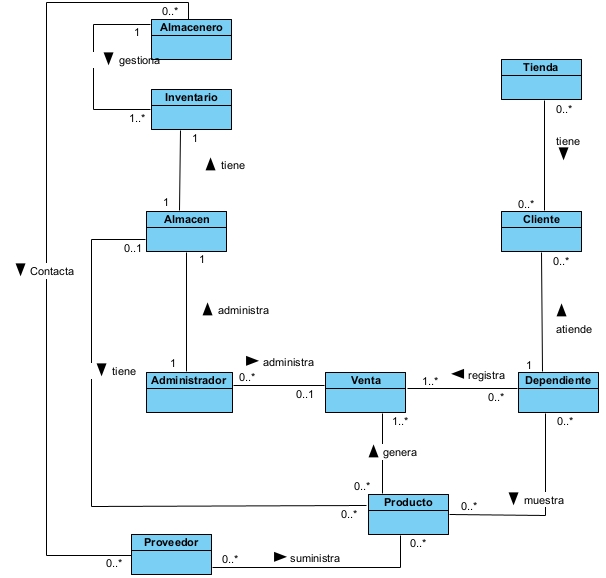
## II.1 Propuesta de solución

Para resolver los problemas de gestión y control en la Tienda Ricardo & Neyde, se desarrollará un sistema informatizado integral. Este sistema permitirá la autenticación de usuarios, garantizando la seguridad y control de acceso. Esto permitirá que solo los usuarios autorizados realicen tareas específicas según sus roles, mejorando así la experiencia del usuario. Además, se llevará a cabo la gestión de productos de la tienda y de la cafetería, facilitando la creación, modificación, búsqueda, visualización y eliminación de productos, así como la gestión de su cantidad, precio, imagen y descripción. Esto asegurará una gestión precisa y actualizada del inventario.

El sistema también incluirá la gestión de inventarios, optimizando el control del stock y proporcionando información en tiempo real para la toma de decisiones estratégicas sobre compras y ventas. La gestión de ventas permitirá la creación, visualización y eliminación de ventas, mejorando la eficiencia operativa mediante el seguimiento y control de transacciones.

## II.2 Mapa Conceptual

Habiendo seleccionado la metodología de desarrollo de software AUP-UCI en su escenario 2, se hace necesario que se modele el modelo conceptual (MC). Es una herramienta que posibilita organizar y representar, de manera gráfica y mediante un esquema, el conocimiento. Explica a los creadores los conceptos significativos en un dominio del problema, es el artefacto más importante para crear durante el análisis orientado a objetos. Una cualidad esencial que debe ofrecer un modelo conceptual es que representa cosas del mundo real, no componentes del software. Es un diagrama estático donde no se define ninguna operación. El modelo conceptual puede mostrar: conceptos, asociaciones entre ellos y atributos de conceptos.



***Figura 1: Mapa conceptual.***

En la Figura 2 queda plasmado el modelo conceptual que será utilizado para el desarrollo del Sistema. La Tienda tiene Clientes, los cuales son atendidos por un Dependiente el cual muestra los Productos y registra la Venta que este genera. El Administrador administra las ventas y el almacén, el cual tiene un inventario que se gestiona por mano del Almacenero, este último contacta al Proveedor que es el encargado de suministrar los Productos.

Conceptos:

**Tienda**: un establecimiento comercial donde se ofrecen productos o servicios para la venta. Las tiendas pueden variar en tamaño y especialización, desde pequeñas tiendas de conveniencia hasta grandes almacenes.

**Almacenero**: persona encargada de la gestión del almacén de una compañía. Sus funciones incluyen la recepción, clasificación, almacenamiento y distribución de mercancías.

**Inventario**: registro detallado de los bienes muebles e inmuebles que forman parte del caudal comercial de una persona o empresa. Se utiliza para llevar un control exhaustivo de la mercadería y las ventas.

**Almacén**: espacio físico donde se depositan o guardan productos o mercancías de forma temporal mientras se espera su distribución. Los almacenes son esenciales para la logística y la gestión de inventarios.

**Administrador**: persona encargada de gestionar y coordinar las actividades dentro de una empresa, asegurando que los recursos disponibles se utilicen de manera eficiente para alcanzar los objetivos establecidos.

**Proveedor:** persona física o jurídica que abastece a una empresa de los insumos y servicios que requiere para empezar o continuar sus labores productivas.

**Producto**: bien o servicio obtenido al final de la cadena productiva, a partir de la transformación de materias primas en bienes elaborados o servicios que prestar.

**Cliente**: persona o entidad que compra los bienes y servicios que ofrece una empresa. Los clientes son esenciales para el éxito de un negocio.

**Dependiente**: trabajador cuya función es atender a las personas que llegan a una tienda, ayudándoles a encontrar y adquirir los productos que buscan.

**Venta**: proceso de intercambio mediante el cual una parte, el vendedor, satisface la necesidad de otra, el comprador, a cambio de una suma de dinero o crédito.

## II.3 Requisitos de software

Los requisitos de software son descripciones detalladas de los servicios que debe proporcionar un sistema de software y las restricciones bajo las que debe funcionar. Reflejan las necesidades y expectativas de los clientes en cuanto a un software que sirva para fines específicos, como controlar dispositivos, procesar pedidos o recuperar información. Estos requisitos se pueden clasificar en dos tipos principales: requisitos funcionales, que describen las acciones y funcionalidades específicas que debe realizar el software, y requisitos no funcionales, que definen atributos de calidad como el rendimiento, la seguridad y la usabilidad

(Shah & Salles, 2018).

### 2.2.1 Requisitos funcionales

Los requisitos funcionales son declaraciones específicas que definen los comportamientos, las funciones y las capacidades que debe exhibir un sistema de software. Describen lo que debe hacer el sistema y describen las tareas que debe realizar para satisfacer las necesidades de los usuarios y los objetivos empresariales(Shah & Salles, 2018).

**Tabla 2: Requisitos Funcionales. Fuente: Creación propia**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nombre** | **Descripción** | **Prioridad** | **Complejidad** |
| **RF1** | **Crear usuario** | Permite crear un nuevo usuario en el sistema solicitando su nombre de usuario, contraseña, correo electrónico y número de teléfono. | **Alta** | **Media** |
| **RF2** | Autenticar usuario | Permite verificar la identidad del usuario que desea acceder al sistema, solicitando su nombre de usuario y contraseña. | Alta | Baja |
| **RF3** | Iniciar sesión con Google | Permite iniciar sesión en el sistema utilizando una API de Google para verificar la identidad con su correo electrónico y su contraseña | Media | Alta |
| **RF4** | Registrar usuario | Permite registrar un nuevo usuario en el sistema solicitando su nombre de usuario, contraseña, correo electrónico y número de teléfono. | Alta | Media |
| **RF5** | Buscar usuario. | Permite buscar un usuario en el sistema, introduciendo su nombre o correo electrónico. | Baja | Baja |
| **RF6** | Cambiar contraseña. | Permite cambiar la contraseña a cualquier usuario solicitando su correo electrónico. | Media | Baja |
| **RF7** | Modificar usuario. | Permite modificar los datos de un usuario existente en el sistema. | Media | Media |
| **RF8** | Mostrar usuario. | Permite mostrar todos los usuarios existentes en el sistema. | Media | Baja |
| **RF9** | Eliminar usuario. | Permite eliminar los usuarios existentes en el sistema. | Media | Baja |
| **RF10** | Activar usuario. | Permite activar a los usuarios existentes en el sistema. | Media | Baja |
| **RF11** | Desactivar usuario. | Permite desactivar a los usuarios existentes en el sistema. | Media | Baja |
| **RF12** | Filtrar por usuarios activos o desactivados. | Permite al administrador filtrar por usuarios activos en el sistema o usuarios desactivados. | Baja | Baja |
| **RF13** | Editar perfil | Permite editar el perfil nombre de su cuenta. | Media | Media |
| **RF14** | Cerrar sesión | Permite cerrar sesión a los usuarios existentes en el sistema. | Baja | Baja |
| **RF15** | Notificar la desactivación de un usuario | Notifica al usuario administrador utilizando una alerta UI la acción realizada con éxito | Baja | Baja |
| **RF16** | Notificar la eliminación de un usuario | Notifica al usuario administrador utilizando una alerta UI la acción realizada con éxito | Baja | Baja |
| **RF17** | Asignar rol | Permite asignar un rol a un usuario especifico |  |  |
| **RF18** | Cambiar rol asignado | Permite cambiar el rol de un usuario con rol asignado |  |  |
| **RF19** | Mostrar producto de la Tienda | Permite mostrar todos los productos de la tienda existentes en el sistema. | Alta | Baja |
| **RF20** | Modificar producto de la Tienda | Permite modificar los datos de un producto existente en el sistema. | Alta | Alta |
| **RF21** | Buscar producto de la Tienda | Permite a un administrador buscar un producto en el sistema, introduciendo su nombre. | Alta | Baja |
| **RF22** | Buscar producto en la Tienda | Permite a un usuario buscar un producto de la tienda por su nombre. |  |  |
| **RF23** | Crear producto de la Tienda | Permite registrar un nuevo producto en el sistema. | Alta | Alta |
| **RF24** | Eliminar producto de la Tienda | Permite eliminar un producto existente en el sistema. | Baja | Baja |
| **RF25** | Notificar nuevo producto de la tienda añadido |  |  |  |
| **RF26** | Notificar producto de la tienda eliminado |  |  |  |
| **RF27** | Notificar producto de la tienda editado |  |  |  |
| **RF28** | Filtrar producto de la cafetería por categoría | Permite filtrar los productos de la cafetería por su categoría | Baja | Baja |
| **RF29** | Mostrar producto de la Cafetería |  |  |  |
| **RF30** | Modificar producto de la Cafetería |  |  |  |
| **RF31** | Buscar producto en la Cafetería | Permite al usuario buscar un producto de la cafetería por su nombre |  |  |
| **RF32** | Crear producto de la Cafetería |  |  |  |
| **RF33** | Eliminar producto de la Cafetería |  |  |  |
| **RF34** | Notificar nuevo producto de la Cafetería añadido |  |  |  |
| **RF35** | Notificar producto de la Cafetería eliminado |  |  |  |
| **RF36** | Notificar producto de la Cafetería editado |  |  |  |
| **RF37** | Mostrar combos |  |  |  |
| **RF38** | Modificar combo |  |  |  |
| **RF39** | Crear combo |  |  |  |
| **RF40** | Eliminar combo |  |  |  |
| **RF41** | Notificar nuevo combo añadido |  |  |  |
| **RF42** | Notificar combo eliminado |  |  |  |
| **RF43** | Notificar combo editado |  |  |  |
| **RF44** | Crear venta | Permite crear una nueva venta en el sistema después de seleccionar los productos que se desean comprar. | Alta | Alta |
| **RF45** | Listar ventas | Muestra todas las ventas realizadas hasta el momento |  |  |
| **RF46** | Filtrar venta por día | Muestra las ventas realizadas el día seleccionado |  |  |
| **RF47** | Agregar producto al carrito | Permite agregar un producto al carrito de compra para proceder luego con el pago |  |  |
| **RF48** | Quitar producto del carrito | Permite eliminar un producto del carrito de compra, de no ser deseado |  |  |
| **RF49** | Actualizar producto en el carrito | Permite aumentar o disminuir la cantidad de un producto ya subido al carrito de compra |  |  |  |
| **RF50** | Compartir enlace de la web | Permite compartir el enlace de la pagina |  |  |  |
| **RF51** | Contactar a la administración a través de las redes sociales | Permite contactar a la administración a través de WhatsApp, Facebook o Instagram. |  |  |  |
| **RF52** | Pago en línea por tarjeta | Permite al usuario pagar su orden en línea con una tarjeta magnética |  |  |  |
| **RF53** | Eliminar Venta |  |  |  |  |
| **RF54** | Notificar carrito de compra vacío |  |  |  |

### 2.2.2 Requisitos No Funcionales

Los requisitos no funcionales definen las cualidades y características del producto, orientadas a mejorar su rendimiento, seguridad, usabilidad y mantenibilidad.

**Usabilidad**

* **RnF-1 Interfaz Intuitiva:** La interfaz debe ser fácil de usar, con navegación clara y botones visibles. Los usuarios deben poder encontrar rápidamente funcionalidades principales como registrar, modificar o buscar usuarios y productos. *(Se verifica mediante lista de chequeo de usabilidad)*
* **RnF-2 Responsividad:** La aplicación debe adaptarse a diferentes tamaños de pantalla y dispositivos (móviles, tabletas, desktops) para asegurar una experiencia de usuario consistente. *(Se puede validar mediante pruebas de Lighthouse)*
* **RnF-3 Feedback en Tiempo Real:** La aplicación debe proporcionar retroalimentación inmediata al usuario cuando se realicen acciones como el registro de usuario, cambios en el perfil o la adición de productos al carrito, mejorando la experiencia de uso. *(Evaluado mediante lista de chequeo de usabilidad)*
* **RnF-4 Optimización de Búsqueda:** Las funciones de filtro y búsqueda deben ser eficientes y rápidas, incluso con grandes volúmenes de datos, mejorando la experiencia del usuario en la navegación. *(Este aspecto puede ser evaluado en pruebas de rendimiento en Lighthouse y JMeter)*

**Rendimiento**

* **RnF-5 Rendimiento de la Aplicación:** La aplicación debe mantener un rendimiento óptimo, con tiempos de respuesta rápidos en operaciones clave, y alcanzar una puntuación mínima de 60 en la métrica de rendimiento de Lighthouse. *(Se verifica mediante pruebas de rendimiento en Lighthouse y carga con Apache JMeter)*
* **RnF-6 Escalabilidad:** La aplicación debe ser capaz de manejar un aumento en la cantidad de usuarios y datos sin degradar el rendimiento, manteniendo un tiempo de respuesta adecuado con hasta 3000 solicitudes en rutas clave. *(Se evalúa con pruebas de carga en Apache JMeter)*

**Seguridad**

* **RnF-7 Seguridad Robusta:** Implementar medidas de seguridad sólidas, incluyendo encriptación de datos sensibles, autenticación segura (como OAuth para Google), y protección contra intentos de acceso no autorizados. *(Validado mediante lista de chequeo de seguridad)*
* **RnF-8 Protección de Datos:** La aplicación debe garantizar la protección de datos sensibles, evitando el acceso no autorizado a recursos protegidos y protegiendo la privacidad de los usuarios. *(Validado mediante lista de chequeo de seguridad)*

Mantenibilidad y Compatibilidad

* **RnF-9 Mantenibilidad:** El código debe estar bien estructurado y documentado para facilitar futuras modificaciones y mantenimiento. Se deben utilizar herramientas de linting y pruebas automatizadas para asegurar la calidad del código y prevenir errores en cambios futuros. *(Validado mediante pruebas unitarias y estructura de camino básico)*
* **RnF-10 Compatibilidad:** La aplicación debe funcionar correctamente en los navegadores más comunes:
  + Google Chrome: Versión 104 o superior
  + Mozilla Firefox: Versión 94 o superior
  + Apple Safari: Versión 15 o superior
  + Microsoft Edge: Versión 104 o superior *(Verificado a través de pruebas de compatibilidad en distintos navegadores)*

**Interfaz**

* **RnF-11 Interfaz de Usuario Clara:** Las interfaces de usuario deben estar diseñadas de manera que el usuario pueda identificar fácilmente dónde está y qué acción está realizando, y debe mostrar la información de manera clara. *(Validado mediante lista de chequeo de usabilidad)*

### Actores del Sistema

En ingeniería de software, un actor del sistema es una entidad que interactúa con un sistema para lograr objetivos específicos, que pueden incluir usuarios humanos, otros sistemas o componentes que envían y reciben mensajes. Los actores son cruciales para definir los requisitos y las funcionalidades del sistema, y pueden clasificarse en actores humanos, como los clientes, y actores no humanos, como las API(Shah & Salles, 2018).

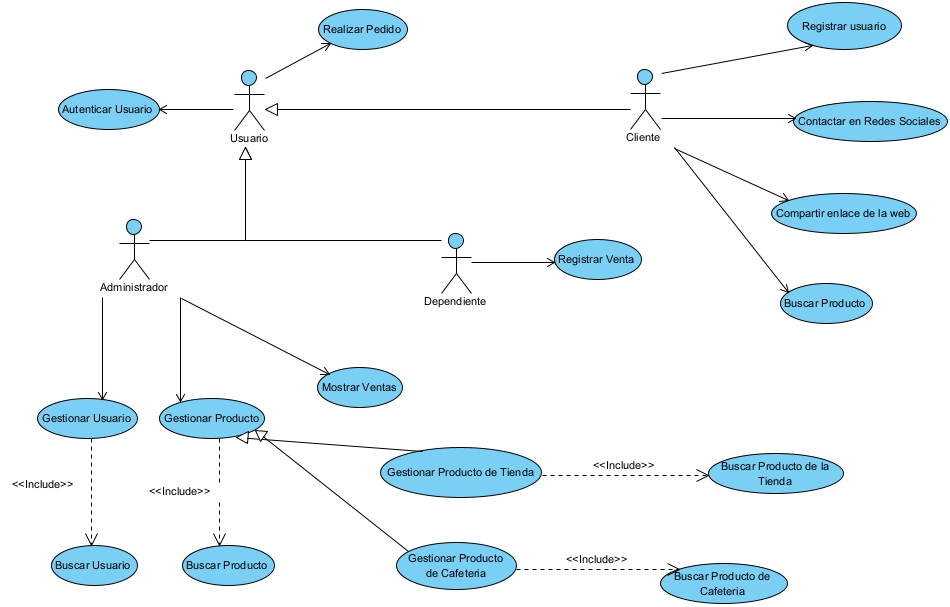
|  |  |
| --- | --- |
| Actores | Descripción |
| Dependiente | El Dependiente tiene la responsabilidad de atender al cliente en su pedido y registrar la venta. |
| Administrador | El Administrador tiene la responsabilidad de garantizar el acceso al sistema, a través de la gestión de usuarios, roles y permisos. Además, tiene acceso a todas las funcionalidades del sistema sin restricción alguna. |

**Tabla 3: Actores del sistema. Fuente: Creación propia**

### Diagrama de Caso de Uso del Sistema

Siguiendo lo establecido por la metodología AUP-UCI en su escenario 2, esta plantea que Proyectos que modelen el negocio con modelo conceptual solo pueden modelar el sistema con Casos de Uso del Sistema.

El propósito principal de un diagrama de casos de uso es identificar y clarificar los requisitos funcionales del sistema. En otras palabras, se centra en "qué" hará el sistema desde la perspectiva del usuario, en lugar de "cómo" lo hará. Los actores en el diagrama representan a los usuarios o entidades externas que interactúan con el sistema, mientras que los casos de uso son las acciones o funciones específicas que el sistema puede realizar(Nuria, 2024).

**

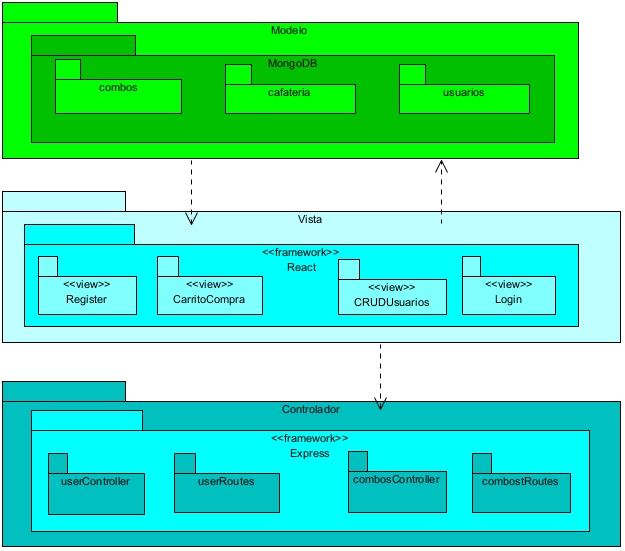
***Figura 2: Diagrama de Caso de uso del sistema***

En figura anterior se muestra el diagrama de caso de uso del Sistema, donde se ven reflejado los casos de usos por lo que están compuesto el Sistema y su relación con los actores. Los actores Administrador y Dependiente heredan del actor Usuario, el cual tiene el caso de uso autenticarse y Realizar Pedido. El actor Administrador tiene los casos de usos Gestión de Usuario, Mostrar Venta y Gestión de Producto al cual se le incluye el caso de uso Gestión de Producto de la Tienda y Gestión de Producto de la Cafetería. El actor Dependiente tiene el caso de uso Registrar Venta. El actor Cliente tiene los casos de uso Compartir enlace de la web, Buscar Producto, Contactar en Redes Sociales y Registrar Usuario.

## II.4.1 Patrones de Arquitectura

En términos generales, un patrón de arquitectura es la conceptualización de una solución genérica y reutilizable, aplicable a un problema de diseño de software en un contexto determinado, satisfaciendo las necesidades del negocio.

Los patrones arquitectónicos son la representación de las buenas prácticas y estructuras de diseño probadas, de modo que puedan reutilizarse. El objetivo es reutilizar las experiencias y conocimientos arquitectónicos que han dado buenos resultados en el pasado(Castro Ramirez, 2023).



***Figura 3: Arquitectura Modelo-Vista-Controlador.***

Este patrón, también conocido como patrón MVC, divide una aplicación interactiva en tres (3) partes como:

* Modelo: contiene la funcionalidad central y los datos.
* Vista: muestra la información al usuario (se puede definir más de una vista).
* Controlador: maneja las entradas del usuario.

Esto se hace para separar las representaciones internas de la información de las formas en que la información se presenta al usuario y es aceptada por éste. Desacopla los componentes y permite una reutilización suficiente del código(Castro Ramirez, 2023).

## II.4.1 Patrones de Diseño

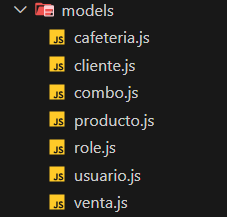
Los patrones de diseño son soluciones reutilizables para problemas comunes de diseño de software, que sirven como plantillas para abordar problemas específicos del desarrollo. Para que una solución pueda considerarse un patrón de diseño, debe incluir un nombre claro, definir el problema que aborda, proporcionar una solución general y analizar las consecuencias de su uso. Los patrones de diseño se clasifican en tipos creacionales, estructurales y conductuales, cada uno de los cuales se centra en diferentes aspectos de la creación, la composición y la interacción de los objetos. Mejoran la flexibilidad, la reutilización y la capacidad de mantenimiento del código, lo que facilita una mejor comunicación entre los desarrolladores(Santos et al., 2021).

Para el desarrollo de la propuesta de solución que se describe en la investigación solo fueron usados los implementados por el Framework, por tal motivo serán los descritos a continuación.

### II.4.1.1 Patrones GRASP

Los estándares GRASP (Patrones de software de asignación de responsabilidades generales) son un conjunto de principios que guían la asignación de responsabilidades a las clases y objetos en el diseño orientado a objetos. Estos principios ayudan a los desarrolladores a crear sistemas que sean más fáciles de mantener, flexibles y comprensibles.

**Experto:** asigne la responsabilidad a la clase que tenga la información necesaria para cumplirla.

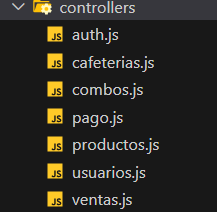
****

***Figura 4: Patrón Experto***

**Creador:** asigna la responsabilidad de crear una instancia de una clase a la clase que la agrega o usa.

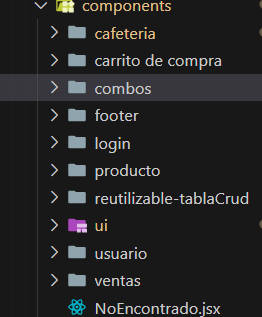
Es utilizado en los controladores, en ellos se encuentran las acciones definidas para el sistema. En la implementación de las acciones se crean instancias de las clases del modelo. En Express en Controller están ubicadas los archivos .js en el cual se definen y se ejecutan las acciones.

**//controllers/…**



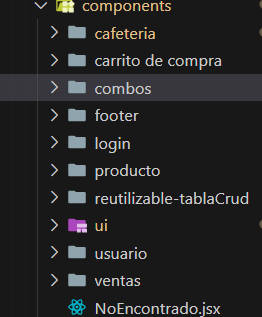
***Figura 5: Patrón Creador***

**Alta cohesión:** asegúrese de que una clase tenga un propósito bien definido y de que sus responsabilidades estén estrechamente relacionadas**.**

****

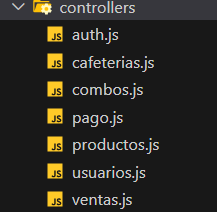
***Figura 6: Patrón Alta Cohesión***

**Bajo acoplamiento:** apunte a un acoplamiento bajo entre clases para reducir las dependencias y aumentar la flexibilidad.

****

***Figura 7: Patrón Bajo Acoplamiento.***

**Controlador:** asigna la responsabilidad de gestionar los eventos del sistema a una clase de controlador, que coordina el flujo de la aplicación(Santos et al., 2021)**.**

****

***Figura 8: Patrón* *controlador.***

### II.4.1.2 Patrones GOF

Los patrones GOF, o Gang of Four, son una colección de 23 patrones de diseño que proporcionan soluciones estandarizadas a problemas comunes de diseño de software. Presentados en el influyente libro «Patrones de diseño: elementos del software reutilizable orientado a objetos», estos patrones ayudan a los desarrolladores a crear sistemas más escalables y fáciles de mantener. Se clasifican en tres tipos principales: patrones creacionales, que se centran en la creación de objetos (por ejemplo, Singleton y Factory Method); patrones estructurales, que se ocupan de la composición de clases y objetos (por ejemplo, Adapter y Decorator); y patrones de comportamiento, que abordan la interacción y la responsabilidad entre objetos (por ejemplo, Observer y Strategy). Al aplicar los patrones GOF, los desarrolladores pueden aprovechar soluciones comprobadas para mejorar la calidad del código y facilitar una mejor arquitectura de software(Santos et al., 2021)**.**

**1-Patrón Singleton**

****

***Figura 9: Patrón* *Singleton.***

El patrón Singleton asegura que solo exista una única instancia de una clase u objeto, que puede ser compartida entre diferentes partes de la aplicación. En tu código, el cliente de consultas useQueryClient () actúa como un Singleton al mantener una instancia compartida del cliente de consultas que se usa para gestionar la caché de datos.

Ejemplo: queryClient.invalidateQueries() reutiliza la misma instancia del cliente de consultas para evitar múltiples instancias en diferentes partes del componente.

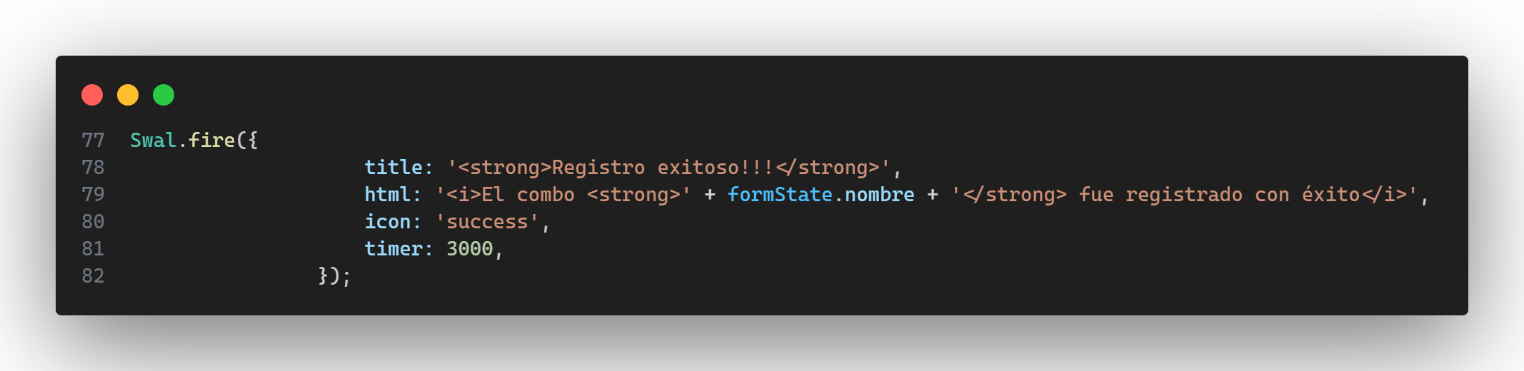
2-Patrón Command

addCombos(), updateCombos(), deleteCombos()

El patrón Command encapsula una petición como un objeto, permitiendo parametrizar clientes con diferentes solicitudes, poner en cola solicitudes o registrarlas. En este caso, los métodos addCombos(), updateCombos(), y deleteCombos() encapsulan las operaciones que se ejecutarán al interactuar con la API. Estas funciones contienen la lógica de qué hacer y cuándo ejecutar la acción.

**Ejemplo:** cada vez que se llama a addCombos() o updateCombos(), se ejecuta una "comanda" que desencadena una acción específica, como enviar una solicitud POST o PUT

3**. Patrón Factory Method**

****

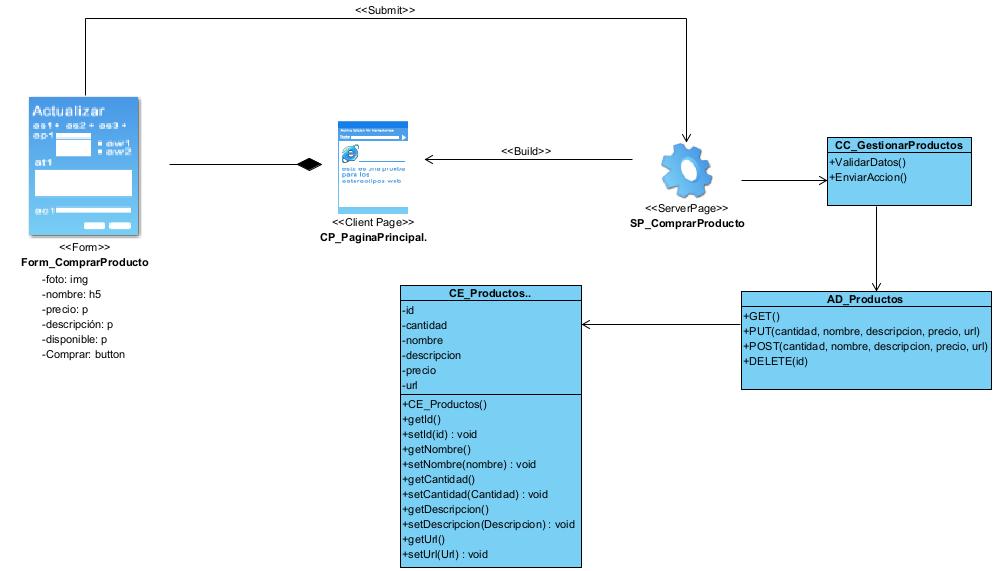
***Figura 10: Patrón* Factory Method*.***

El patrón Factory Method define una interfaz para crear un objeto, pero deja que las subclases decidan qué clase instanciar. Swal.fire() es un ejemplo de un Factory Method, ya que crea diferentes tipos de alertas (éxito, error, advertencia) en función de los parámetros que se le pasen.

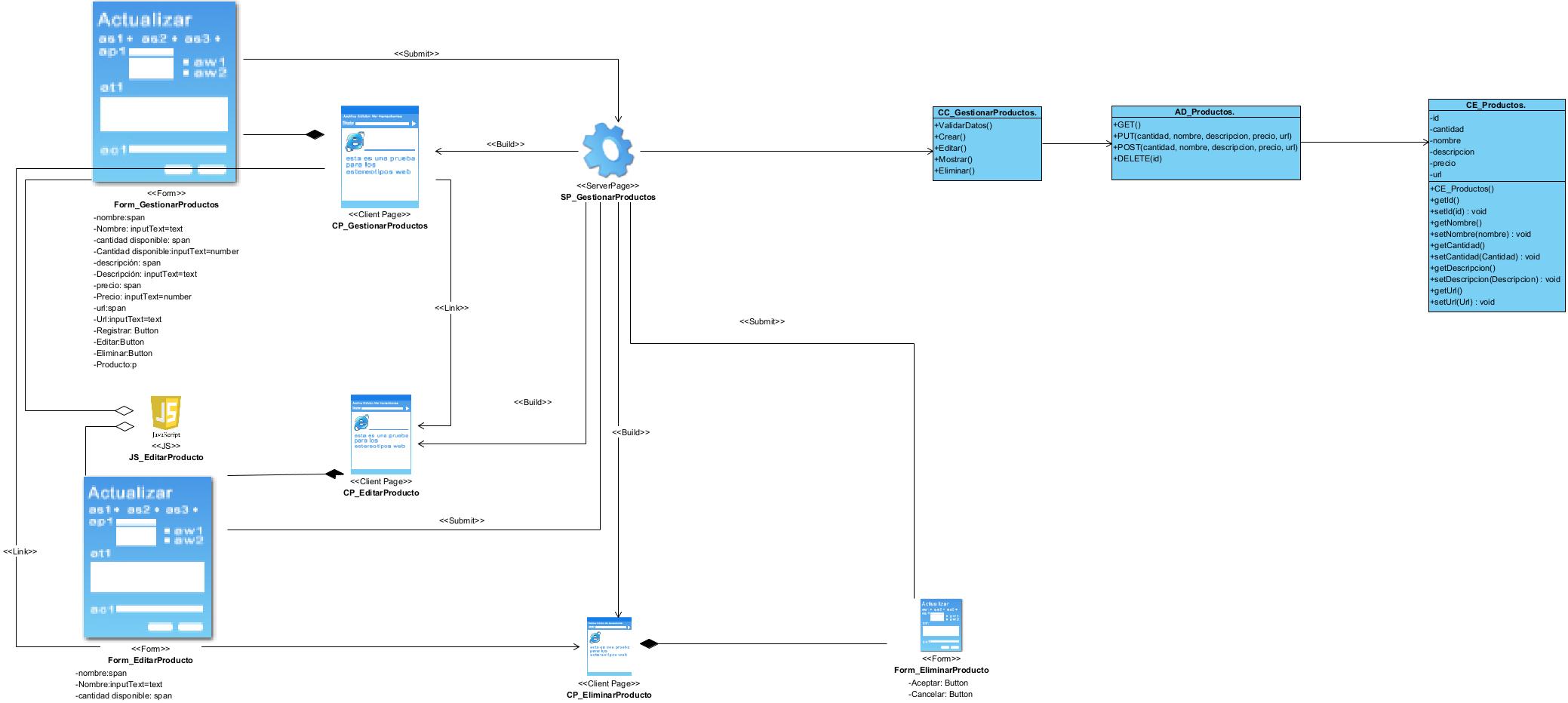
**Ejemplo:** dependiendo del estado de la operación (éxito o error), se llama a Swal.fire() para crear y mostrar la alerta adecuada.

### II.4.2 Diagrama de clase del diseño

Un diagrama de clases de diseño es un tipo de diagrama UML (lenguaje de modelado unificado) que representa las clases, sus atributos, métodos y las relaciones entre ellas dentro de un sistema. Sirve como modelo para la arquitectura del sistema y detalla cómo las diferentes clases interactúan y colaboran para cumplir con los requisitos del sistema. Este diagrama es crucial para los desarrolladores, ya que ayuda a visualizar la estructura del software y facilita la comprensión, la comunicación y la implementación del diseño durante el proceso de desarrollo del software. Al proporcionar una representación clara de los componentes del sistema, los diagramas de clases de diseño ayudan a garantizar que el software se cree de acuerdo con el diseño y los requisitos especificados(NTUU " et al., 2022).



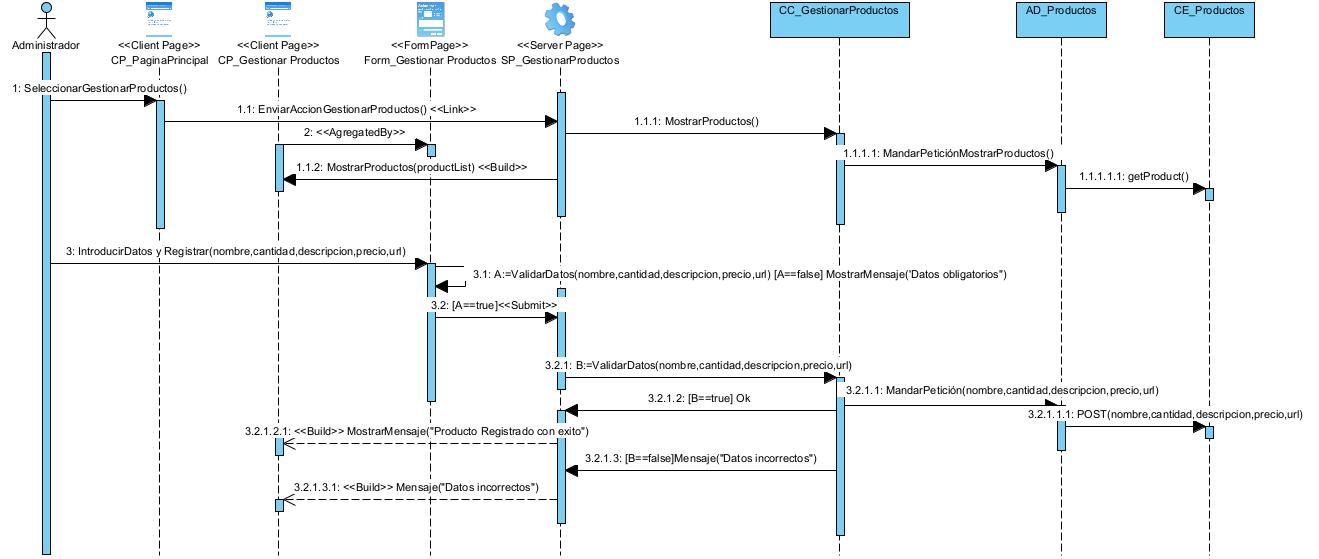
***Figura 11: Diagrama de Clase de Diseño del CU "Comprar Productos".***

**

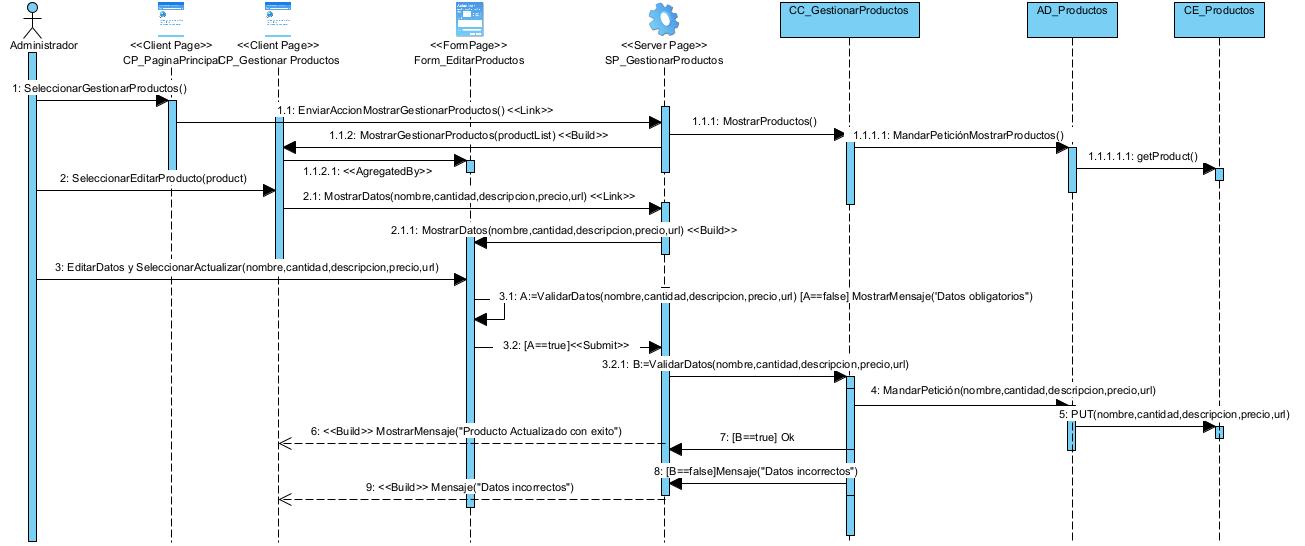
***Figura 12: Diagrama de Clase de Diseño del CU "Gestionar Productos".***

### II.4.3 Diagrama de Secuencia

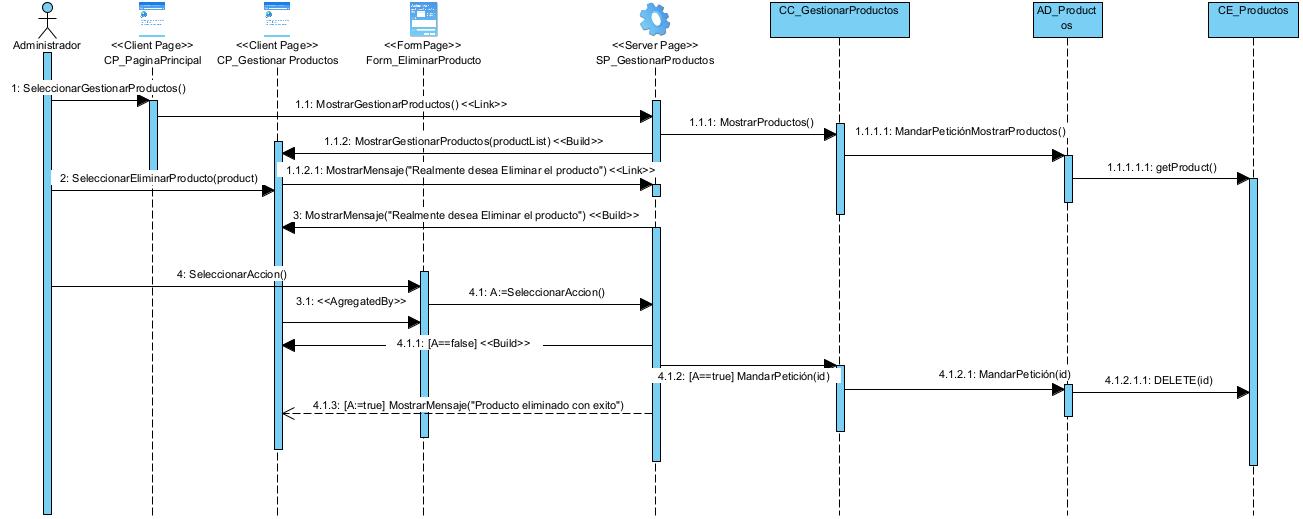
Un diagrama de secuencia es un tipo de diagrama UML que ilustra cómo interactúan los objetos en un escenario particular de un caso de uso, centrándose en el orden de los mensajes que se intercambian entre ellos a lo largo del tiempo. Representa visualmente la secuencia de eventos y muestra las interacciones entre los diferentes componentes u objetos de un sistema, junto con el momento en que se producen estas interacciones. Los diagramas de secuencia son esenciales para comprender el comportamiento dinámico de un sistema, ya que ayudan a los desarrolladores y a las partes interesadas a visualizar el flujo de control y datos, lo que facilita la identificación de posibles problemas y la optimización del diseño. Al proporcionar una descripción clara de las interacciones, los diagramas de secuencia facilitan una mejor comunicación entre los miembros del equipo y sirven como una herramienta valiosa tanto para el diseño como para la documentación(NTUU " et al., 2022).



***Figura 13: Diagrama de Secuencia del caso de uso “Crear Producto”***

**

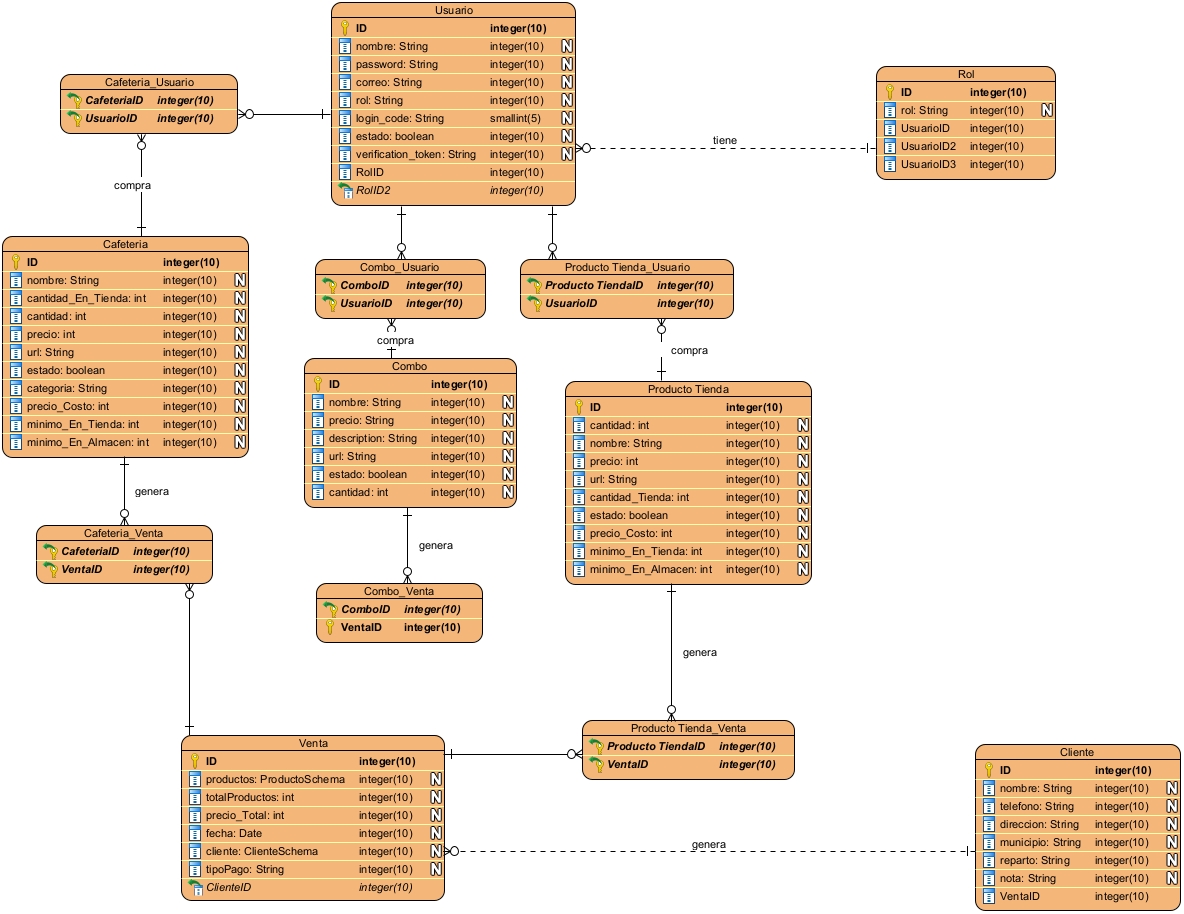
***Figura 14: Diagrama de Secuencia del caso de uso “Editar Producto”.***

**

***Figura 15: Diagrama de Secuencia del caso de uso “Eliminar Producto”.***

### II.4.4 Modelo de Datos

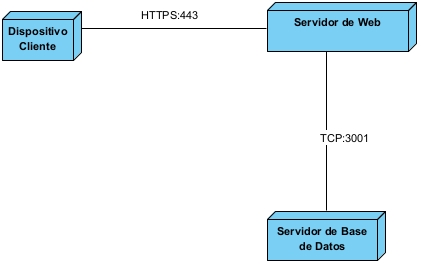
Un modelo de datos es un marco conceptual que define cómo se estructuran, almacenan y manipulan los datos dentro de un sistema. Proporciona una representación visual de los elementos de datos, sus relaciones y las reglas que los rigen, lo que mejora la comunicación entre las partes interesadas y sirve como modelo para el diseño de bases de datos. Los modelos de datos ayudan a identificar posibles problemas en las primeras etapas del proceso de desarrollo, facilitan la resolución eficaz de los problemas y actúan como documentación durante todo el ciclo de vida del software. En general, son esenciales para administrar los datos de manera eficiente y garantizar la integridad y la coherencia de las aplicaciones de software(NTUU " et al., 2022).



***Figura 16: Modelo de Datos del Sistema***

### II.4.5 Diagrama de Despliegue

Un diagrama de implementación es un tipo de diagrama estructural en ingeniería de software que ilustra la disposición física de los artefactos de software en los nodos de hardware. Sirve para visualizar cómo se distribuyen los componentes de software en varios dispositivos, destacando la configuración del sistema y las rutas de comunicación entre los nodos. Este diagrama es esencial para la planificación de la implementación, el análisis del rendimiento y la documentación de la arquitectura del sistema, ya que garantiza la administración y el mantenimiento efectivos de los sistemas de software en un entorno físico(NTUU " et al., 2022).



***Figura 17: Diagrama de Despliegue.***

## Conclusiones del capítulo

En el presente capitulo que finaliza se abordaron los aspectos básicos relacionados con el análisis y diseño. Se definieron 54 requisitos funcionales. Se definió el modelo conceptual, los diagramas de casos de uso del sistema, de clase de diseño, de entidad-relación, de despliegue. Se expusieron los actores del sistema. Se definieron los patrones de caso de uso, de diseño y los arquitectónicos utilizados en el sistema. Lo antes mencionado sentó las bases para la siguiente etapa de construcción del sistema de gestión y comercio electrónico en la Tienda Ricardo & Neyde.

# 

# CAPÍTULO III: Implementación y prueba del sistema

## 

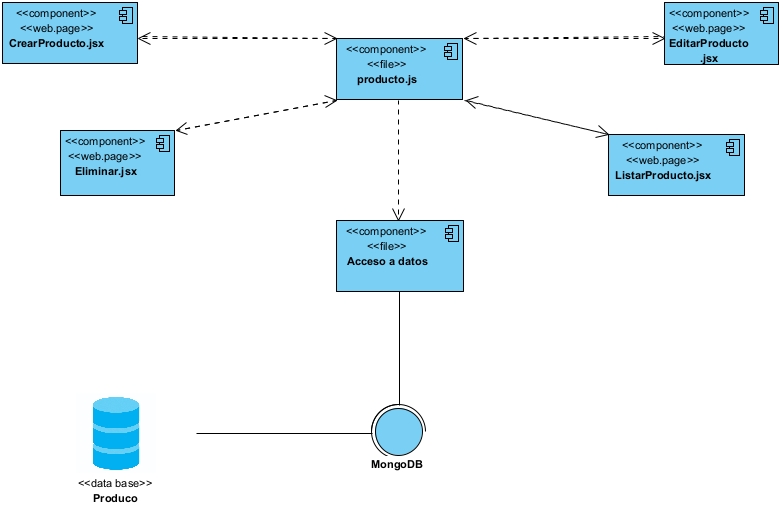
## Como resultado de las fases anteriores se obtendrá un producto funcional para entregar al cliente, el cual debe garantizar los requisitos definidos con la calidad requerida. En el capítulo se aborda sobre las acciones que se llevan a cabo durante la fase de implementación y pruebas. Se analiza el modelo de implementación y se diseña y proponen las pruebas para comprobar el correcto funcionamiento del sistema web.

## III.1 Modelo de Implementación

## El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes podemos encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos(Unknown, 2013).

## Diagrama de Componentes

Un diagrama de componentes es un tipo de diagrama estructural del lenguaje unificado de modelado (UML) que ilustra la organización y las dependencias entre un conjunto de componentes. Proporciona una representación visual de los componentes de un sistema y muestra cómo interactúan y se comunican entre sí. Cada componente representa una parte modular del sistema que encapsula un conjunto de funciones o datos relacionados, y el diagrama normalmente incluye interfaces, puertos y relaciones entre los componentes. Esto ayuda a comprender la arquitectura del sistema y facilita un mejor diseño e implementación de los sistemas de software. Los diagramas de componentes son particularmente útiles en la ingeniería de software para visualizar la estructura de alto nivel de las aplicaciones y garantizar que todas las partes funcionen juntas de manera efectiva(Shan, 2020).



***Figura 18: Diagrama de Componentes del CU “Gestionar Producto”.***

## III.2 Pruebas de Software

Las pruebas de software son evaluaciones sistemáticas que se llevan a cabo para evaluar la funcionalidad, el rendimiento y la confiabilidad de las aplicaciones de software. Se utilizan para identificar defectos o errores en el software antes de su implementación, garantizando que cumpla con los requisitos especificados y funcione según lo esperado. Las pruebas pueden adoptar diversas formas, incluidas las pruebas unitarias, las pruebas de integración, las pruebas del sistema y las pruebas de aceptación, cada una de las cuales se centra en diferentes aspectos del software. El objetivo principal de las pruebas de software es mejorar la calidad del producto de software, reducir el riesgo de fallos en la producción y garantizar una experiencia de usuario positiva(Shan, 2020).

### III.2.1 Pruebas unitarias

Corresponden a la validación de una pieza, componente, módulo o subprograma específico de un sistema para comprobar que cada una de esas partes funciona correctamente por separado.

**Métodos de pruebas**

**Método de Caja Blanca**

Las técnicas de pruebas de *caja blanca* son aquellas que se basan en el conocimiento interno del código de un programa o sistema. Estas técnicas son muy útiles para encontrar errores en el código y garantizar que no exista [código muerto](https://es.wikipedia.org/wiki/C%C3%B3digo_muerto#:~:text=En%20programaci%C3%B3n%2C%20se%20conoce%20como,algo%20que%20jam%C3%A1s%20se%20utiliza.) o algún aspecto del sistema sin probar(Gomez, 2023).

Para la realización de este método se optó por la utilización de la técnica de Camino Básico.

**Método de Caja Negra**

Las pruebas de caja negra, también conocidas como pruebas funcionales, son un tipo de pruebas de software basadas en el grado de conocimiento de los procesos - en este caso nulo - que se dispone al momento de realizar un conjunto de pruebas.

Las operaciones que se realizan en el proceso, por ser de tipo *caja negra* no son visibles por el usuario, ya que desconoce su implementación. Sin embargo, a pesar de no conocer la forma en la que se le da solución al problema, conoce los datos de entrada y la salida que arroja el proceso(Ramiro Maquieira, 2023)**.**

**Técnicas de pruebas**

**Pruebas Unitarias**

Las pruebas unitarias son el proceso en el que se prueba la unidad funcional de código más pequeña. Las pruebas de software ayudan a garantizar la calidad del código y son una parte integral del desarrollo de software. Una práctica recomendada en el desarrollo de software es escribir el software como unidades pequeñas y funcionales, y luego escribir una prueba unitaria para cada unidad de código. Puede escribir primero pruebas unitarias como código. Luego, ejecute ese código de prueba de forma automática cada vez que realice cambios en el código del software. De esta forma, si una prueba falla, puede aislar con rapidez el área del código que tiene el error. Las pruebas unitarias imponen paradigmas de pensamiento modular y mejoran la cobertura y calidad de las pruebas. Las pruebas unitarias automatizadas permiten que usted o sus desarrolladores dispongan de más tiempo para centrarse en la programación(AWS, 2023b).

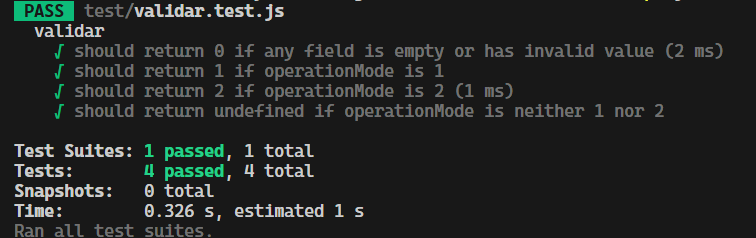


***Figura 19: Función de prueba***

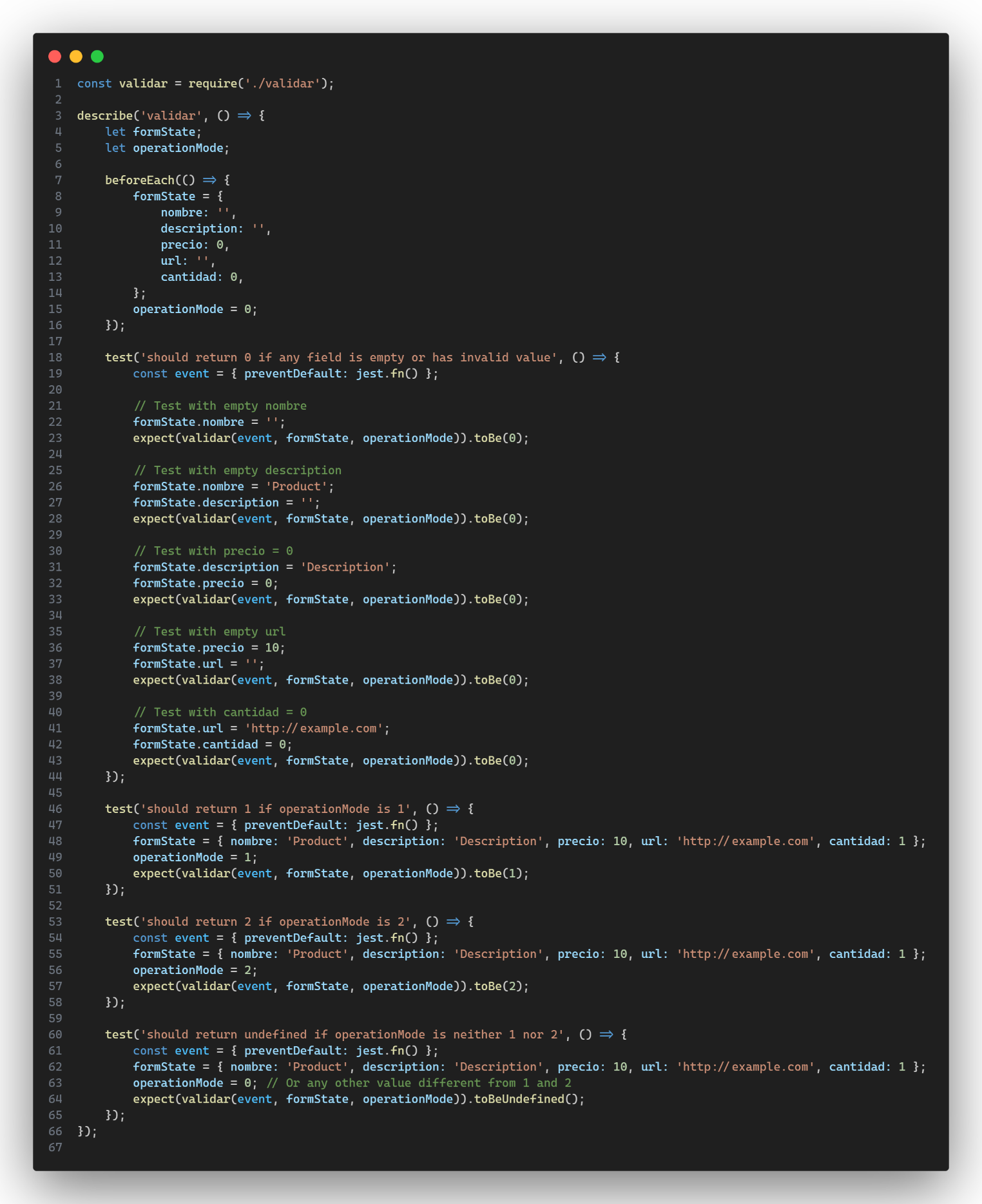


***Figura 20: Prueba unitaria 3 fallidas y 1 pasada***

La función validar estaba intentando acceder a la variable formState que no estaba definida dentro de su ámbito. Esto causó que todas las pruebas fallaran, ya que formState y operationMode no estaban siendo correctamente manejados en el contexto de las pruebas. Para resolver el problema, se modificó la función validar para que aceptara formState y operationMode como parámetros en lugar de depender de variables globales. También se ajustaron las pruebas para pasar estos parámetros a la función validar. Después de estos cambios, las pruebas verificaron adecuadamente los casos en que formState contiene valores válidos y operationMode es 1, 2, o un valor diferente, asegurando que la función devolviera los resultados esperados. Esto permitió que todas las pruebas pasaran correctamente, confirmando que la función validar ahora maneja correctamente todos los escenarios previstos.



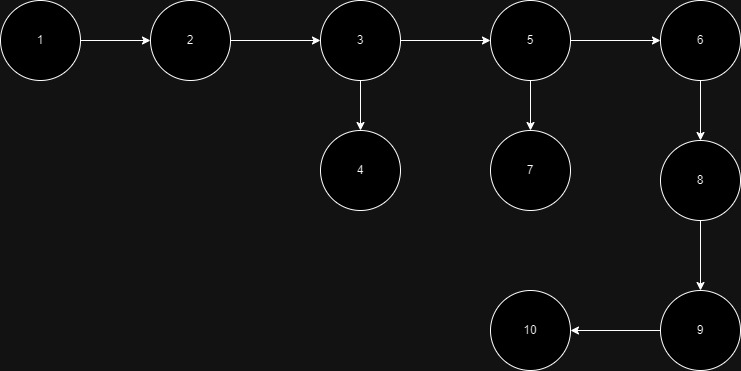
***Figura 21: Prueba Unitaria Satisfactoria***



***Figura 22: Prueba Unitaria***

### Camino Básico

La prueba del camino básico es una técnica de pruebas de Caja Blanca propuesta por Tom MacCabe. Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico. La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control(Martínez, 2011)**.**

****

***Figura 23: Representación del grafo de flujo: Técnica de camino básico.***

**Complejidad Ciclomática**

|  |
| --- |
| **V(G)** |
| A (Aristas) = 9 |
| N (Nodos) = 10 |
| V(G) = A - N + 2= 9-10+2=1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Caso de Prueba** | **Descripción** | **Condición de Ejecución** | **Datos de Entrada** | |  | | --- | |  |  |  | | --- | | **Tipo de Datos Esperado** | | **Evaluación del Caso de Prueba** |
| Caso de Prueba 1 | Validación falla | Algún campo es inválido | formState = { nombre: '', description: '', precio: 0, url: '', cantidad: 0 }; operationMode = 0 | 0 | Satisfactoria |
| Caso de Prueba 2 | operationMode es 1 | Todos los campos válidos, operationMode es 1 | formState = { nombre: 'Product', description: 'Description', precio: 10, url: '<http://example.com>', cantidad: 1 }; operationMode = 1 | 1 | Satisfactoria |
| Caso de Prueba 3 | operationMode es 2 | Todos los campos válidos, operationMode es 2 | formState = { nombre: 'Product', description: 'Description', precio: 10, url: '<http://example.com>', cantidad: 1 }; operationMode = 2 | 2 | Satisfactoria |
| Caso de Prueba 4 | operationMode es distinto a 1 o 2 | Todos los campos válidos, operationMode no es 1 ni 2 | formState = { nombre: 'Product', description: 'Description', precio: 10, url: '<http://example.com>', cantidad: 1 }; operationMode = 0 | undefined | Satisfactoria |

***Tabla 4: Prueba unitaria: Técnica de camino básico.***

### **III.2.2 Pruebas de aceptación**

Según, el International Software Testing Qualification Board (ISTQB, por sus siglas en inglés) define la “Aceptación” como: pruebas formales con respecto a las necesidades del usuario, requerimientos y procesos de negocio, realizadas para determinar si un sistema satisface los criterios de aceptación que permitan que el usuario, cliente u otra entidad autorizada pueda determinar si acepta o no el sistema(ISTQB, 2023)**.**

**Tipos de pruebas**

**Pruebas de Funcionalidad**

Las [pruebas funcionales](https://www.techopedia.com/definition/19509/functional-testing) en las [pruebas de software](https://www.techopedia.com/definition/17681/software-testing) son una forma de determinar si el software o una aplicación funcionan como se espera. Las pruebas funcionales no se ocupan de cómo se produce el procesamiento, sino de si éste ofrece los resultados correctos o tiene algún fallo.

Al realizar una prueba funcional, se busca cualquier laguna, error o cualquier cosa que falte en los requisitos del software o la aplicación.

La diferencia entre las pruebas [del sistema](https://www.techopedia.com/definition/22445/system-testing) y las pruebas funcionales es que las primeras prueban todo el sistema, mientras que las segundas sólo prueban una característica(Zap Chernyak, 2024b).

**Pruebas no Funcionales**

Para la realización de las pruebas no funcionales se optó por aplicar la siguiente prueba que a continuación se describe:

**Pruebas de Usabilidad**

Las pruebas no funcionales son la mejor manera de hacer que el software sea más fácil de usar, sobre todo realizando pruebas de usabilidad que evalúen lo fácil que es para los usuarios aprender a utilizar y manejar el software(Zap Chernyak, 2024a).

## III.3 Casos de pruebas

En ingeniería del software, un caso de prueba (en inglés, test case) es un conjunto de condiciones o variables bajo las cuales se determinará si una aplicación, un sistema de software o una característica o comportamiento de estos resulta o no aceptable. Se pueden realizar muchos casos de prueba para determinar que un requisito es completamente satisfactorio. Lo que caracteriza un escrito formal de caso de prueba es que hay una entrada conocida y una salida esperada, los cuales son formulados antes de que se ejecute la prueba. La entrada conocida debe probar una precondición y la salida esperada debe probar una postcondición. Los casos de prueba escritos incluyen una descripción de la funcionalidad que se probará, la cual es tomada ya sea de los requisitos o de los casos de uso, y la preparación requerida para asegurarse de que la prueba pueda ser dirigida.

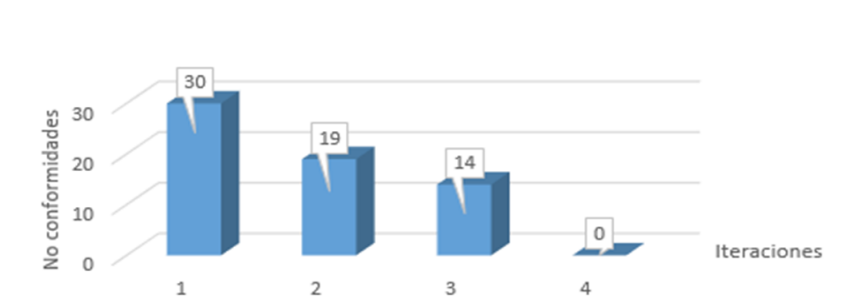
Se presenta el caso de prueba del requisito funcional Crear Producto utilizando del método de Caja Negra la técnica Partición de Equivalencia, donde se elige uno de los posibles escenarios que se pueden presentar durante la ejecución del requisito, detectando un error en el escenario EP2.2 del campo No. Inventario mostrado en la Tabla 5:

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre del requisito | Descripción general | Escenario de pruebas | Flujo del escenario |
| RF25 Crear Producto | Permite crear un nuevo Producto | EP1: Adicionar Producto de forma correcta. | 1- Se muestra una interfaz Lista de Productos con el listado de los Productos existentes.  2- Se selecciona la opción Añadir Nuevo Producto.  3- Se introducen los siguientes datos: Nombre, Cantidad en Tienda, Cantidad en Almacén, Precio de Costo, Precio, Mínimo en la Tienda, Mínimo en el Almacén, Imagen.  4- Se selecciona la opción Guardar.  5- El sistema muestra un mensaje de notificación: "Producto creado co” |
| EP1.2: Adicionar Producto de forma incorrecta. | 1- Se muestra una interfaz Lista de Productos con el listado de los Productos existentes.  2- Se selecciona la opción Añadir Nuevo Producto.  3- Se introducen datos, dejando campos obligatorios vacíos.  4- Se señalan los campos y se muestra el mensaje: "Este campo es requerido". |
| EP1.3: Cancelar Adicionar Producto | 1- Se muestra una interfaz Lista de Productos con el listado de los Productos existentes.  2- Se selecciona la opción Añadir Nuevo Producto.  3-Se dejan en blanco los campos a introducir o se llenan los siguientes datos: Nombre, Cantidad en Tienda, Cantidad en Almacén, Precio de Costo, Precio, Mínimo en la Tienda, Mínimo en el Almacén, Imagen.  4- Se selecciona la opción Cancelar.  5- Se anulan las acciones y se cierra la interfaz, regresando a la interfaz Lista de Productos. |

***Tabla 5: Diseño de caso de prueba: RF25 Crear Producto.***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nombre del requisito | Escenario | Descripción | Juego de datos | Resultados obtenidos | Observaciones |
| Crear Producto | EP2.1 | Se adiciona un nuevo Producto para mostrar si los campos Nombre, Cantidad en Tienda, Cantidad en Almacén, Precio de Costo, Precio, Mínimo en la Tienda, Mínimo en el Almacén, Imagen, aceptan campos vacíos. | Nombre: vacío  Cantidad en Tienda: vacío  Cantidad en Almacén: vacío  Precio de Costo: vacío  Precio: vacío  Mínimo en la Tienda: vacío  Mínimo en el Almacén: vacío  Imagen: vacío | El sistema muestra el mensaje: “Campos vaciós, todos los campos son obligatorios.” |  |
| EP2.2 | Se adiciona un Producto para mostrar si los campos aceptan valores existentes. | Nombre: vacio  Precio: vacio  Imagen: vacio | El sistema muestra el mensaje: “Campos vaciós, todos los campos son obligatorios.” | El error fue corregido satisfactoriamente. Para adicionar un Producto el campo todos los campos tienen que estar llenos. |

***Tabla 6: Diseño de prueba de validación RF29 Crear Producto***

******

***Figura 24: Resultado de las pruebas de aceptación.***

Se realizaron cuatro iteraciones de pruebas al módulo de gestión de los productos de la tienda, haciendo uso de los Diseños de casos de prueba elaborados, con lo que se obtuvieron los resultados siguientes:

• en la primera iteración se detectaron 30 no conformidades, principalmente en campos de los formularios que aceptan entradas incorrectas o valores vacíos y algunos funcionales, por ejemplo, el botón Editar Producto no funciona correctamente, la funcionalidad Nuevo Producto no listaba el campo de buscar imagen.

• Para una segunda iteración se resolvieron 17 no conformidades y se detectaron 6, 2 de aplicación, 1 de diseño y 3 de validación.

• Para una tercera iteración se resolvieron 5 no conformidades de validación.

• La cuarta iteración arrojó resultados satisfactorios, ya que se resolvieron las no conformidades restantes y no se encontraron nuevas.

### Pruebas de Rendimiento

Las pruebas de rendimiento de *software* se definen como un **análisis de comprobación del funcionamiento del sistema frente a múltiples escenarios** de ensayo, con el fin de examinar los componentes de la aplicación. Esto quiere decir que este tipo de pruebas se basan en garantizar la calidad y operatividad de un determinado sistema.

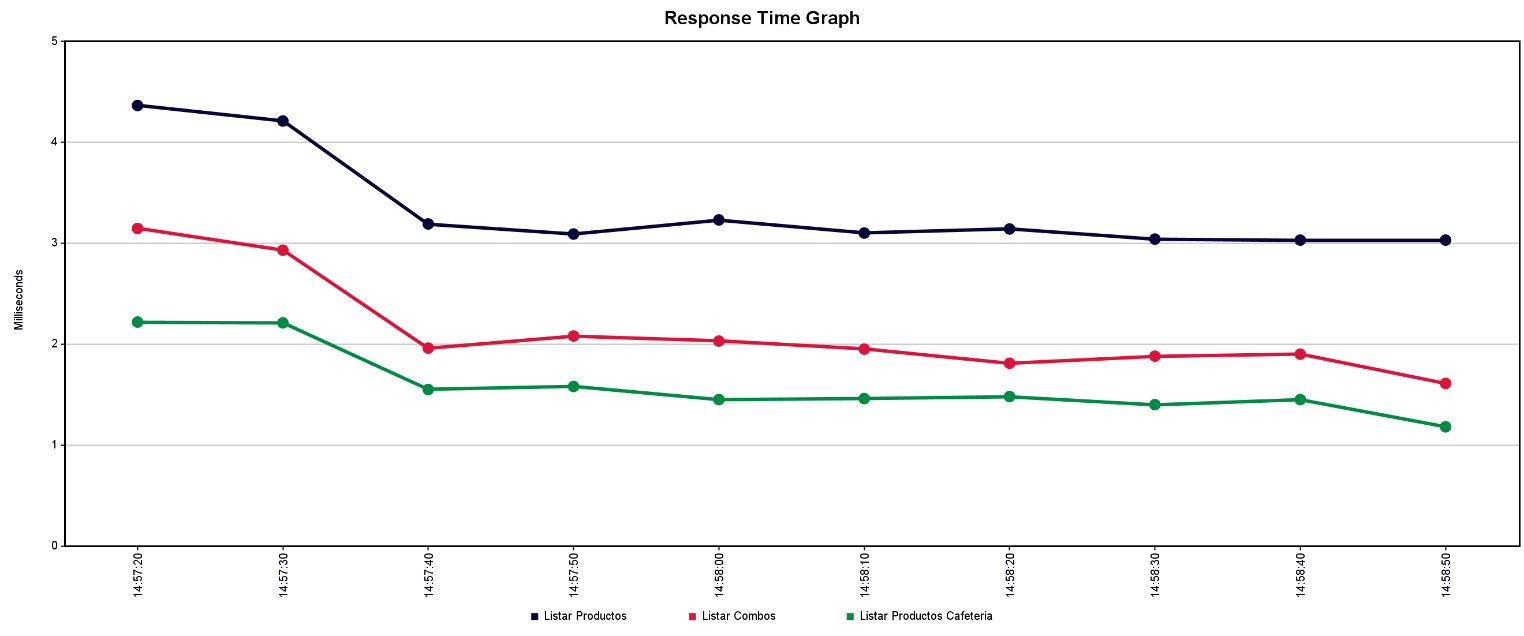
Estos test destacan también como un tipo de prueba no funcional, que se **encarga de determinar propiedades como la capacidad de respuesta**, la estabilidad, escalabilidad, etc., de un programa o aplicación bajo una carga de trabajo específica(Mallón, 2022).

**Apache JMeter** es un software de código abierto basado en Java puro, desarrollado por primera vez por Stefano Mazzocchi de la Apache Software Foundation, diseñado para realizar pruebas de carga del comportamiento funcional y medir el rendimiento. Apache JMeter es una herramienta de prueba que se utiliza para analizar y medir el rendimiento de diferentes servicios y productos de software. Es un software de código abierto basado en Java puro que se utiliza para probar aplicaciones web o aplicaciones FTP.

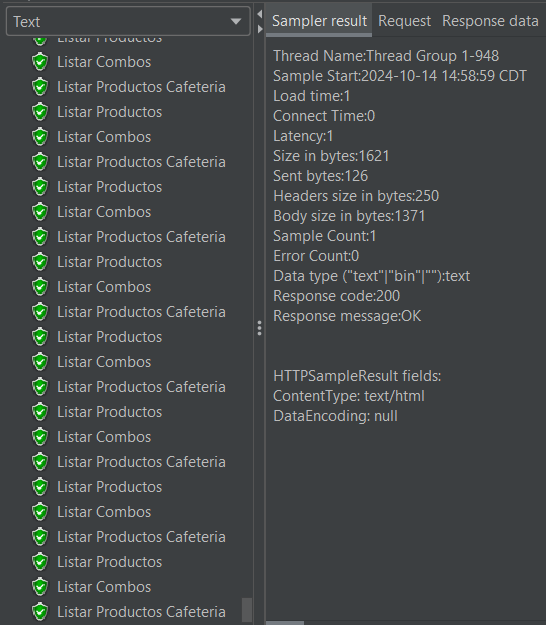
Se utiliza para ejecutar pruebas de rendimiento, pruebas de carga y [pruebas funcionales](https://successive.tech/manual-vs-automation-testing) de aplicaciones web. JMeter también puede simular una carga pesada en un servidor mediante la creación de toneladas de usuarios virtuales simultáneos en el servidor web(Successive, 2020).

Para las pruebas de Rendimiento se usó el software Apache Jmeter, utilizando las pruebas:

* **View Results Tree**: muestra los detalles de cada solicitud enviada durante la prueba. Permite verificar y analizar las respuestas individuales, incluyendo el contenido de la respuesta y los encabezados.
* **Summary Report**: proporciona un resumen estadístico de la prueba, mostrando métricas clave como el número de solicitudes, el tiempo medio de respuesta y la tasa de errores. Es útil para obtener una visión general del rendimiento de la aplicación.
* **Response Time Graph**: grafica los tiempos de respuesta de las solicitudes a lo largo del tiempo. Ayuda a visualizar el rendimiento de la aplicación y a identificar patrones o anomalías en los tiempos de respuesta.



***Figura 25: Resultado de las pruebas de tiempo de respuesta Apache Jmeter.***



***Figura 26: Resultado de las pruebas de tiempo de respuesta Apache Jmeter.***

## Resumen de los Resultados

1. **Cantidad de Solicitudes**: se simularon un total de **3000 solicitudes**, distribuidas entre tres grupos principales: listar Combos, Listar Productos, y Listar Productos Cafetería. Cada grupo recibió **1000 solicitudes** a lo largo del tiempo.
2. **Resultados Exitosos**: todas las solicitudes respondieron con un **estatus 200 (OK)**, lo que significa que no hubo errores en el procesamiento de los datos ni fallos en el servidor. Esto refleja un manejo robusto de las peticiones en el backend y un flujo de datos estable.
3. **Tiempos de Respuesta**: el **tiempo de carga promedio** (Load time) fue extremadamente bajo, en torno a **1 ms**, con una **latencia igualmente baja** de **1 ms**. Este rendimiento rápido indica que el sistema está respondiendo de manera muy eficiente, incluso bajo carga pesada.
4. **Líneas de Tiempo y Rendimiento**:
   * el gráfico generado fue **casi una línea horizontal**, lo que indica que los tiempos de respuesta fueron consistentes y no se observaron picos significativos. Este comportamiento es ideal para una aplicación que necesita manejar solicitudes de manera constante y rápida.
   * la curva estable sugiere que **React Query** gestionó las peticiones eficientemente, proporcionando tiempos de respuesta uniformes sin importar la cantidad de solicitudes simultáneas.
5. **Consumo de Recursos**:
   * el tamaño en bytes de cada respuesta fue constante, con un promedio de **1621 bytes** por respuesta.
   * las solicitudes enviadas tenían un tamaño mínimo, lo que es beneficioso para la reducción del tráfico de red y mejora el rendimiento general.

**Análisis Avanzado de Rendimiento con React Query**

**React Query** es una biblioteca que simplifica la gestión de datos en aplicaciones React. Proporciona una forma sencilla de realizar consultas a APIs y gestionar la caché de los datos recuperados. Además, React Query ofrece varias características poderosas que facilitan el manejo de estados, la paginación, la revalidación inteligente y las mutaciones. A continuación, veremos algunas de sus ventajas(OscarBravelop, 2023).

Se hizo uso de React Query para gestionar las peticiones HTTP, demostrando ser una solución eficaz. Este hook avanzado permite:

* **caching**: React Query almacena en caché los resultados de las solicitudes, lo que minimiza la necesidad de realizar peticiones repetitivas, mejorando el rendimiento.
* **Sincronización automática**: las peticiones se mantienen actualizadas en tiempo real sin recargar la página, lo que proporciona una experiencia de usuario fluida.
* **Reintentos automáticos**: en caso de fallos temporales en las solicitudes, React Query puede reintentarlas automáticamente, garantizando una mayor fiabilidad.

### Prueba de Usabilidad

**Listas de Chequeo**

Una lista de chequeo o verificación es una herramienta impresa a modo de formato, utilizada para recoger y compilar de forma estructurada datos asociados a un proceso o situación particular definida. Los datos reunidos representan una entrada para el uso de otras herramientas de control de calidad como el [diagrama de Pareto](https://www.ingenioempresa.com/diagrama-de-pareto/) o [dispersión](https://www.ingenioempresa.com/diagrama-de-dispersion/). En este sentido, la hoja de verificación es una herramienta genérica utilizada para multitud de propósitos que van más allá de la calidad(Betancourt, 2016).

Se ha realizado una evaluación detallada del sitio web, analizando diversos aspectos de usabilidad, visibilidad, consistencia, control del usuario, entre otros. El objetivo era asegurarnos de que la experiencia del usuario sea fluida, intuitiva y eficiente.

**(Ver Anexo) 2. Lista de Chequeo**

**Estadísticas del sitio:**

* De un total de **115 indicadores**, el sitio ha cumplido con **95 indicadores** (82.6%), lo que refleja un nivel de cumplimiento bastante sólido.
* **9 indicadores** no proceden (NP), lo que significa que no aplican a este caso.
* Se ha identificado **11 áreas de mejora** (9.6%), donde se puede optimizar aún más la experiencia del usuario.

### Prueba de Seguridad

**Lista de Chequeo**

Se ha realizado una evaluación detallada del sitio web, analizando diversos aspectos de seguridad, usuarios, consistencia, control del usuario, entre otros. El objetivo era asegurarnos de que la experiencia del usuario sea fluida, intuitiva y eficiente.

**(Ver Anexo) 3. Lista de Chequeo de Seguridad**

**Análisis Estadístico**

**Total de Indicadores Evaluados:** 48

**Indicadores Cumplidos (1):** 40

**Indicadores No Cumplidos (0):** 8

El sistema cumple con la mayoría de los controles de seguridad (83.33%), lo cual indica un nivel alto de seguridad en las áreas evaluadas.

## Conclusiones del capítulo

En este capítulo se desarrolló el modelo de implementación del sistema para mostrar sus componentes y relaciones mediante un diagrama de componentes. Se realizaron pruebas al sistema para evaluar su calidad, comenzando con pruebas funcionales mediante el método de Caja Negra, que detectaron un error que fue corregido, y pruebas de aceptación que en su primera iteración presentaron 30 no conformidades, reducidas a 6 en la tercera iteración, logrando finalmente una cuarta iteración satisfactoria. En la prueba de Caja Blanca, se encontró un error en el primer caso de prueba, mientras que el segundo fue exitoso, obteniendo así un producto libre de errores y listo para su ejecución. Además, se comprobó que el sistema web para la gestión de la Tienda Ricardo & Neyde satisface las necesidades del cliente, demostrando ser altamente eficiente en la gestión de grandes volúmenes de solicitudes simultáneas sin afectar los tiempos de respuesta. El uso de React Query optimizó el manejo de las peticiones, garantizando un rendimiento constante, mientras que los resultados de las pruebas de carga, con gráficos estables y sin errores, sugieren que la aplicación puede escalar fácilmente y ofrecer una experiencia de usuario excelente incluso bajo alta demanda.

# 

# CONCLUSIONES

* El sistema se diseñó según los principios de usabilidad y diseño centrado en el usuario, basados en teorías de sistemas de información. Esto permitió desarrollar una interfaz accesible y una estructura que optimiza la eficiencia en la gestión de datos para satisfacer las necesidades de la tienda.
* Se utilizó la metodología AUP UCI, lo que facilitó un desarrollo iterativo, ágil y controlado del sistema. Las herramientas incluyeron React para el frontend, Node.js y Express para el backend, y MongoDB como base de datos, permitiendo una gestión modular y adaptable a cambios.
* Cada función se implementó como un módulo autónomo, permitiendo escalabilidad y mantenimiento sencillo. Este enfoque modular optimizó el uso de código reutilizable y facilitó la integración de futuras mejoras sin comprometer la funcionalidad del sistema.
* La validación incluyó pruebas unitarias, de integración y de aceptación, confirmando que el sistema cumple con los requisitos funcionales y de calidad. Las pruebas de aceptación permitieron asegurar que el sistema gestionara los procesos de la tienda de manera eficiente y sin errores.

# 

# RECOMENDACIONES

* Implementar un módulo de recomendación de productos.
* Implementar un módulo de reconocimiento de imágenes para la búsqueda de productos.
* Implementar un módulo para la venta de productos de ropa de la Boutique.

# 

# REFERENCIAS BIBLIOGRÁFICAS

A, D. (2020, febrero 4). Qué es React: Definición, características y funcionamiento. *Tutoriales Hostinger*. https://www.hostinger.es/tutoriales/que-es-react

Abigail Orús, E. texto proporciona información general S. no se hace responsable de la veracidad o exactitud del contenido N. ciclos de actualización, & Texto, D. M. Q. L. E. P. C. I. M. A. Q. L. R. E. E. (2024, abril 23). *Tema: Comercio electrónico en el mundo*. Statista. https://es.statista.com/temas/9072/comercio-electronico-en-el-mundo/

AWS. (2023a). *¿Qué es JavaScript? - Explicación de JavaScript (JS)—AWS*. https://aws.amazon.com/es/what-is/javascript/

AWS. (2023b). *¿Qué son las pruebas unitarias?: Explicación de las pruebas unitarias en AWS*. Amazon Web Services, Inc. https://aws.amazon.com/es/what-is/unit-testing/

Betancourt, D. F. (2016, agosto 2). *Lista de chequeo o verificación como herramienta de calidad*. https://www.ingenioempresa.com/lista-de-chequeo/

Brañas, A. (2023, junio 12). *Desbloquea los Verdaderos Beneficios de un Sistema de Gestión de Compras*. Qflow. https://qflowbpm.com/es/compras/beneficios-sistema-compras/

Casero, A. (2023, julio 21). *Ventajas de usar Node.js | KeepCoding Bootcamps*. https://keepcoding.io/blog/ventajas-de-usar-node-js/

Castro Ramirez, E. R. (2023). *Patrones en la Arquitectura de Software*. https://es.linkedin.com/pulse/patrones-en-la-arquitectura-de-software-elmo-renato-castro-ramirez

DQS. (2024). *¿Qué es un sistema de gestión?* DQS. https://www.dqsglobal.com/es-sv/aprenda/centro-de-conocimiento-dqs/que-es-un-sistema-de-gestion

Ekon. (2001). *¿Qué es un sistema de gestión y para qué sirve?* https://www.ekon.es/blog/sistemas-de-gestion-integral-para-el-funcionamiento-optimo-de-la-empresa/

Fowler, M., & Scott, K. (1997). *UML distilled: Applying the standard object modeling language*. Addison Wesley Longman.

García de Zúñiga, F. (2024, julio 1). *¿Qué es Visual Studio Code y cuáles son sus ventajas? | Blog de Arsys*. Arsys. https://www.arsys.es/blog/que-es-visual-studio-code-y-cuales-son-sus-ventajas

García, L. Y. S. (2017). *Componente para la comparación de plantillas de minucias de huellas dactilares.*

Gomez, C. (2023, febrero 28). *Técnicas de prueba de caja blanca*. Diario de QA. https://www.diariodeqa.com/post/técnicas-de-prueba-de-caja-blanca

IBM. (2023, noviembre 7). *¿Qué es MongoDB? | IBM*. https://www.ibm.com/mx-es/topics/mongodb

ISTQB. (2023). *Buscar—ISTQB Glossary*. https://glossary.istqb.org/es\_ES/search

Karanikas, L., & Öhman, T. (2007). *The Store—The Physical Place of Branding*.

Kim, B. (2019). ICT-Based Business Communication with Customers in the 4th Industrial Revolution Era. *Business Communication Research and Practice*, *2*(2), 55-61. https://doi.org/10.22682/bcrp.2019.2.2.55

Luca, D. D. (2010, noviembre 23). *¿Qué es CSS3?* Damián De Luca. https://damiandeluca.com.ar/que-es-css3

Mallón, X. (2022, agosto 8). *¿Qué son las pruebas de rendimiento? [2024]*. https://keepcoding.io/blog/que-son-las-pruebas-de-rendimiento/

Martínez, E. S. (2011). *Propuesta de Procedimiento para realizar pruebas de Caja Blanca a las aplicaciones que se desarrollan en lenguaje Python*.

MongoDB Documentation. (2024). *MongoDB Documentation*. https://www.mongodb.com/docs/

NTUU ", KPI", Sydorov, N. O., Sydorova, N. M., & NTUU «KPI». (2022). Software engineering and big data software. *PROBLEMS IN PROGRAMMING*, *3-4*, 69-72. https://doi.org/10.15407/pp2022.03-04.069

Nuria. (2024). *Guía completa sobre el diagrama de casos de uso | Boardmix*. https://boardmix.com/es/articles/case-use-diagram/

OscarBravelop. (2023, julio 26). React Query: Gestión de datos eficiente para aplicaciones React. *Medium*. https://medium.com/@OscarBravelop/react-query-gesti%C3%B3n-de-datos-eficiente-para-aplicaciones-react-d5922728a788

Pallavi Gladwin, & Sebastian T. Joseph. (2023). An Analysis on Current Fashions, Complexities & Prospects for India in E-Commerce. *International Journal For Multidisciplinary Research*, *5*(5), 7148. https://doi.org/10.36948/ijfmr.2023.v05i05.7148

Pressman, R. S. (2002). *Visual Paradigm—EcuRed*. https://www.ecured.cu/Visual\_Paradigm

Ramiro Maquieira, G. (2023). *Qué es black box testing o pruebas de caja negra | OpenWebinars*. OpenWebinars.net. https://openwebinars.net/blog/que-es-black-box-testing-o-pruebas-de-caja-negra/

Saavedra, J. A. (2023, junio 1). *Qué es HTML5: Para qué sirve, qué novedades ofrece, ventajas del lenguaje*. https://ebac.mx/blog/que-es-html5

Sagrero, A. (2024). *INVENTARIOS Concepto | PDF | Inventario | Pagos*. Scribd. https://es.scribd.com/document/378014248/INVENTARIOS-Concepto

Santander. (2020, diciembre 21). *Metodologías de desarrollo software | Blog Santander Open Academy*. https://www.santanderopenacademy.com/es/blog/metodologias-desarrollo-software.html

Santos, M. E. S. D., Rocha, T. S., & Perkusich, M. B. (2021). O ensino de engenharia de software no nível superior: Um mapeamento sistemático. *Revista Principia - Divulgação Científica e Tecnológica do IFPB*, *1*(56), 116. https://doi.org/10.18265/1517-0306a2021id3874

Shah, V., & Salles, M. A. V. (2018). *Actor-Relational Database Systems: A Manifesto* (arXiv:1707.06507). arXiv. http://arxiv.org/abs/1707.06507

Shan, D. (2020). Innovation on “Software Engineering” Under Engineering Education Professional Certification. *Teacher Education and Curriculum Studies*, *5*(3), 114. https://doi.org/10.11648/j.tecs.20200503.20

Simões, C. (2021, julio 27). *¿Qué es Node.js, y para qué sirve?* Blog ITDO - Agencia de desarrollo Web, APPs y Marketing en Barcelona. https://www.itdo.com/blog/que-es-node-js-y-para-que-sirve/

Successive. (2020, junio 19). An Overview of Apache JMeter. *Successive Digital*. https://successive.tech/blog/what-is-jmeter/

Sunil, M. (2024). *Beneficios de ReactJS para soluciones empresariales*. WeblineIndia. https://www.weblineindia.com/es/blog/benefits-of-reactjs-for-business.html

Unknown, P. por. (2013). *MODELO DE IMPLEMENTACIÓN*. http://ithleovi.blogspot.com/2013/06/unidad-5-modelo-deimplementacion-el.html

Zap Chernyak, A. (2024a). *Pruebas no funcionales: Proceso, herramientas, tipos y mucho más.* https://www.zaptest.com/es/pruebas-no-funcionales-que-es-tipos-enfoques-herramientas-y-mas

Zap Chernyak, A. (2024b). ¿Qué son las pruebas funcionales? Tipos, ejemplos, lista de comprobación y aplicación. *https://www.zaptest.com/es*. https://www.zaptest.com/es/que-son-las-pruebas-funcionales-tipos-ejemplos-lista-de-comprobacion-y-aplicacion

# 

# ANEXOS

**Anexo.1 Entrevista**

**Nombre del entrevistador:** Ricardo Alejandro D´Escoubet Montes de Oca.

**Centro de estudio:** Universidad de las Ciencias Informáticas (UCI).

**Nombres de los entrevistados:** Ricardo Luis D´Escoubet Bravo (Administrador), Neyde Yeste Rosales (Administradora).

**Centro de trabajo:** Tienda Ricardo & Neyde

**Objetivo:** Conocer la necesidad de la propuesta de solución y definir los requisitos funcionales y no funcionales.

**Preguntas:**

**Estimado/a,**

Le agradecemos de antemano por su valiosa colaboración en el levantamiento de requisitos para el desarrollo de la página web de la tienda. Nos sería de gran utilidad si pudiera brindarnos sus respuestas a las siguientes preguntas:

1. ¿Podría detallar los pasos clave que se siguen en los procesos de venta, abastecimiento y control de inventario de los productos en la tienda?
2. ¿Cuáles son las normas o políticas principales que se deben cumplir en la tienda Denis & Pérez?
3. En su opinión, ¿cuáles son las principales áreas de mejora o deficiencias que observa en el funcionamiento actual de la tienda?
4. ¿Cómo se organiza actualmente la distribución de los productos tanto en la tienda como en el almacén?
5. ¿Qué objetivos principales espera alcanzar con la implementación de la página web para la tienda?
6. ¿Considera que la página web debería incluir alguna característica específica en términos de usabilidad, seguridad, diseño de interfaz u otros aspectos que aseguren su calidad y buen funcionamiento?

**Anexo.2 Lista de Chequeo de Usabilidad**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Indicador a evaluar** | **Evaluación** | | **NP** |
| 1 | ¿La página refleja la identidad de la empresa logos, compañía...)? | **1** | |  |
| 2 | ¿Cada pantalla empieza con un título que describe su contenido? | **1** | |  |
| 3 | ¿Cuándo se selecciona un icono se diferencia de los no seleccionados? | **1** | |  |
| 4 | ¿Los enlaces del menú se resaltan cuando se seleccionan? | **1** | |  |
| 5 | ¿Los iconos que aparecen se identifican claramente con lo que representan? | **1** | |  |
| 6 | ¿El menú de navegación aparece en un lugar destacado? | **1** | |  |
| 7 | ¿No utiliza más de siete opciones principales en el menú de navegación? | **1** | |  |
| 8 | ¿Si la respuesta a una acción se retrasa, aparece un mensaje o indicio como que el sistema está procesando la acción? | **1** | |  |
| 9 | ¿El sitio le indica al usuario en que parte de la estructura del sitio web se encuentra, es decir si muestra 'migas de pan'? | **1** | |  |
| 10 | ¿El nombre de los enlaces es el mismo que el título de la página a la que dirige? | **1** | |  |
| 11 | ¿El logo de la organización está ubicado en el mismo lugar en todas las páginas, y hacer click en el logo retorna al usuario a la página más lógica (Ejemplo: la página de inicio)? | **1** | |  |
| 12 | ¿Los títulos de las páginas, tablas e imágenes son descriptivos y distintivos? | **1** | |  |
| 13 | ¿Las etiquetas de las categorías describen con precisión la información de las mismas? | **1** | |  |
| 14 | ¿Cuándo una tarea involucra documentos fuente, la interfaz es compatible con las características del documento fuente? |  | | **x** |
| 15 | ¿Las imágenes se muestren con buena resolución? | **1** | |  |
| 16 | ¿No se muestran errores ortográficos? | **1** | |  |
| 17 | ¿No hay ninguna imagen con información relevante? | **0** | |  |
| 18 | ¿El lenguaje es simple, con un tono adecuado? | **1** | |  |
| 19 | ¿La información que se presenta en la aplicación es fácil de entender y memorizar? | **1** | |  |
| 20 | ¿Utiliza los conceptos establecidos para las funciones estándar? (“buscar" para las búsquedas, etc.) | **1** | |  |
| 21 | ¿Evita el lenguaje técnico: términos informáticos o propios de Internet? | **1** | |  |
| 22 | ¿Se utiliza siempre la misma nomenclatura para las mismas funciones? | **1** | |  |
| 23 | ¿Los acrónimos y abreviaturas son definidos al ser usados por primera vez? |  | | **x** |
| 24 | ¿No hace uso de términos extranjeros? | **1** | |  |
| 25 | ¿Utiliza un texto específico y descriptivo en los vínculos? | **0** | |  |
| 26 | ¿La información es de rápida lectura, y con una disposición asequible? | **1** | |  |
| 27 | ¿Los vínculos basados en nombres de la gente, conducen a las biografías cortas o a sus propios blogs, no a un correo electrónico? |  | | **x** |
| 28 | ¿Si se desea incluir un enlace de correo electrónico, se muestra el correo y no el nombre de la persona? |  | | **x** |
| 29 | ¿Existe una manera lógica de acceder a páginas relacionadas o a otras secciones? |  | | **X** |
| 30 | ¿Tras una acción relevante hay una opción de vuelta atrás? | **1** | |  |
| 31 | ¿Si una acción tiene consecuencias, el sistema proporciona información y pide confirmación antes de continuar? | **1** | |  |
| 32 | ¿En las páginas internas hay un acceso a la página de inicio en una zona visible y reconocible? | **1** | |  |
| 33 | ¿El Sitio cuenta con un mapa o buscador que facilite el acceso directo a los contenidos? | **1** | |  |
| 34 | ¿Al dar click en el botón “Atrás” siempre lleva al usuario de vuelta a la página de dónde vino? | **1** | |  |
| 35 | ¿El sitio no deshabilita el botón “Atrás” y dicho botón aparece activo en la barra de herramientas del navegador en todas las páginas? | **1** | |  |
| 36 | ¿El sitio evita que los usuarios se registren de manera innecesaria? | **1** | |  |
| 37 | ¿La página se ve con cualquier resolución de pantalla? | **1** | |  |
| 38 | ¿Es posible aumentar y disminuir el tamaño de letra? | **0** | |  |
| 39 | ¿Las imágenes de los productos se pueden ampliar? |  | | **x** |
| 40 | ¿Ofrece el contenido en otros formatos, como dispositivos móviles? | **1** | |  |
| 41 | ¿Es posible imprimir la web sin perder información? |  | | **x** |
| 42 | ¿Se pueden guardar las páginas web? | **0** | |  |
| 43 | ¿El sitio provee una clara retroalimentación cuando una tarea ha sido completada exitosamente? | **1** | |  |
| 44 | ¿Una imagen que sirve como enlace es fácilmente distinguible? |  | | **X** |
| 45 | ¿En caso que se muestre información relacionada con registros obtenidos de la base de datos existe un sistema de navegación donde el usuario pueda especificar cuantos elementos desea ver en la página? | **0** | |  |
| 46 | ¿Los usuarios son informados si es necesario un plug-in del navegador o resolución específico? |  | | **x** |
| 47 | ¿Las páginas que utilizan nuevas tecnologías siguen funcionando cuando dicha tecnología no está presente (por ejemplo, los plug-ins de Flash)? |  | | **x** |
| 48 | ¿Cuándo es necesaria la descarga de un plug-in, hay un enlace a la página donde obtenerlo? |  | | **x** |
| 49 | ¿El motor de búsqueda maneja correctamente (No arroja ningún resultado) las búsquedas vacías (cuando no se introduce nada)? | **1** | |  |
| 50 | ¿Las etiquetas de navegación y links contienen las “palabras clave” que los usuarios necesitan para alcanzar su objetivo? | **01** | |  |
| 51 | ¿Los usuarios pueden ordenar y filtrar los resultados? | **1** | |  |
| 52 | ¿Los enlaces que invocan acciones (ej. descargas, nuevas ventanas) están claramente distinguidos de los links que cargan otras páginas? | **1** | |  |
| 53 | ¿La página de resultados de una búsqueda indica claramente cuántos resultados tuvo la búsqueda? | **0** | |  |
| 54 | ¿La página de resultados de una búsqueda no muestra resultados duplicados (ni duplicados reales ni duplicados muy parecidos)? | **1** | |  |
| 55 | ¿La caja de búsqueda es suficientemente grande para manejar la longitud de las consultas más comunes? | **1** | |  |
| 56 | ¿Las búsquedas cubren todo el sitio, no una porción de él? | **1** | |  |
| 57 | ¿La interfaz de búsqueda está ubicada donde los usuarios esperan encontrarla (en la parte superior derecha de la página)? | **1** | |  |
| 58 | ¿Las imágenes tienen tamaños adecuados que no dificultan el acceso a las páginas? | **1** | |  |
| 59 | ¿Existe un cambio visible cuando el ratón apunta a algo clickeable (excluyendo los cambios de cursor)? | **1** | |  |
| 60 | ¿Todos los botones“clickeables” son efectivamente presionables? | **1** | |  |
| 61 | ¿Los íconos son visualmente y conceptualmente distintos pero mantienen una armonía? | **1** | |  |
| 62 | ¿Para tareas similares, los diálogos, formularios son similares? | **1** | |  |
| 63 | ¿Existe una clara distinción entre campos “requeridos” y “opcionales” en los formularios? | **0** | |  |
| 64 | ¿Las preguntas en los formularios están agrupadas de manera lógica y cada grupo tiene un título descriptivo? | **1** | |  |
| 65 | ¿Los campos en los formularios contienen ayudas, ejemplos o modelos de respuestas para demostrar el dato que se debe introducir? | **0** | |  |
| 66 | ¿El nombre de los botones de un formulario es adecuado, aplicado a la acción, no general (Ej.: utilizar “Enviar” en vez de “OK”)? | **1** | |  |
| 67 | ¿Las listas de opciones, botones de radio y casillas son preferibles a las cajas de texto en los formularios? | **1** | |  |
| 68 | ¿Se mantiene una navegación consistente y coherente en todas las pantallas? | **1** | |  |
| 69 | ¿La distribución y ubicación de los elementos estructurales que contienen las páginas se mantiene constante a lo largo de la aplicación? | **1** | |  |
| 70 | ¿Se usa la misma fuente para todos los navegadores? | **1** | |  |
| 71 | ¿Puede utilizarse en cualquier navegador? | **1** | |  |
| 72 | ¿Se mantiene una tipografía coherente en todo el sitio web? | **1** | |  |
| 73 | ¿Existen enlaces redundantes? Enlaces redundantes: Son enlaces con rótulos diferentes que llevan a una misma página. | **1** | |  |
| 74 | ¿El link al mapa del sitio aparece en todas las páginas del sitio? | **0** | |  |
| 75 | ¿Añade una descripción en las imágenes? | **1** | |  |
| 76 | ¿Señala claramente los vínculos a archivos PDF como tal? |  | | **x** |
| 77 | ¿Especifica el tamaño de los archivos PDF? |  | | **x** |
| 78 | ¿El sitio tiene una URL correcta, clara y fácil de recordar? | **0** | |  |
| 79 | ¿Tiene un tiempo de respuesta rápida? | **1** | |  |
| 80 | ¿Los vínculos llevan a donde prometen ir? | **1** | |  |
| 81 | ¿El sistema de navegación es amplio y sencillo (muchos ítems en un menú) en vez de un menú profundo? | **1** | |  |
| 82 | ¿Se usan nombres estandarizados (“mapa web”, “acerca de…”)? | **1** | |  |
| 83 | ¿Se proporciona un texto equivalente para todo elemento no textual, tales como imágenes, para explicar su contenido a discapacitados visuales? | **1** | |  |
| 84 | ¿El documento está estructurado para que pueda ser leído con o sin una hoja de estilo, utilizando adecuadamente los tags de HTML? | **1** | |  |
| 85 | ¿Las presentaciones multimedia, en caso de existir están sincronizadas con sus subtítulos? |  | | **x** |
| 86 | ¿El cambio de idioma en los textos está identificado?  El texto que se encuentre en otro idioma debe estar en cursiva o en un formato diferente al del resto del texto. |  | | **x** |
| 87 | ¿El sitio funciona correctamente en un lector de pantalla y/o navegador de voz? |  | | **x** |
| 88 | ¿Es posible la navegación sin ratón? | **0** | |  |
| 89 | ¿Se han creado atajos de teclado? |  | | **x** |
| 90 | ¿Todos los estilos se han creado en hojas CSS? | **1** | |  |
| 91 | ¿Cumple el sitio con el principio de usabilidad de realizar las operaciones con un máximo de tres click? | **1** | |  |
| 92 | ¿Existe suficiente contraste entre el color del fondo y el del texto? | **1** | |  |
| 93 | ¿Los tipos y tamaños de letra son legibles y distinguibles? | **1** | |  |
| 94 | ¿Añade color de fondo a los div que llevan imagen de fondo? (Para los usuarios que desactivan las imágenes, desaparece el contraste entre texto y fondo, convirtiéndose en texto ilegible.) | **0** | |  |
| 95 | ¿Poseen las páginas animaciones innecesarias? | **1** | |  |
| 96 | ¿El sitio puede ser usado sin desplazamiento horizontal? | **1** | |  |
| 97 | ¿El uso de los colores es moderado? | **1** | |  |
| 98 | ¿Se usan los estilos (negritas, cursivas...) con moderación? Si todo está resaltado con negrita o cursiva, el cerebro se acostumbra y deja de parecerle destacado. | **1** | |  |
| 99 | ¿Utiliza la misma tipografía los mismos aspectos, ejemplo: mantener una tipografía estándar por elementos? | **1** | |  |
| 100 | ¿Utiliza un interlineado adecuado para una buena lectura? | **1** | |  |
| 101 | ¿Se usan frases breves y concisas: que resuman los puntos clave y vayan al grano? | **1** | |  |
| 102 | ¿Los párrafos son cortos? | **1** | |  |
| 103 | ¿Los textos están corregidos? | **1** | |  |
| 104 | ¿Resalta en negrita los conceptos principales de los textos densos? | **1** | |  |
| 105 | ¿Utiliza listas de boliche y numeradas? |  | | **X** |
| 106 | ¿El sitio evita el uso excesivo del texto en mayúsculas? | **1** | |  |
| 107 | ¿El texto dentro del sitio no debe estar justificado? | **1** | |  |
| 108 | ¿No se utiliza el texto subrayado en el cuerpo del texto para resaltar a menos que sea un hipervínculo? | **1** | |  |
| 109 | ¿Existe suficiente espacio entre los elementos de acción (links, botones, etc.) para prevenir que el usuario haga click en el elemento incorrecto? | **1** |  | | |
| 110 | ¿Hay ausencia de enlaces rotos o que no lleven a ninguna página? | **1** |  | | |
| 111 | ¿Se dan indicaciones para completar campos problemáticos? | **1** |  | | |
| 112 | ¿Los botones de acción, (tales como “Enviar”) siempre son invocados por el usuario y no automáticamente invocados por el sistema cuando el último campo de un formulario ha sido lleno? | **1** |  | | |
| 113 | ¿El espacio entre los campos del formulario es suficiente como para distinguirlos unos de otros? | **1** |  | | |
| 114 | ¿Se evita el contenido importante del sitio en ventanas emergentes? | **1** |  | | |
| 115 | ¿La zona de acción del vínculo esta ampliada para acertar a la primera? | **1** |  | | |
| 116 | ¿El buscador (si existe) permite errores tipográficos y ortográficos (tildes)? | **1** |  | | |
| 117 | ¿En caso de errores de consistencia dentro del sitio, ¿se ofrece un mensaje de personalizado mediante una página explicativa?, (Por ejemplo: Error 404 para página inexistente) | **1** |  | | |
| 118 | ¿El mensaje de error permite volver a la situación anterior? | **1** |  | | |
| 119 | ¿Entrega información de contacto con múltiples opciones dentro y fuera de Internet? (Por ejemplo: Formulario, Teléfono, Fax, Correo electrónico, Chat, y hasta Wave, Buzz, Skype…, teléfono institucional, mesa de ayuda) |  | **x** | | |
| 120 | ¿Ofrece área de Preguntas Frecuentes con datos de ayuda a usuarios? |  | **x** | | |
| 121 | ¿Ofrece páginas de ayuda que explican cómo usar el Sitio? |  | **x** | | |
| 122 | ¿Si la ayuda obliga a salir de la zona principal, se proporciona un medio para moverse entre esa ventana y la ayuda? |  | **x** | | |
| 123 | ¿La ayuda no interrumpe la tarea del usuario? |  | **x** | | |
| 124 | ¿El sitio está diseñado para necesitar el mínimo de ayuda y de instrucciones? |  | **X** | | |
| 125 | ¿La ayuda está organizada en pasos? |  | **x** | | |
| 126 | ¿Se dan ejemplos para facilitar la tarea? |  | **x** | | |
| 127 | ¿Se utilizan explicaciones cortas en la ayuda? |  | **x** | | |
| 128 | ¿Existe un vínculo a los datos de contacto en un lugar bien visible en todas las páginas web del sitio? | **1** |  | | |
| 129 | ¿Información necesaria en página de contacto? (Atención al cliente, Consultas, Soporte técnico, Solicitudes de empleo.) | **1** |  | | |
| 130 | ¿Existe una página con las indicaciones sobre la protección de datos en el sitio web? | **0** |  | | |
| 131 | ¿Incluye un vínculo a los datos legales en todas las páginas? | **0** |  | | |
| 132 | ¿Puede el usuario ponerse en contacto con el encargado del Sitio Web para hacer sugerencias o comentarios? | **1** |  | | |
| 133 | ¿Funcionan correctamente los formularios de contacto? | **1** |  | | |
| 134 | ¿Hay alguien encargado de recibir y contestar estos mensajes? | **1** |  | | |
| 135 | ¿El sitio soporta a los usuarios novatos y expertos brindando diferentes niveles de explicación? (ej. en páginas de ayuda y mensajes de error) | **1** |  | | |
| 136 | ¿La política de privacidad del sitio es fácil de encontrar, especialmente esas páginas que piden información personal? |  | **x** | | |
| 137 | ¿Cuándo existen múltiples pasos en una tarea, el sitio muestra todos los pasos que deben ser completados y provee una retroalimentación al usuario indicándole la posición actual en toda la ruta de la tarea? | **1** |  | | |
| 138 | ¿La funcionalidad de los controles para nuevos dispositivos es exactamente la misma que para los otros dispositivos? |  | **x** | | |
| 139 | Se defina de manera correcta gráficos y tablas utilizando atributos (leyendas, unidades de medida, etc.) | **1** |  | | |
| 140 | ¿Las partes o secciones más importantes de los sitios son accesibles desde la página de inicio? | **1** |  | | |
| 141 | ¿Las páginas no requieren volver a escribir la información solicitada en páginas anteriores? | **1** |  | | |
| 142 | ¿Existen aceleradores, accesos rápidos a operaciones frecuentes? |  | **x** | | |
| 143 | ¿El cursor se desplaza adecuadamente en un formulario al presionar “tabulador”? | **1** |  | | |
| 144 | ¿Se implementen validaciones antes de que el usuario envíe información? | **1** |  | | |

**Anexo 3. Lista de Chequeo de Seguridad**

|  |  |  |
| --- | --- | --- |
| Evidencia ¿Qué? | Nivel de Importancia | Respuesta 1/0 |
| ¿Los datos de autenticación del usuario son transferidos a través de un canal cifrado? | Alta | 1 |
| ¿Existen cuentas con contraseñas en blanco? | Alta | 0 |
| ¿Los datos de autenticación del usuario son transferidos a través del método POST? | Alta | 1 |
| ¿Se muestran a los usuarios errores específicos de validación como resultado de un acceso fallido? | Alta | 1 |
| ¿El usuario recibe información técnica de los errores? | Alta | 1 |
| ¿Las alarmas y notificaciones son enviadas a través de correo, ventana emergente, SMS, etc.? | Alta | 1 |
| ¿Se muestra una notificación de usuario inexistente desde una misma dirección IP? | Alta | 1 |
| ¿Se muestra una notificación con la cantidad de intentos fallidos al intentar iniciar sesión con contraseña incorrecta? | Alta | 0 |
| ¿El usuario puede cambiar la contraseña en su primer inicio de sesión? | Alta | 1 |
| ¿Existen los campos de usuario y contraseña para acceder al sistema? | Alta | 1 |
| ¿La aplicación permite identificar una respuesta diferente al utilizar un usuario válido con contraseña incorrecta? | Alta | 1 |
| ¿Existen cuentas con contraseña igual al nombre de usuario? | Alta | 1 |
| ¿Las contraseñas son una combinación de caracteres alfanuméricos, incluyendo caracteres especiales, mayúsculas y minúsculas? | Alta | 1 |
| ¿Las contraseñas tienen una longitud mínima de ocho caracteres? | Alta | 1 |
| ¿La cuenta del usuario se bloquea después de un número determinado de intentos fallidos? | Alta | 0 |
| ¿Existe un formulario CAPTCHA o reCAPTCHA en los formularios de registro? | Alta | 0 |
| ¿Existe un mecanismo de bloqueo de usuario después de un máximo de seis intentos fallidos? | Alta | 0 |
| ¿Se requiere re-autenticación para operaciones críticas? | Alta | 0 |
| ¿El sistema permite al usuario acceder desde diferentes ubicaciones? | Alta | 1 |
| ¿Se utiliza el protocolo SSL para proteger la comunicación cliente-servidor? | Alta | 1 |
| ¿El usuario puede intentar iniciar sesión si ya tiene una sesión activa en el servidor? | Alta | 1 |
| ¿Se permite configurar las contraseñas con un tiempo de caducidad de hasta noventa días? | Alta | 0 |
| ¿El cambio de contraseña está permitido solo validando la contraseña anterior? | Alta | 1 |
| ¿Es posible acceder directamente a páginas protegidas a través de la barra de direcciones del navegador? | Alta | 1 |
| ¿La aplicación permite el acceso a recursos al modificar algún parámetro? | Alta | 1 |
| ¿La aplicación cierra automáticamente la sesión del usuario tras inactividad durante un tiempo? | Alta | 1 |
| ¿Es posible acceder nuevamente a la aplicación si se copia la URL después de cerrar sesión? | Alta | 1 |
| ¿Al cerrar sesión, se puede acceder presionando 'Atrás' en el navegador? | Alta | 1 |
| ¿Los datos sensibles son enviados al servidor en mensajes HTTP? | Alta | 1 |
| ¿Está deshabilitada la caché del lado del cliente para datos sensibles en formularios? | Alta | 1 |
| ¿Existe un botón visible de cierre de sesión en todas las páginas que requieren autenticación? | Alta | 1 |
| ¿Es posible acceder a recursos protegidos sin autenticación? | Alta | 1 |
| ¿Es posible acceder a funciones reservadas para perfiles o privilegios específicos? | Alta | 1 |
| ¿Un usuario estándar puede acceder a funciones administrativas? | Alta | 1 |
| ¿Un usuario estándar puede modificar sus privilegios asignados? | Alta | 1 |
| ¿Un usuario estándar puede modificar los privilegios de otro usuario? | Alta | 1 |
| ¿Un usuario administrador puede modificar sus propios privilegios? | Alta | 1 |
| ¿Un usuario administrador puede eliminar su propia cuenta? | Alta | 1 |
| ¿El control de acceso está asegurado del lado del servidor? | Alta | 1 |
| ¿Está deshabilitada la navegación por directorios? | Alta | 1 |
| ¿El usuario tiene privilegios de DBA? | Alta | 1 |
| ¿El usuario tiene los privilegios necesarios para usar el sistema? | Alta | 1 |
| ¿Existe un mecanismo de administración de seguridad que prevenga el uso de herramientas de desarrollo? | Alta | 0 |
| ¿Existe un mecanismo para la gestión de usuarios, privilegios y trazabilidad? | Alta | 1 |
| ¿Los usuarios acceden solo a los recursos para los cuales están autorizados? | Alta | 1 |
| ¿Es posible activar, desactivar o bloquear cuentas ante requisitos de negocio o posibles ataques? | Alta | 1 |