

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
ESCUELA DE SISTEMAS



TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN SISTEMAS Y COMPUTACIÓN

ANÁLISIS, DISEÑO Y DESARROLLO DE UN SISTEMA DE
INFORMACIÓN WEB PARA AUTOMATIZAR LOS PROCESOS DE
COMPRAS, INVENTARIOS Y VENTAS (E-COMMERCE).

CASO DE ESTUDIO: COMPUNEX.

AUTORES:

CISNEROS JÁCOME DANIELA ESTEFANÍA

VACA HERNÁNDEZ JORGE DAVID

DIRECTOR: ING. JAVIER CÓNDOR MTR.

QUITO JULIO, 2022

Contenido

Dedicatoria.....	5
Agradecimientos	6
Resumen.....	7
1. CAPÍTULO 1: Marco teórico.....	8
1.1 Conceptos, metodologías y herramientas TI.....	8
1.1.1 Sistema de información.....	8
1.1.2 Metodología de desarrollo	9
1.1.3 Arquitectura de software.....	12
1.1.4 Desarrollo Backend.....	14
1.1.5 Desarrollo Frontend	15
1.1.6 Base de datos.....	16
1.2 Conceptos de comercialización.....	19
1.2.1 Comercio.....	20
1.2.2 Comercio electrónico	20
1.2.3 Ventas	22
1.2.4 Inventario	23
1.2.5 Compras	25
2. CAPÍTULO 2: Situación actual.....	27
2.1 Antecedentes de la empresa	27
2.2 Situación Financiera.....	28
2.3 Situación Organizacional	29
2.4 Esquema de ventas	31
2.5 Proyección a comercio electrónico	33
3. CAPÍTULO 3: Análisis y diseño.....	34
3.1 Requerimientos Funcionales del Sistema de Información	34

3.1.1	Módulo Compras	35
3.1.2	Módulo Ventas.....	35
3.1.3	Módulo Inventario	36
3.1.4	Módulo Garantías.....	36
3.2	Requerimientos Funcionales del E-Commerce	37
3.3	Requerimientos No Funcionales	37
3.4	Requerimientos propios de todo Sistema de Información	38
3.5	Diseño de la base de datos.....	39
3.5.1	Módulo Compras	39
3.5.2	Módulo Ventas.....	40
3.5.3	Módulo Inventarios	42
3.5.4	Módulo Garantías.....	42
3.5.5	Gestión de permisos	43
3.6	Diseño de casos de uso.....	44
3.7	Diseño de interfaz.....	48
3.7.1	Sistema de Información	49
3.7.2	E-Commerce	53
4.	CAPÍTULO 4: Desarrollo y Pruebas.....	60
4.1	Administración del sistema	63
4.1.1	Sprint 1.....	64
4.1.2	Revisión del Sprint 1.....	75
4.1.3	Sprint 2.....	76
4.1.4	Revisión del Sprint 2.....	78
4.2	Módulo Inventario.....	78
4.2.1	Sprint 2.....	79
4.2.2	Revisión del Sprint 2.....	82
4.2.3	Sprint 3.....	83

4.2.4	Revisión del Sprint 3.....	86
4.3	Módulo Compras.....	86
4.3.1	Sprint 3.....	86
4.3.2	Revisión del Sprint 3.....	90
4.3.3	Sprint 4.....	91
4.3.4	Revisión del Sprint 4.....	93
4.4	Módulo Ventas y Garantías.....	93
4.4.1	Sprint 4.....	94
4.4.2	Revisión del Sprint 4.....	98
4.4.3	Sprint 5.....	99
4.4.4	Revisión del Sprint 5.....	102
4.5	E-Commerce.....	102
4.5.1	Sprint 5.....	103
4.5.2	Revisión del Sprint 5.....	106
4.5.3	Sprint 6.....	108
4.5.4	Revisión del Sprint 6.....	115
4.5.5	Sprint 7.....	117
4.5.6	Revisión del Sprint 7.....	120
5.	CAPÍTULO 5: Conclusiones y Recomendaciones.....	121
5.1	Conclusiones	121
5.2	Recomendaciones.....	122
	Bibliografía.....	124
	Anexos	127
	Listado de Figuras.....	128
	Listado de Tablas	131

Dedicatoria

A mis padres Eddy y Jeannet, por brindarme su amor y apoyo incondicional, pero sobre todo por ser mi fuente de inspiración y ejemplo a seguir. Esto es por y para ustedes.

A mi hermano, Juan Andrés, por aconsejarme, enseñarme, y ser mi compañía.

A mis mejores amigas, Emilia y Milena, por siempre escucharme y alegrar mis días a pesar de la distancia.

Daniela Estefanía Cisneros Jácome

A mi madre Silvana por todo el esfuerzo, la paciencia y el amor incondicional que me arropo. Siendo siempre mi fortaleza para solucionar cada reto u obstáculo que se presenta en mi vida. Que a pesar de todas las adversidades tiene la fuerza para sacarnos adelante.

A mi hermana, Karlita que siempre estuvo dándome su apoyo y cariño, sacándome una sonrisa en los momentos más tristes y complicados.

A mis abuelitos, Michita y Aurelio que me enseñaron la perseverancia, la humildad, y la importancia del estudio, quienes con su ejemplo me motivan a ser siempre mejor.

A mi padre, Jorge que me dejó grandes enseñanzas de vida.

A mi novia, Pame por todo ese amor, cariño y comprensión que me ha brindado durante momentos tristes y felices en mi vida, siendo mi compañía y apoyo.

Y es gracias a ustedes que hoy puedo alcanzar mi meta.

Jorge David Vaca Hernández

Agradecimientos

A mi padre, por darme la oportunidad de estudiar en esta universidad y por enseñarme a soñar en grande.

A mi madre, por siempre estar pendiente de mí y alentarme a seguir cuando quería rendirme.

A mi hermano, abuelitos y demás familia, por sus palabras de motivación y cariño constante.

A mis amigos, por haber compartido conmigo tantas experiencias, por hacerme reír y hacer divertida esta etapa de mi vida.

A mis profesores y en especial a mi tutor de tesis, por haberme impartido su conocimiento tanto en el ámbito profesional como en lo personal.

Y a todos aquellos que de alguna forma intervinieron en mi formación académica y en la realización de este trabajo.

Daniela Estefanía Cisneros Jácome

A mi madre quien a pesar de todas las adversidades ha sido el pilar fundamental en mi vida junto con mi hermana, siempre llenándome de motivación y apoyo incondicional.

Mi gratitud a mi director de trabajo de titulación el Ingeniero Javier Córdor que me direcciono, acompaño e instruyo con su sabiduría y conocimiento. También a mi compañera y amiga Daniela Cisneros con quien realizamos este trabajo.

A mis Docentes por impartirme gran parte del conocimiento que con su paciencia, enseñanza y experiencia me prepararon para llevar a cabo el logro de una vida Profesional.

Jorge David Vaca Hernández

Resumen

Durante los últimos años, el desarrollo web y el surgimiento de nuevas plataformas digitales han transformado drásticamente el funcionamiento tradicional de los negocios. La automatización de procesos internos junto con la comercialización de productos y servicios a través de internet, permiten a las empresas tomar ventaja para llegar a sus clientes de una forma más rápida, eficiente y eficaz. Por tal motivo, el presente trabajo de titulación plantea el análisis, diseño y desarrollo de un sistema de información web y e-commerce basado en la metodología SCRUM para automatizar los procesos de la empresa COMPUNEX. Este proyecto brindará la facilidad de contar con un sistema que le permita gestionar sus módulos de compras, ventas e inventarios y a su vez, disponer de un comercio en línea para la venta de sus distintos productos. Todo esto con el fin de que se consiga llevar un mejor manejo y control de la información y al mismo tiempo, pueda expandir el alcance de su mercado.

1. CAPÍTULO 1: Marco teórico

El siguiente capítulo abarcará todos aquellos conceptos teóricos que son fundamentales para la realización del presente trabajo de disertación. Comprende las metodologías, *frameworks* y herramientas de desarrollo de software que serán utilizadas, siendo la metodología Scrum el principal marco de trabajo. Asimismo, se definirán términos importantes relacionados con el manejo de compras, inventarios y ventas, incluyendo el comercio electrónico.

1.1 Conceptos, metodologías y herramientas TI¹

Las metodologías y herramientas de TI son esenciales para el desarrollo de proyectos de software debido a que permiten que estos se implementen de manera más rápida y organizada. Se presentan como marcos de trabajo que posibilitan la obtención de ventajas competitivas para quienes los aplican, ya que gracias a estos se pueden atender los requerimientos de los clientes en menor tiempo y ser más flexibles y adaptables ante los cambios externos que se presenten.

1.1.1 Sistema de información

Los sistemas de información (SI²) se han convertido en uno de los principales ámbitos de estudio que existen hoy en día en el área de organización de las empresas. Comprende un conjunto de procedimientos y funciones orientadas a la recopilación, elaboración, evaluación, almacenamiento y distribución de información. Estos sistemas son de gran importancia para todos los procesos que involucran la toma de decisiones, debido a que brindan mayores oportunidades de éxito y, sobre todo, su uso otorga una ventaja significativa frente a otras empresas. De igual manera, los SI ayudan a romper las brechas geográficas existentes, abriendo

¹ TI: Technology Information

² SI: System Information

la posibilidad de que pequeñas y grandes organizaciones evolucionen hacia una nueva era digital con el fin de ser más eficientes (Abrego, Sánchez, & Medina, 2016).

1.1.2 Metodología de desarrollo

Estas metodologías se presentan como un marco de trabajo eficiente enfocado en el desarrollo de software que surgió en la década de los años 70. Previo a su aparición, la implementación de cualquier tipo de software era prácticamente empírica y artesanal, provocando que la mayor parte de proyectos fracasaran en cumplir sus objetivos y sobre todo en satisfacer las necesidades de sus usuarios. Asimismo ocurrían múltiples problemas con entregas fuera del tiempo establecido y en repetidas ocasiones con presupuestos excedidos como resultado de dicho desarrollo. Por lo tanto, muchas de las organizaciones desarrolladoras que sobrevivieron a esta denominada “Crisis del Software” se encontraron con la necesidad de adoptar nuevos modelos y metodologías clásicas que fueron progresivamente incorporando estándares, controles y formalidades al desarrollo de software (Zumba, 2018).

En la actualidad existen distintas metodologías de desarrollo de software que son utilizadas según los requerimientos del sistema a implementar y debido a esto, no es posible definir si una metodología es mejor que otra, ya que cada una posee diferentes escenarios en donde se pueden destacar. Entre las metodologías más conocidas se encuentran las clásicas o tradicionales, que han evolucionado para adaptarse a las exigencias actuales; y las metodologías ágiles que se utilizan para desarrollos rápidos y con gran escalabilidad donde se destacan principalmente SCRUM y *Extreme Programming* (XP) (Rolando, Zea, Contento, & García, 2017).

1.1.2.1 Scrum

El marco de trabajo Scrum fue diseñado para el desarrollo ágil de software y simultáneamente para fomentar la colaboración y participación de equipos en proyectos complejos. Scrum define un conjunto de reglas y prácticas asociadas con roles, las mismas que cumplen un propósito específico para garantizar el éxito de los procesos que se ejecutan. Se encuentra basada en una serie de ciclos iterativos e incrementales cuyo enfoque principal gira en torno a tres pilares: transparencia, inspección y adaptación (Navarro, Fernández, & Morales, 2013).

Los equipos Scrum son multifuncionales y autogestionados, lo que les otorga flexibilidad y principalmente adaptabilidad para llevar distintas formas de trabajo. Están formados principalmente por tres roles siendo estos: el Scrum Máster o director del proyecto; quien es el líder y se asegura de que se adopte la metodología tal cual es establecida, brindando apoyo a todos los miembros del equipo. El **Product Owner** o dueño del producto quien se encarga de representar a los interesados externos e internos y es la voz del cliente. Por último, el **Development Team** o equipo de desarrollo quienes son los encargados de generar las iteraciones funcionales del producto (Zeballos, 2012).

Scrum establece un evento principal denominado **sprint** que corresponde a un periodo de tiempo de trabajo en el que se desarrolla una versión utilizable e incrementable del proyecto, pudiendo durar este tiempo entre una a máximo cuatro semanas. En la figura a continuación se observan todos los eventos y artefactos que forman parte de una sola iteración, los mismos que serán detallados más adelante.

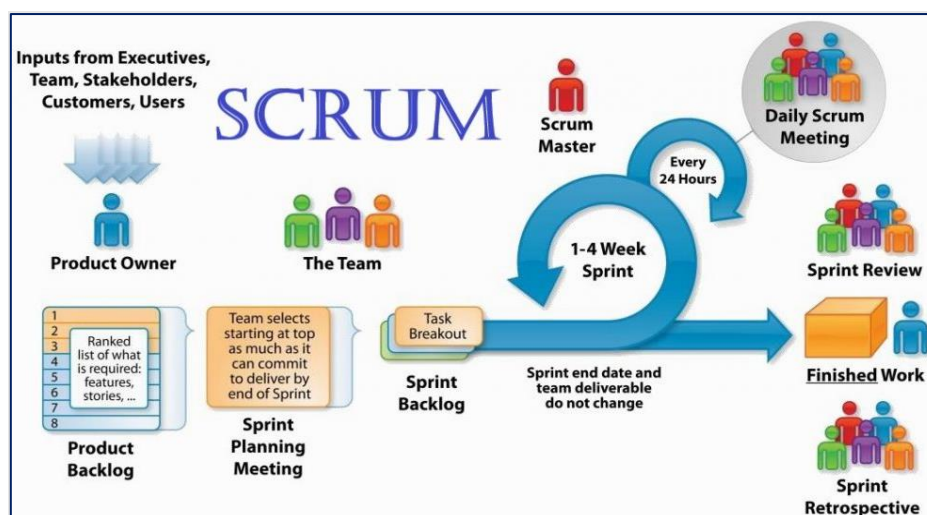


Figura 1-01. Estructura de un sprint (Apeiron Software, 2019)

Entre los eventos más importantes que suceden durante un *sprint* se encuentran:

- ***Sprint Planning***. Corresponde a la planeación del trabajo en donde se define el diseño del sistema y se estima las horas que puede tardar en completarse dicho *sprint*, es decir, se establece qué se va a entregar y cómo se lo logrará.
- ***Daily Scrums***. Son pequeñas reuniones de no más de 15 minutos en las que se detalla brevemente todos los avances desde la última reunión, lo que se planea conseguir en la siguiente y se informa si es que se ha presentado algún inconveniente durante el desarrollo del trabajo.
- ***Sprint Review***. Siempre ocurre al final del *sprint* y es en esta etapa en la que se verifica el cumplimiento o no de todas las actividades, se presenta el producto terminado y su correspondiente funcionamiento.
- ***Sprint Retrospective***. Se realiza con el objetivo de obtener una retroalimentación por parte de todo el equipo *Scrum* para analizar el desempeño general del *sprint* y se elabora un plan de mejora para las siguientes iteraciones.

Por otro lado, **Scrum** también comprende una serie de documentos o artefactos que permiten llevar un mejor control y transparencia de las tareas que se ejecutan dentro de cada **sprint**. Uno de estos conocido como **Product Backlog**, es una lista de todos los requerimientos definidos por el dueño del producto. En ella se definen tanto características como funciones de los productos, así como los distintos cambios que puedan presentarse para futuras entregas, convirtiéndose así en una lista dinámica que se encuentra siempre en evolución. Otro artefacto importante es la lista de pendientes o **Sprint Backlog** que contiene todos los requerimientos de alta prioridad del **Product Backlog** y aquellos que quedaron sin terminarse de iteraciones anteriores (Schwaber & Sutherland, 2017).

1.1.3 Arquitectura de software

Corresponde a la organización y estructura de un sistema con todos sus componentes y sus relaciones, el ambiente, y los principios que rigen su diseño. Es una parte fundamental para alcanzar el cumplimiento de los requerimientos tanto funcionales como no funcionales del sistema.

En la actualidad, existen distintos patrones de arquitectura que son aplicados dependiendo de la necesidad con la que se desarrollen los sistemas. Entre los más comunes se encuentran la programación por capas, el modelo-vista-controlador, el patrón cliente-servidor y la arquitectura de microservicios. La elección adecuada de la arquitectura de software permite una mejor comprensión y comunicación sobre todo para aquellos sistemas complejos, posibilitando la evolución de estos.

Para el presente trabajo se utilizará la arquitectura basada en microservicios que será explicada con más detalle a continuación.

Arquitectura basada en microservicios

Está enfocada en el diseño de software como un conjunto de pequeños servicios que se ejecutan de forma independiente aunque cooperan unos con otros a través de protocolos livianos de comunicación como HTTP³. Cada servicio proporciona funcionalidad y soporte a las distintas capacidades del negocio mientras se despliega de forma automatizada.

Esta arquitectura se diferencia de los patrones tradicionales o monolíticos ya que promueve el desarrollo de aplicaciones compuestas por unidades autónomas, modulares y autocontenidas. Su ventaja principal recae que para la implantación de cada servicio se puede optar por diversos lenguajes, tecnologías y bases de datos según la función para la que sean requeridas (López & Maya, 2017).

Tal como se observa en la siguiente figura, el manejo de los datos se realiza de forma descentralizada, a través de la separación del modelo de datos de forma análoga en microservicios que se comunican por medio de una interfaz bien definida mediante API ligeras.

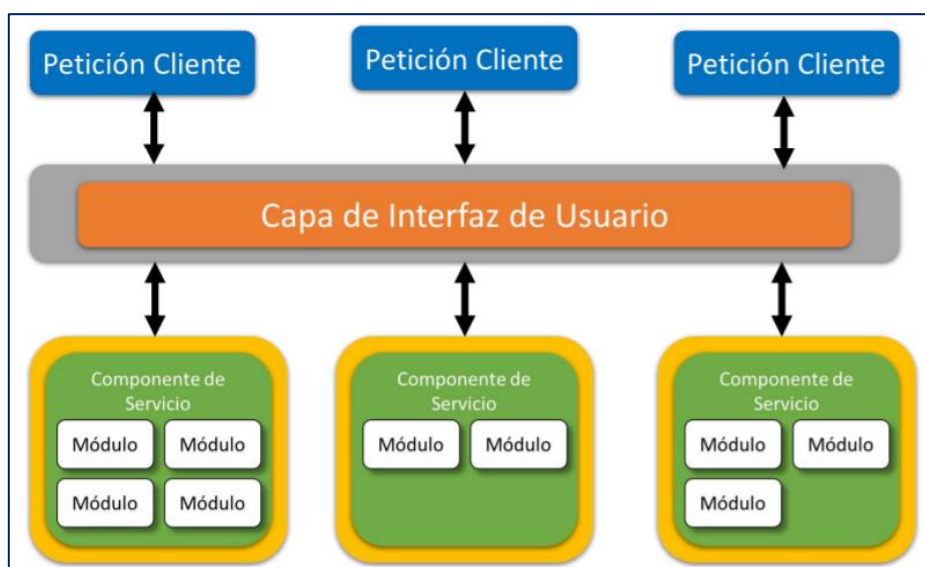


Figura 2-01. Patrón básico de la arquitectura de microservicios. (López & Maya, 2017)

³ HTTP: Hypertext Transfer Protocol

Gracias a la independencia de cada microservicio, se presenta una fuerte resistencia a errores ya que cualquiera de estos puede operar y sobre todo escalar, sin tener que alterar el desempeño de los otros. Es decir, si se presenta un error en algún servicio no es necesario detener toda la aplicación, sino solamente el componente afectado.

1.1.4 Desarrollo Backend

El **backend** corresponde a la capa de acceso a los datos y la lógica que se encarga del funcionamiento de las páginas web; es decir, de lo que se encuentra oculto a la vista de los usuarios. Por lo tanto, un desarrollador de **backend** proporciona toda la información que solicita un cliente cuando este realiza una petición a una página a través de un servidor (Márquez, 2020).

La utilización del **backend** dentro de distintas soluciones web permite determinar la eficiencia con la que se ejecutarán las aplicaciones. De igual manera, proporciona la experiencia que el usuario final tendrá, pudiendo esta ser positiva o negativa, al interactuar con páginas o sistemas web ya que es aquí donde interviene la optimización de recursos, la seguridad del sitio y otros factores manejados en la capa de **backend**.

Un web **framework** es una herramienta de programación de alto nivel, denominada web porque se obtiene una página web como interfaz para el usuario. Comprende una estructura de código organizado que permite la reutilización de sus componentes y es usada principalmente para el desarrollo rápido de aplicaciones. Por lo general, se lo utiliza con tecnologías como **JavaScript**, **CSS**⁴, **HTML**⁵, **XML**⁶ y **PHP**, independientemente del lenguaje de programación.

⁴ **CSS**: Cascading Style Sheets

⁵ **HTML**: Hypertext Markup Language

⁶ **XML**: Extensible Markup Language

A continuación, se expondrá con mayor profundidad sobre el *framework FastAPI* que será la herramienta utilizada para el desarrollo del *backend* del presente trabajo.

FastAPI

Es un web *framework* basado en Python 3.6+ que se utiliza para construir *APIs*⁷ de *RESTful* de forma rápida y sencilla. Este marco de desarrollo brinda un alto rendimiento y trabaja en conjunto con lenguajes como *Go* y *NodeJS*, que permiten desarrollos muchos más veloces con un incremento entre 200% y 300% en cuanto a rapidez. Una ventaja muy grande sobre su competencia es su buena documentación, facilidad de usar y aprender; permitiendo así disminuir tanto el tiempo de desarrollo como los errores humanos (FastAPI, 2022).

1.1.5 Desarrollo Frontend

Se denomina *frontend* a las tecnologías que trabajan del lado del cliente y que se enfocan en todo aquello con lo que los usuarios pueden visualizar o interactuar a través de la web (Vázquez del Cerro, 2020). El *frontend* está conformado principalmente por las tecnologías HTML, CSS y *JavaScript*, a través de las cuales puede presentar información al cliente de forma más estilizada. Además, busca implementar interfaces gráficas con un adecuado diseño de interacción para otorgar una buena experiencia a todos sus usuarios.

Usualmente, un desarrollador de *frontend* suele trabajar en conjunto con diseñadores quienes se encargan de crear la representación visual que los programadores traducirán en código; para que, posteriormente aquellos desarrolladores *backend* añadan funcionalidad a esos diseños conectándolos a sistemas como base de datos. Por lo que podemos apreciar que tanto *backend* como *frontend* necesitan del otro para funcionar correctamente.

⁷ **API:** Application Programming Interface

Se expondrá con más detalle sobre el *framework Nuxt.js* que será la herramienta utilizada para el desarrollo del *frontend* del presente trabajo.

Nuxt.js

Nuxt es un marco de desarrollo web *Node.js* de alto nivel y de código abierto, utilizado para crear aplicaciones que pueden ser implementadas e integradas tanto en modo universal (SRR⁸) o en aplicaciones de una sola página (SPA⁹). Sin embargo, su potencialidad reside en el desarrollo en los modelos universales debido a que puede ejecutar código *JavaScript* en el lado del cliente como en el del servidor (Tiam Kok, 2020).

Es un *framework* creado en base a una arquitectura modular que mediante el uso de distintos módulos o *plugins* permite desarrollar de manera sencilla y veloz sin necesidad de reutilizar código. Gracias a sus mejores prácticas basadas en *Vue.js* y *Node.js*, sus aplicaciones se encuentran optimizadas ya que incluye un analizador de paquetes y oportunidades para ajustar en tiempo de ejecución. Del mismo modo, presenta mensajes para la prevención de errores y soluciones atractivas predeterminadas para que la experiencia del desarrollador sea más agradable (Nuxt.js, 2021).

1.1.6 Base de datos

Se mencionó por primera vez al término base de datos en una conferencia realizada en California donde se abordaba un problema relacionado con el manejo y almacenamiento de los datos. Acuñaron a este como un grupo o conjunto de información para el manejo de grandes cantidades de datos relacionados entre sí. De acuerdo con (Pulido, Escobar, & Núñez, 2019),

⁸ **SRR**: Server-side rendering

⁹ **SPA**: Single page applications

una base de datos (BD¹⁰) no es más que un conjunto de datos organizados, de fácil acceso y con posibilidad de actualización.

En el ámbito informático, se la conoce como un sistema formado por un conjunto de datos almacenados ya sea en un computador o servidor y por una serie de programas que le permitan manipular dichos datos. Entre estos programas, se encuentran los sistemas para la gestión de bases de datos. Este tipo de software se encarga de manejar la interfaz con la base de datos, el usuario y las aplicaciones con la que este interactúa. Actualmente, estas bases de datos han generado un impacto decisivo tanto en el sector de la tecnología, como en el mundo en general ya que son utilizadas cada vez más por una mayor cantidad de industrias (Martínez, 2019).

La estructura de una base de datos está compuesta por varios elementos siendo el modelo de datos uno de los más importantes. Este corresponde a una agrupación de herramientas conceptuales que describen la naturaleza de los datos y su semántica. Además, permite relacionar y crear restricciones de integridad y consistencia (Bertone & Thomas, 2017).

Actualmente existe una gran variedad de bases de datos disponibles, cuyo uso en particular depende del propósito y necesidad que se desea satisfacer, encontrándose entre las más utilizadas las siguientes:

- Bases de datos relacionales: Son centradas en las relaciones que tiene todo un conjunto de datos. Están organizadas en tablas con columnas y filas, cuyas tablas se relacionan entre sí. Además, estas bases de datos representan la forma más eficiente de acceder a información estructurada.
- Bases de datos orientadas a objetos: Se basa en la abstracción de objetos del mundo real guardados en forma de datos. Están basadas en la programación orientada a objetos.

¹⁰ **BD:** Database

- Bases de datos distribuidas: Consiste en el almacenamiento de dos o más archivos ubicados en diferentes locaciones y que constituyen una base de datos. Generalmente están almacenadas en un conjunto de computadores, ubicadas en la mismo lugar físico o dispersas en diferentes redes.
- Bases de datos NoSQL: Conocidas también como NoSQL, son bases de datos no relacionales que permiten que los datos no estructurados y semiestructurados se almacenen y manipulen, sin necesidad de contar con campos preestablecidos como sucede con las bases de datos relacionales. Las bases de datos NoSQL fueron ganando popularidad a medida que las aplicaciones web se hacían más comunes y complejas.

Por otro lado, existen otros tipos de bases de datos no tan comunes que se centran en funciones científicas, financieras, aeroespaciales, entre otras. Estas se desarrollan bajo nuevas tecnologías junto a importantes avances basados en la nube y automatización, enfocando a las bases de datos en direcciones tecnológicas totalmente nuevas. (Oracle, 2021).

Modelo entidad-relación

En 1976, Peter Chan introdujo el modelo conceptual para el diseño de base de datos relacionales siendo este una representación básica de entidades, relaciones y atributos. Posteriormente, fue mejorado y actualizado gracias a que se agregaron otros conceptos tales como jerarquías de generalización, denominándolo así un modelo ampliado entidad-relación. Este es un modelo de datos de alto nivel que se basa en la abstracción de sus interrelaciones en el mundo real. Consiste en una colección de objetos básicos denominados entidades, y de relaciones entre estos objetos, los cuales cuentan con características llamadas atributos.

PostgreSQL

Es un sistema de gestión y motor de base de datos basado en los objetos y sus relaciones. Está distribuido bajo licencia BSD¹¹ y es un producto de código fuente libre. Es considerado por muchos el sistema más potente del mercado. Se encuentra basado en la estructura cliente-servidor con la finalidad del utilizar y gestionar multiprocesos logrando garantizar estabilidad del sistema puesto que un fallo en uno de sus procesos no afectará al resto del sistema, sino que continuará funcionando. Además, permite la manipulación de datos basado en el algebra relacional (Zea, Molina, & Redrován, 2017).

PostgreSQL presenta las siguientes características principales:

- Propiedades ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad)
- Replicación asincrónica/sincrónica
- **Backups** de seguridad en caliente
- Acceso encriptado a través de SSL¹²
- Multiplataforma (**Linux, Windows, MAC OS, Unix** y sus variantes)
- **Savepoints** para realizar transacciones y regresar a un punto anterior si es necesario.

1.2 Conceptos de comercialización

La comercialización se define como el conjunto de procesos que se desarrollan desde que el productor genera un producto hasta que este le llega al consumidor, mediante un mercado determinado. Es un mecanismo de coordinación para el intercambio entre diferentes agentes que participan en la cadena productiva y en ella intervienen varias funciones como la venta, compra e inventarios que serán detalladas con más profundidad a continuación.

¹¹ **Licencia BSD:** licencia de software otorgada para los sistemas Berkeley Software Distribution

¹² **SSL:** Secure Socket Layer

1.2.1 Comercio

Es toda transacción económica que involucra la compra y venta de bienes, ya sean para su futura utilización, venta o transformación. Hoy en día, el comercio constituye una de las funciones primordiales dentro de las empresas puesto que lleva a cabo las actividades necesarias para la venta de artículos o servicios; y es quien permite establecer o fortalecer las relaciones con los clientes. De igual manera, es de gran importancia debido a que, a través de este departamento de comercio, se dirigen y controlan otros aspectos involucrados en el área de ventas y compras. Es por esta razón, que es fundamental realizar una adecuada planificación comercial acorde con los lineamientos de cada organización con el fin de cumplir sus objetivos empresariales fijados (Valdivia, 2015).

1.2.2 Comercio electrónico

El término comercio electrónico o *e-commerce* surge gracias a la masiva expansión del internet junto con el flujo de información transfronteriza de los últimos años. Esto se debe a los múltiples cambios que ha experimentado la comercialización de productos y servicios por modificar su tradicional forma de operación de compras y ventas al adoptar nuevas alternativas tecnológicas que se encuentran siempre a la vanguardia del mercado. (Meltzer, 2018).

Se define a este como todo proceso de compra y venta de productos, bienes o servicios que emplean las tecnologías de la información y medios electrónicos como el internet. Surgió a partir de la creciente necesidad de las organizaciones de adoptar el uso de la informática para aplicar las nuevas tecnologías y de esta manera lograr afianzar la relación entre cliente y proveedor. El *e-commerce* comprende los procesos a través de los cuales se llega a los consumidores involucrando diversas actividades como ventas, marketing, recepción de órdenes de pedido, entregas y otros servicios de administración. Además, se caracteriza principalmente

por su alta disponibilidad e interactividad proporcionando así a sus clientes mayores ventajas en cuanto a comodidad y personalización se refieren. (Basantes, y otros, 2016).

Tipos de comercio electrónico

Al comercio electrónico se lo puede clasificar en 3 grandes grupos según su naturaleza, su tipo de flujo o por el tipo de transacción. Sin embargo, los más utilizados recaen dentro del último grupo mencionado y lo más importantes serán descritos a continuación.

- Comercio electrónico B2B

El comercio electrónico B2B o *Business to Business* corresponde a todas las transacciones económicas y comerciales efectuadas entre empresas que utilizan internet. No existe intervención alguna por parte de consumidores para realizar ventas o negociaciones, simplemente se encuentran los negocios del mercado. Gracias a que hacen uso de distintas tecnologías logran obtener múltiples beneficios como mayor eficiencia y aumento de ventas, ahorro de tiempo y costo, pero aún más importante logran adentrarse en el mercado.

- Comercio electrónico B2C

El comercio electrónico *Business to Consumer* es actualmente el más utilizado a nivel mundial por aquellas personas que desean comprar productos a través de tiendas virtuales de internet. Es en este tipo de negocio en línea en donde las organizaciones intentan llegar a un consumidor específico con el objetivo de vender sus bienes o servicios. Se benefician gracias a la comodidad y rapidez con la que se efectúan las compras, por la constante actualización de los sistemas, y por la facilidad en la que pueden ofrecer sus servicios sin importar la distancia.

- Comercio electrónico C2C

El comercio electrónico C2C o *Consumer to Consumer* se deriva de los tipos de comercio en línea ya existentes y es el que más ha evolucionado gracias al uso de redes sociales. Son interacciones comerciales efectuadas entre consumidores, en donde el consumidor final adquiere de un consumidor inicial los productos que por algún motivo este ya no necesita. Es mayormente utilizado por pequeños comerciantes gracias a su fácil accesibilidad y reducidos costos de aplicación.

Sin importar el tipo de comercio electrónico, este puede ser usado en distintos entornos pues se encarga de proporcionar todos los medios necesarios para que se logre la negociación entre los consumidores considerando vínculos y aspectos socioculturales. No obstante, a pesar de las ventajas que ofrece, se presentan ciertas limitaciones pudiendo estas ser técnicas como los costos elevados y la dificultad de integración con los procesos tradicionales de las empresas, o limitaciones no técnicas refiriéndose especialmente a aquellas vulnerabilidades de fraude u otros delitos que se realizan por este medio. Por lo tanto, es decisión de cada empresa el adentrarse o no en este mundo tecnológico. (Pesántez, Romero, & González, 2020).

1.2.3 Ventas

Se considera una venta a todo proceso aquel en el que un vendedor se encarga de la satisfacción de las necesidades de un cliente de tal forma que se obtenga un beneficio continuo para ambas partes (American Marketing Association, 2021). Comprende todo tipo de interacciones interpersonales para realizar transacciones de intercambio con compradores existentes o potenciales. Son una de las actividades primordiales para aquellas organizaciones que se dedican a ofrecer tanto productos como servicios dentro un nicho de mercado específico. El

éxito de estas dependerá de la efectividad con la que realicen este proceso y de las técnicas o estrategias empleadas para su ejecución.

Los procesos de venta se pueden agrupar en dos grandes tipos, siendo estos: presenciales y no presenciales. Los primeros corresponden a aquellos desarrollados en un ambiente más reducido ya que requieren de la presencia obligatoria tanto del vendedor como del comprador. Por lo general, suelen ser rápida, inmediatas e incentivan la comunicación interpersonal. Se caracterizan por efectuarse en distintos lugares comerciales en donde es el cliente quien se dirige a estos en busca de la adquisición de algún artículo en específico y dependiendo de este puede probarlo, degustarlo y observarlo con más detalle antes de realizar su compra. Por otro lado, se encuentran las ventas no presenciales que utilizan otros canales de comunicación especialmente medios digitales para contactarse con sus clientes. Entre estos medios están el correo electrónico, la telefonía móvil, la televisión y el más importante el internet, donde aparece la más destacada venta en línea.

Este tipo de venta ha crecido incontrolablemente en los últimos años ya que abarca ámbitos a nivel mundial y engloba cada vez más a muchos más compradores. En esta se ofrecen productos en sitios web de internet y son los compradores quienes pueden visualizar los distintos artículos mediante imágenes para después efectuar la compra online. Es decir, cancelar por su producto a través de plataformas de pago online para que posteriormente dicho artículo se lo sea entregado, ya sea de forma digital o presencial (Arenal Laza, 2017).

1.2.4 Inventario

Dentro de todas las empresas, los inventarios constituyen un activo de vital importancia dentro de las transacciones continuas que representan la base para las operaciones de empresas sin importar la actividad económica en la que se desarrollen; pudiendo estas ser diferentes ámbitos

como agrícolas, construcciones, industriales, comerciales, etcétera. El correcto manejo de inventarios contribuye directamente al éxito de las empresas, ya que si se lo lleva con una gestión incorrecta podría llevar a la quiebra de dichos negocios.

Existen varios tipos de inventario, y según múltiples fuentes se los puede clasificar de muchas maneras, cada uno con un enfoque diferente y distintos casos en los que se destacan. Entre los principales tenemos:

- Inventario promocional

Como su nombre lo indica se basa en las promociones y cómo aumentar su demanda, lo que genera que sea difícil predecir el cambio en la cantidad vendida en cualquier lugar dado, haciendo que sea imposible determinar cuánto inventario adicional es necesario. Por otro lado, para los productos que tienen poco movimiento, es decir que se venden uno o dos cada semana, una promoción podría ayudar que las ventas crezcan.

- Inventario minorista en bodega

Se encuentra situado en una tienda o bodega que esta al por menor y puede estar compuesto por un inventario en tránsito o un inventario a través de la bodega.

- Inventario de materias primas

Es utilizado cuando se trabaja materias primas para la producción. Se enfoca en la importancia del abastecimiento en el inventario, ya que al quedarse sin materia prima se detiene toda la cadena de producción llevando a pérdidas costosas.

- Inventario de producción en proceso

Permite gestionar el proceso para transformar la materia prima en un producto terminado. Se basa en distribuir en distintas estaciones de trabajo para que si falla una máquina la producción no se detenga.

- Inventario de productos terminados

Conlleva algunas dificultades ya que, por lo general, es el más costoso de mantener debido a que un producto ya fabricado puede deteriorarse, estropearse, volverse obsoleto, dañarse o ser robado.

- Inventario de piezas de repuesto

Se utiliza para el manejo de determinados partes o piezas del producto fabricado, las cuales son requeridas para el mantenimiento y reparación.

Dependiendo de la situación es aplicable cada tipo de inventario con sus ventajas y desventajas, permitiendo tener un mejor control sobre los futuros costos y beneficios (Esper & Waller, 2017).

1.2.5 Compras

Desde la revolución industrial, las empresas tomaron un rumbo muy diferente a su manera tradicional de operar debido a que se produjo el primer auge tecnológico. Desde ahí, las organizaciones se han visto obligadas a cambiar la manera de gestionar sus compras e inventarios, buscar los mejores proveedores del mercado y elaborar mejores análisis de la demanda y la cantidad de las ventas. De esta manera, logran cumplir el objetivo primordial de todo negocio de generar mayores ganancias.

Dentro del proceso de compra, el aprovisionamiento ya sea de materia prima, productos o servicios se realiza en base a distintas actividades que van más allá de una simple negociación y trámites administrativos. Estas actividades permiten a la organizaciones lograr buenas transacciones, ahorrar en costos y mejorar en su logística para así obtener la satisfacción del cliente por la calidad y el tiempo que puede ganar al trabajar con dicha empresa.

Estas deben estar provistas con el suficiente inventario antes y después de una venta, analizando el equilibrio entre lo que se necesita comprar y lo que se dispone en stock para mantener un

adecuado manejo de cualquier industria. Por lo tanto, toda empresa organizada y con buenas prácticas comerciales maneja su proceso de compra en base a metodologías e indicadores de gestión (Espinoza, 2018). Sin embargo, dependerá de cada negocio el determinar cómo efectuar el proceso de compra para obtener el máximo beneficio.

2. CAPÍTULO 2: Situación actual

El presente capítulo contemplará la situación actual de la empresa Compunex. Se mencionará su organización tanto administrativa como financiera, y también todos aquellos procesos principales que realiza para alcanzar sus objetivos empresariales como lo son la venta y mantenimiento de equipos y suministros tecnológicos. Esto permitirá analizar el presente entorno en el que se desenvuelve la organización y conocer aquellos aspectos en los que puede mejorar gracias a la incorporación de un sistema de información.

2.1 Antecedentes de la empresa

Compunex es un negocio establecido en el año 2001 en la ciudad de Quito. Surgió a partir de la visión tecnológica que tuvieron la pareja de fundadores Jorge Olavo Vaca y Silvana Marilú Hernández. Ellos identificaron la necesidad que la población de Quito manifestaba, ya que durante esos años comenzaba el auge tecnológico de las computadoras personales hacia un uso masivo en el Ecuador, así como también la adopción de la automatización a través del *software* y *hardware* en empresas públicas y privadas. Por este motivo, crearon la empresa Compunex, enfocada en la venta y soporte de equipos de computación tanto a personas como a empresas.

Esta se caracteriza por ofrecer a sus clientes los mejores artículos informáticos del mercado con precios asequibles y, sobre todo, por brindar un servicio de calidad. Cuenta con un personal profesional y altamente calificado para instalación de redes estructuradas, diseño de páginas web, entre otros.

Actualmente, dispone de registros manuales y limitados programas ofimáticos como *Word* y *Excel* para llevar a cabo el manejo y gestión de toda la información proveniente de sus procesos administrativos, los cuales son altamente susceptibles a daños o pérdidas y por consiguiente, vuelven deficiente el trabajo del personal y el servicio a sus clientes. Simultáneamente, el

negocio realiza todas sus ventas de manera presencial ofreciendo todos sus productos en una tienda física sin lograr abarcar un mayor segmento de mercado.

2.2 Situación Financiera

La salud financiera de esta empresa mejoró notablemente en el año 2020, debido a que experimentó un aumento en la ganancia bruta comparado con el 2019. Según Compunex, esto se debe a que la pandemia ocasionada por el virus SARS-CoV-2 adelantó y forzó a una transformación digital en todas las industrias y áreas, abriendo un mayor nicho de mercado.

La siguiente tabla refleja el estado financiero de la empresa correspondiente a los dos últimos años.

	2019	2020
Compras	\$113.424	\$175.368
Ventas en Efectivo	\$80.316	\$111.612
Ventas por Transferencia/Depósito	\$57.336	\$96.132
Ventas con tarjeta	\$19.152	\$52.404
Total de ventas	\$156.804	\$260.148
Internet	\$479,88	\$479,88
Servicios Básicos	\$685,68	\$740,52
Sueldos	\$9.600,00	\$9.600,00
Gastos	\$10.765,56	\$10.820,40
Utilidad bruta	\$32.614,44	\$73.959,60

Tabla 1-02. Estado financiero de Compunex. (Cisneros & Vaca, 2022)

Durante el año 2020, Compunex registró compras mensuales entre \$13.000 a \$16.000 teniendo un total de \$175.368 anuales los cuales han crecido en un 54,61% con respecto al año 2019, mostrando así un panorama favorecedor en cuanto a compras realizadas. Por otro lado, las ventas se han incrementado un 65,91% con respecto al 2019, siendo las ventas con tarjeta una

forma de pago que se potencializó durante la pandemia según la empresa, debido a la falta de dinero en efectivo por muchos de sus clientes.

Con la información proporcionada, la empresa busca mantener e incrementar ese margen de ventas, innovando y buscando nuevas soluciones que le permitan crecer.

2.3 Situación Organizacional

Compunex tiene definidos seis departamentos principales los cuales operan en conjunto para el mejor funcionamiento del negocio. Entre estos, el más importante corresponde a la dirección general, en donde se encuentran aquellos cargos de alto mando como lo son el gerente general y los cuatro sub directivos encargados de controlar y supervisar cada uno de los departamentos restantes. Todos estos tienen el objetivo primordial de asumir una tarea determinada del giro del negocio y cumplirla de forma autónoma con sus propios recursos.

Dentro de la organización se pueden encontrar los siguientes departamentos:

- Financiero

Es aquel encargado de gestionar aquellas funciones relacionadas con la contabilidad, es decir, del manejo del flujo de dinero. Principalmente, este departamento administra el presupuesto del negocio tanto para gastos o inversiones y al mismo tiempo, consigue financiación para el mismo.

- Ventas

Planifica todas las estrategias para el comercio de los productos y servicios que ofrece la empresa. Es el responsable de hacer efectivas las ventas, mejorando cada vez su proceso en beneficio de los clientes, los mismos que después de ser adquiridos deben ser retenidos gracias al esfuerzo de la organización y sobre todo de este departamento.

- Compras

Es aquí donde se manejan los inventarios del negocio. Su responsabilidad primordial recae en la adquisición de todos los insumos que la empresa requiere para operar con normalidad. Además, se encarga de buscar los mejores proveedores y de gestionar los inventarios para que estos estén siempre disponibles y organizados.

- Tecnología

Este departamento es el encargado de administrar toda la infraestructura tecnológica a nivel de *software* y de *hardware* de la empresa. Es aquí donde se desarrolla y se da mantenimiento a los sistemas de información.

- Servicio Técnico

Es el responsable del mantenimiento y arreglo de los distintos equipos tecnológicos. Además, se encarga de brindar soporte a los clientes en caso de requerirlo.

La figura a continuación muestra la estructura jerárquica de la empresa Compunex con todos los departamentos mencionados anteriormente y los diferentes niveles de subordinación existentes. Del mismo modo, se visualizan aquellas líneas de autoridad y poder de decisión que poseen los distintos trabajadores según su cargo.

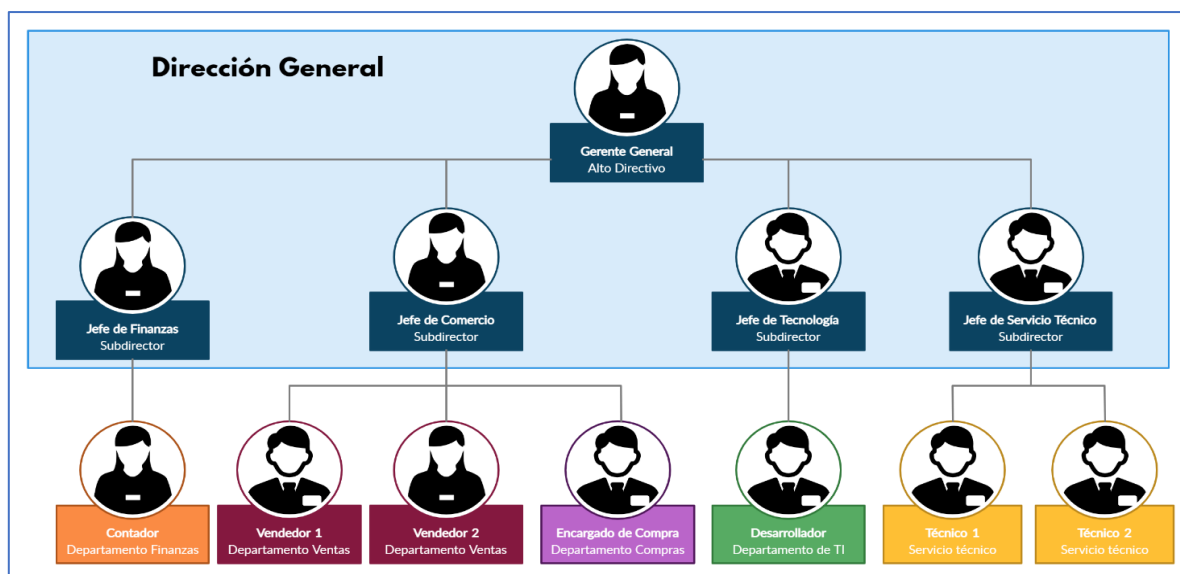


Figura 3-02. Organigrama de la empresa Compunex. (Cisneros & Vaca, 2022)

2.4 Esquema de ventas

Se puede considerar como tradicional al esquema de ventas que realiza la empresa Compunex, ya que este se efectúa de manera presencial e involucra un contacto directo con sus clientes en la tienda física que poseen; sin embargo, este se limita a la zona donde se encuentra ubicada.

Este proceso de ventas comienza con el contacto inicial entre el vendedor y el cliente, el cual se establece una vez que el departamento de ventas se haya encargado de la adquisición de los posibles clientes gracias a la aplicación de diversas estrategias comerciales. Es en este contacto, donde el cliente presenta todos sus requerimientos o necesidades actuales; mientras que el negocio, por su lado, realiza un exhaustivo análisis de la información receptada para poder ofertar aquel producto o servicio que resulte ser la mejor solución o se adapte a lo que el cliente solicita y para posteriormente presentarle dichas propuestas.

Seguidamente, se realiza un intercambio de ideas entre el vendedor y el cliente, con las negociaciones pertinentes para que este quede satisfecho con su futura compra y, de ser necesario, efectuar las debidas modificaciones hasta que se logre conseguir lo que el cliente busca. Concluida esta aclaración y negociación, se procede con el pago y cierre de la venta,

esperando que el beneficio de esta sea mutuo. Además, se añade dentro de este proceso un seguimiento de la satisfacción que se lleva el cliente tanto del producto como el proceso de venta.

A continuación, se presenta un resumen en forma gráfica del proceso de ventas descrito anteriormente.

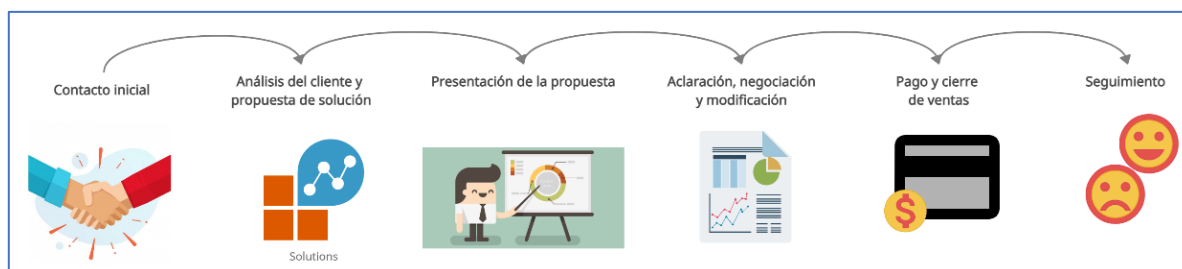


Figura 4-02. Proceso de venta de la empresa Compunex. (Cisneros & Vaca, 2022)

Compunex maneja dos líneas diferentes de mercado a las que van enfocados sus productos. Principalmente, brinda atención a clientes particulares quienes generalmente buscan soluciones para uso doméstico. Los productos más involucrados en esta línea son impresoras de consumo, clones, televisores, laptops, accesorios y periféricos de computación. Cabe recalcar que, para este tipo de venta se cuenta con descuentos para el cliente, en especial si el pago se lo realiza en efectivo.

De igual manera, cuenta con atención al sector empresarial donde se satisface necesidades tecnológicas a la vanguardia y requerimientos específicos de cada organización. Entre los productos más requeridos por parte de este tipo de clientes son servidores, tarjetas gráficas profesionales, clones totalmente compatibles y licenciados, equipamiento para redes y cableado estructurado. Para este tipo de venta, también se presentan varios descuentos según la cantidad de la compra y el método de pago.

Además, un gran sector de clientes de la empresa proviene de distintas provincias del país, especialmente de Imbabura, Carchi, Cotopaxi y Sucumbíos. Este grupo de clientes representa aproximadamente el 37% de las ventas mensuales.

2.5 Proyección a comercio electrónico

Al ser esta una empresa enfocada en equipos informáticos cuya demanda ha crecido significativamente en los últimos años, requiere de nuevas alternativas para la comercialización de sus productos distintas a la tradicional venta presencial que se realiza al momento. Además, dadas las presentes circunstancias originadas por el masivo contagio de SARS-COV2, muchos de sus actuales clientes no pueden acceder a los productos y servicios que brinda la empresa debido a ciertas restricciones que se mantienen vigentes, limitando su capacidad de abarcar nuevos mercados y por ende expandirse como negocio.

De igual manera, es importante mencionar que para la ejecución de los procesos que realiza Compunex no se dispone de ningún software especializado, ya que todo está gestionado mediante herramientas ofimáticas como Excel y Word, lo que muchas veces ocasiona que servicio no sea tan eficiente como se espera.

Por esta razón se necesita un nuevo método y proceso de venta donde el cliente pueda tener acceso a los productos y servicios del negocio de manera instantánea y donde pueda realizar sus compras de una manera cómoda y segura. Por lo que, la incorporación de un e-commerce representa la mejor solución para llevar una mejor gestión interna y externa de la empresa.

3. CAPÍTULO 3: Análisis y diseño

El siguiente capítulo corresponde al levantamiento de todos los requerimientos que la empresa solicita para el desarrollo del sistema. A través de un profundo análisis, se busca identificar los procesos principales de la organización, los cuales serán la base fundamental para la realización de los distintos módulos de los que se compondrá el sistema de información. Además, se incluyen dos de los principales diagramas para el diseño de software, siendo estos los diagramas de base de datos y de casos de uso; así como también el diseño de las interfaces de usuario tanto del sistema como del *e-commerce*.

3.1 Requerimientos Funcionales del Sistema de Información

El sistema contará con 4 módulos principales que abarcan todos los procesos de la empresa, siendo estos: compras, ventas, inventarios y garantías. El ingreso al mismo será controlado a través de un inicio de sesión con usuario y contraseña. Además, se permitirá la creación de varios usuarios a los que se les asignará un único rol. Dicho rol poseerá todas las autorizaciones a través de las cuales se determinará el acceso a los diferentes módulos mencionados anteriormente con sus respectivos permisos de escritura, lectura, edición y eliminación; teniendo en cuenta que, para mayor control y consistencia de datos solamente se realizará un eliminado lógico. Estos roles podrán ser creados según la necesidad de la empresa de otorgar ciertos permisos a sus empleados en base a la función que desempeñen, sin embargo, siempre existirá por defecto el rol “administrador” que poseerá los permisos totales del sistema y el rol “cliente” para todo usuario que se registre en la tienda en línea.

A continuación se muestran los requerimientos funcionales del sistema, los mismos que serán clasificados en los módulos a los que pertenecen.

3.1.1 Módulo Compras

Los requerimientos de este módulo giran en torno al manejo de la información tanto de las compras realizadas por parte de la empresa, así como también de los proveedores de quienes se obtienen dichas compras. Estos requerimientos se muestran en la siguiente tabla.

ID	Requerimiento
RF - 01	Verificación de ingreso al módulo
RF – 02	Ingresar un proveedor
RF – 03	Consultar un proveedor
RF – 04	Modificar un proveedor
RF – 05	Eliminar un proveedor
RF – 06	Ingresar una orden de compra
RF – 07	Consultar una orden de compra
RF – 08	Modificar una orden de compra
RF – 09	Eliminar una orden de compra

Tabla 2-03. *Requerimientos funcionales módulo compras. (Cisneros & Vaca, 2022)*

3.1.2 Módulo Ventas

La siguiente tabla recopila todos los requerimientos del módulo ventas, los cuales se desarrollan en base al manejo de toda la información de las ventas realizadas por parte de la empresa, así como también de los clientes a quienes se efectuaron dichas ventas.

ID	Requerimiento
RF – 10	Verificación de ingreso al módulo
RF – 11	Ingresar un cliente
RF – 12	Consultar un cliente
RF – 13	Modificar un cliente
RF – 14	Eliminar un cliente

RF – 15	Ingresar una orden de venta
RF – 16	Consultar una orden de venta
RF – 17	Modificar una orden de venta
RF – 18	Eliminar una orden de venta

Tabla 3-03. *Requerimientos funcionales módulo ventas. (Cisneros & Vaca, 2022)*

3.1.3 Módulo Inventario

Los requerimientos que presenta este módulo están relacionados con la administración de todos los ítems del inventario de la empresa junto con sus productos relacionados, los mismos que son detallados en la tabla a continuación.

ID	Requerimiento
RF – 19	Verificación de ingreso al módulo
RF – 20	Ingresar un ítem
RF – 21	Consultar un ítem
RF – 22	Modificar un ítem
RF – 23	Eliminar un ítem
RF – 24	Ingresar un producto
RF – 25	Consultar un producto
RF – 26	Modificar un producto
RF – 27	Eliminar un producto

Tabla 4-03. *Requerimientos funcionales módulo inventarios. (Cisneros & Vaca, 2022)*

3.1.4 Módulo Garantías

Los requerimientos de este módulo están hechos en base al estado de garantía en que puede encontrarse un ítem de la empresa. Dichos requerimientos se muestran en la siguiente tabla.

ID	Requerimiento
RF – 28	Verificación de ingreso al módulo
RF – 29	Ingresar una garantía
RF – 30	Consultar una garantía
RF – 31	Eliminar una garantía

Tabla 5-03. *Requerimientos funcionales módulo garantías. (Cisneros & Vaca, 2022)*

3.2 Requerimientos Funcionales del E-Commerce

El sistema de comercio electrónico contempla todos los requerimientos funcionales mostrados en la siguiente tabla.

ID	Requerimiento
RF – 01	Registro de clientes
RF – 02	Mostrar un catálogo de los productos disponibles de la empresa
RF – 03	Búsqueda de productos
RF – 04	Añadir los productos a un carrito de compras
RF – 05	Generar un comprobante al concluir el proceso de venta

Tabla 6-03. *Requerimientos funcionales E-commerce. (Cisneros & Vaca, 2022)*

3.3 Requerimientos No Funcionales

A continuación, se presentan los requerimientos no funcionales tanto del sistema de información como del e-commerce.

ID	Requerimiento
RNF – 01	El sistema deberá ser desarrollado bajo la arquitectura de capas <i>frontend</i> , <i>backend</i> y base de datos

RNF – 02	El sistema de base de datos relacionales a utilizar será PostgreSQL
RNF – 03	La capa de <i>backend</i> deberá ser implementada bajo el <i>framework</i> FastApi
RNF – 04	La capa de <i>frontend</i> deberá ser implementada bajo el <i>framework</i> Nuxt.js
RNF – 05	Los usuarios podrán acceder al sistema y a la tienda en línea desde cualquier navegador

Tabla 7-03. *Requerimientos no funcionales.* (Cisneros & Vaca, 2022)

3.4 Requerimientos propios de todo Sistema de Información

En la tabla a continuación se detallan todos los requerimientos propios de un sistema asociados a la administración de este, los cuales giran en torno al manejo de todos los usuarios registrados, los distintos roles que existen con sus respectivos permisos, y también, de los módulos y submódulos de los que se compone el sistema.

ID	Requerimiento
RF – 33	Ingresar un usuario
RF – 34	Consultar un usuario
RF – 35	Modificar un usuario
RF – 36	Eliminar un usuario
RF – 37	Ingresar un rol
RF – 38	Consultar un rol
RF – 39	Modificar un rol
RF – 40	Eliminar un rol
RF – 41	Consultar permisos asociados a un rol
RF – 42	Ingresar un permiso
RF – 43	Consultar un permiso
RF – 44	Modificar un permiso
RF – 45	Eliminar un permiso

RF – 46	Ingresar un módulo
RF – 47	Consultar un módulo
RF – 48	Modificar un módulo
RF – 49	Eliminar un módulo
RF – 50	Ingresar un submódulo
RF – 51	Consultar un submódulo
RF – 52	Modificar un submódulo
RF – 53	Eliminar un submódulo

Tabla 8-03. *Requerimientos del sistema. (Cisneros & Vaca, 2022)*

3.5 Diseño de la base de datos

La base de datos del sistema fue diseñada tomando en cuenta el modelo entidad relación y las tres formas de normalización necesarias para disminuir la redundancia de datos y simplificar la dependencia de columnas. Más adelante se presentan todas las tablas del modelo con sus respectivos atributos clasificadas en los módulos a los que pertenecen de acuerdo con el diseño del sistema detallado anteriormente.

3.5.1 Módulo Compras

El presente módulo está formado por las tablas proveedores, compras y detalle de compras. Su funcionamiento en conjunto permite el registro de los proveedores y de las compras con su respectivo formato de cabecera detalle. Las tres tablas del módulo mantienen una relación entre sí, debido a que un proveedor puede facturar varias veces a la empresa y simultáneamente cada factura puede tener distintos elementos asociados a la misma compra.

Tal como se detalla en la siguiente figura, para la identificación de un proveedor se requiere de toda la información necesaria para su contacto como lo es su nombre, teléfono, correo

electrónico y dirección. De la misma manera, para el registro de la cabecera de compra son fundamentales el número de la factura, la fecha de su realización y los valores del subtotal, descuento aplicado en caso de haberlo y total de la compra. Como ya se mencionó, dicha cabecera posee su propio detalle en el que se encuentran los datos de compra de un producto en específico, siendo estos su precio y cantidad.

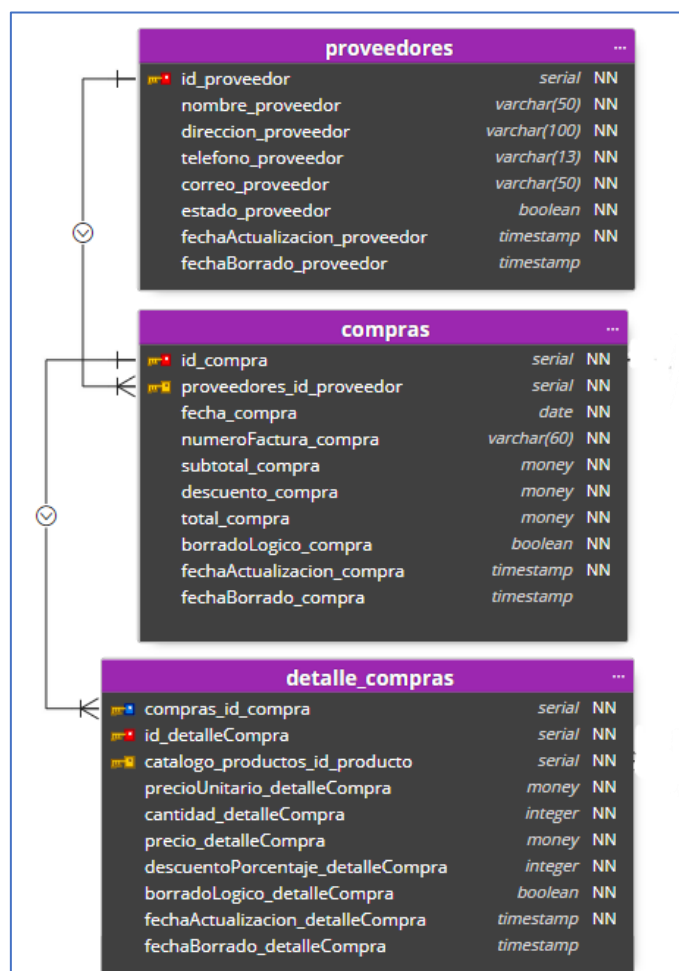


Figura 5-03. Diseño del módulo de compras. (Cisneros & Vaca, 2022)

3.5.2 Módulo Ventas

Para el módulo de ventas se presentan las tablas clientes, ventas y detalle de ventas. Las tres tablas en conjunto permiten el registro tanto de clientes como de las compras que estos realicen con su debido detalle. Es en este módulo en el que se describen todas las posibles ventas que la empresa efectúa, pudiendo estas realizarse a un mismo cliente.

Para el correcto registro de un nuevo cliente se requiere de cierta información personal que incluye datos como nombre, número de cédula, dirección, teléfono y correo electrónico. Así mismo, en la cabecera de cada venta se reúnen aspectos fundamentales para el registro de esta, incluyendo valores de subtotales, descuentos y totales, fecha de su realización y número de venta. Por otro lado, el detalle de ventas contiene toda la información de la lista de productos que han sido vendidos en una sola venta, tal como se refleja en la figura a continuación.

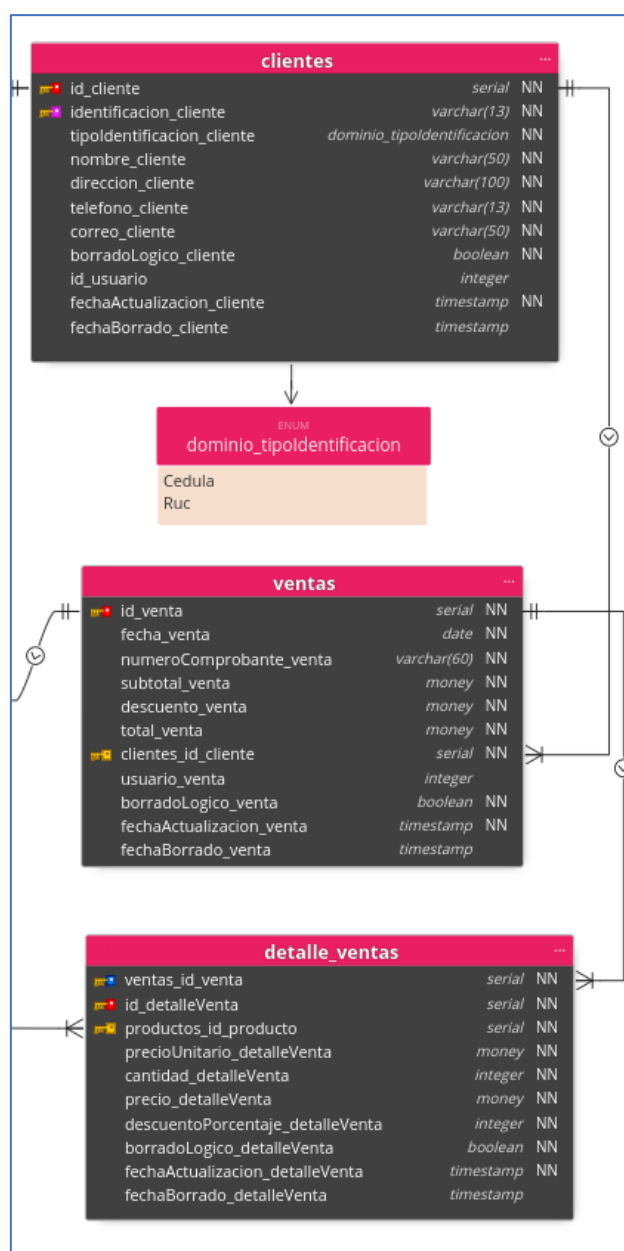


Figura 6-03. Diseño del módulo ventas. (Cisneros & Vaca, 2022)

3.5.3 Módulo Inventarios

Este módulo permite llevar un registro de todos los productos existentes en el inventario de la empresa de manera unitaria con toda su información detallada. Para la administración de los productos es fundamental almacenar ciertos datos relacionados con la descripción de este como lo son la marca, unidad de medida y sobre todo la cantidad existente del mismo. Por otro lado, dentro del inventario conviene registrar toda la información relacionada con la venta de dicho producto, es decir, sus precios de venta o descuentos. Además, es aquí en donde se verifica si algún producto sigue disponible dentro del inventario o si ya fue vendido, gracias al atributo estado que se refleja en la figura más adelante.

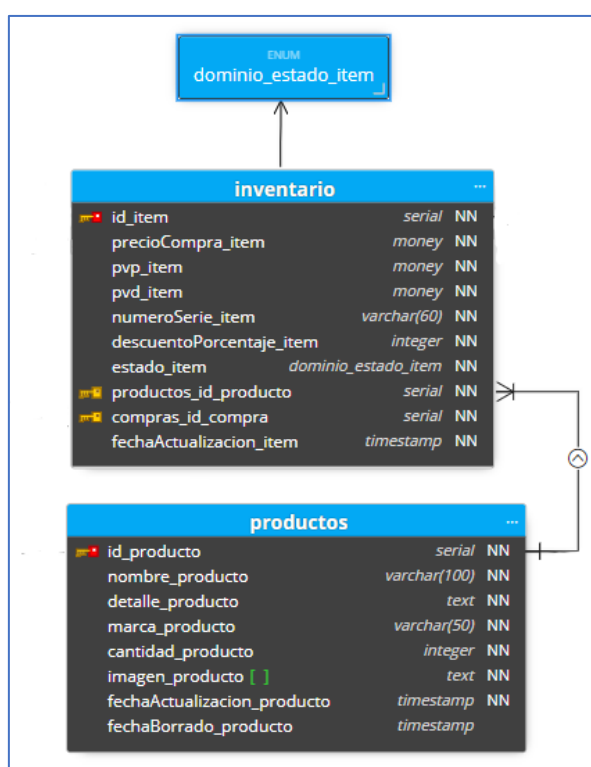


Figura 7-03. Diseño del módulo inventarios. (Cisneros & Vaca, 2022)

3.5.4 Módulo Garantías

El módulo incluye una única tabla que permite registrar todas las garantías que se ofrecen a todos los productos dentro de la empresa. Para esto, se requiere almacenar cierta información

importante como la fecha de ingreso del producto a garantía y el detalle del motivo de su entrada, tal como se muestra a continuación.



garantias			
id_garantia	serial	NN	
inventario_id_item	serial	NN	
clientes_id_cliente	serial	NN	
fechaEntrada_garantia	timestamp	NN	
detalle_garantia	text	NN	
borradoLogico_garantia	boolean	NN	
fechaActualizacion_garantia	timestamp	NN	
fechaBorrado_garantia	timestamp		

Figura 8-03. Diseño módulo garantías. (Cisneros & Vaca, 2022)

3.5.5 Gestión de permisos

Como se observa en la siguiente figura, para la administración del sistema se trabaja con seis tablas principales que corresponden a usuarios, roles, autorizaciones, permisos, módulos y submódulos. Las tablas en conjunto posibilitan el manejo de todos los usuarios del sistema a los que se les fue asignado un rol en particular. Dicho rol será reconocido por su nombre y descripción que detalla la función que desempeña un empleado en la empresa. Además, cada rol posee ciertos permisos que se encuentran asociados a una autorización, la misma que indica la posibilidad de leer, crear, editar o eliminar un registro dentro de un submódulo específico. Y al mismo tiempo este submódulo es parte un módulo principal.

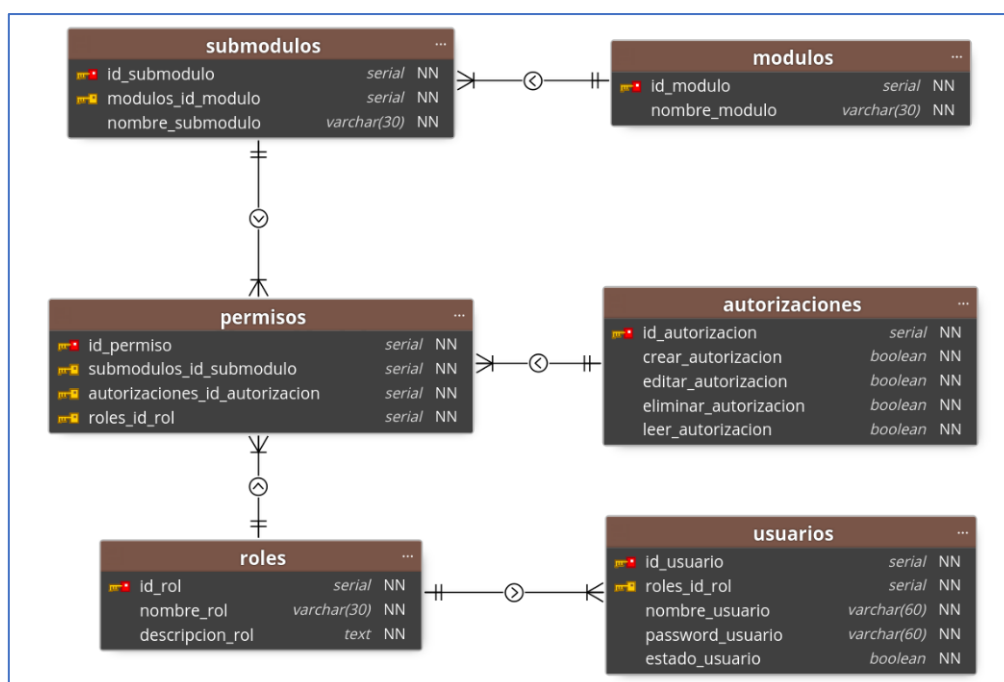


Figura 9-03. Diseño de gestión del sistema. (Cisneros & Vaca, 2022)

El modelo entidad relación completo de todo el sistema se adjunta en los anexos del proyecto.

3.6 Diseño de casos de uso

Para poder visualizar las posibles interacciones que podría tener un usuario con el sistema se diseñó un diagrama de caso de uso general. Este traduce de manera gráfica todos aquellos requerimientos funcionales principales los mismos que están divididos en paquetes que simbolizan los módulos de compras, inventarios, ventas y garantías.

Como se mencionó anteriormente, la interacción de cualquier usuario viene establecida por el rol que posee y ya que existen dos roles principales por defecto en el sistema, administrador y cliente, se los utilizará para ejemplificar los diferentes casos de usos tal como se observarán en las figuras más adelante.

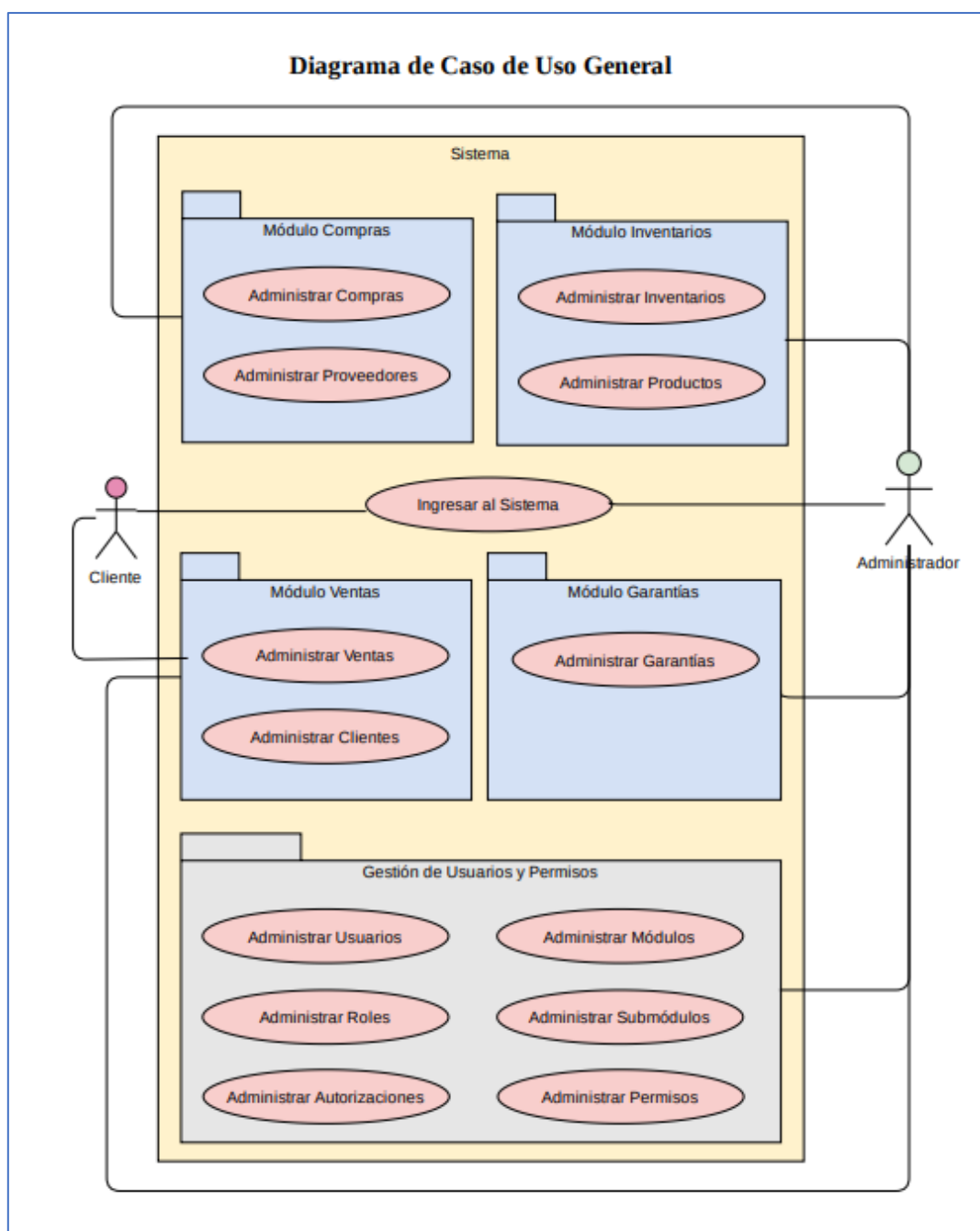


Figura 10-03. Diagrama de caso de uso general. (Cisneros & Vaca, 2022)

Este diagrama muestra una visión general de cómo un mismo actor, en este caso con el rol administrador, tiene permitido el ingreso al sistema para posteriormente acceder a todos los módulos de los que se compone. Por otro lado, el actor con el rol cliente puede ingresar al sistema solamente para acceder al módulo de ventas. Sin embargo, no se detalla qué acciones pueden realizar dentro de cada uno de estos módulos, por lo que, también se realizaron los

casos de uso al siguiente nivel. Estos permiten visualizar de mejor manera las relaciones del actor con los requisitos funcionales de cada submódulo. Además, se observa cómo a partir de un caso de uso principal se derivan otros cuatro casos correspondientes a las acciones de insertar, modificar, consultar y eliminar.

El módulo ventas, que presenta los submódulos clientes y ventas, es el único al que acceden los dos roles por defecto. Para el caso específico de ventas, el rol administrador tiene permisos para gestionar completamente la información del submódulo, mientras que, el actor con rol de cliente solo puede realizar consultas, es más, solo la consulta específica de sus órdenes de compras en la empresa tal como se muestra en la siguiente figura.

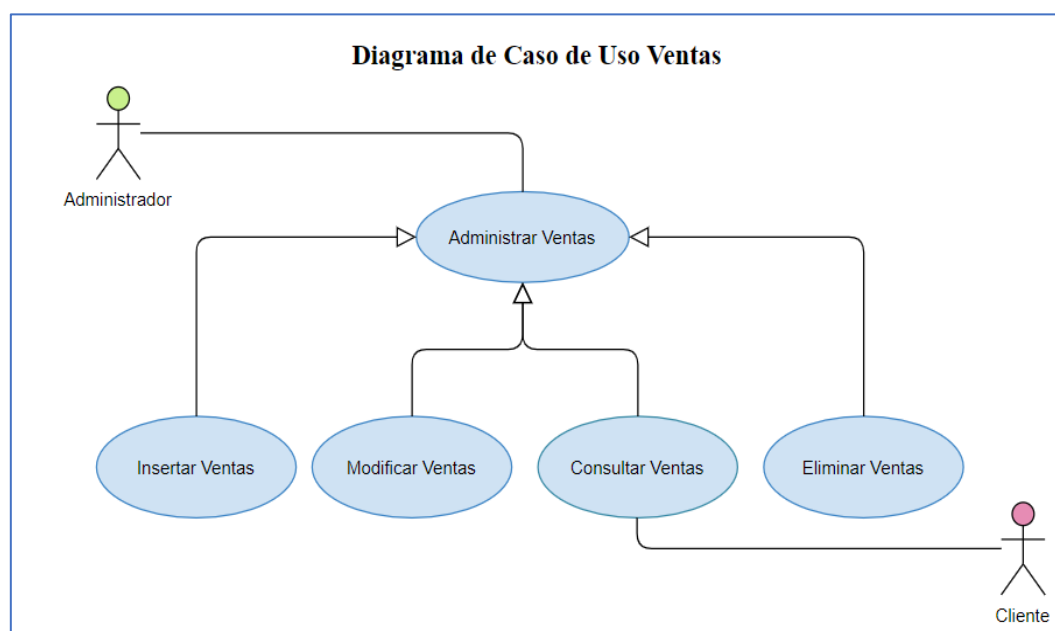


Figura 11-03. Caso de uso ventas. (Cisneros & Vaca, 2022)

De igual manera para el submódulo de clientes, el actor administrador tiene permiso para las cuatro acciones de gestión de datos. Sin embargo, para el rol cliente solo está permitida la operación de consulta de clientes. Es decir, un cliente podrá únicamente ver sus datos pero no modificarlos.

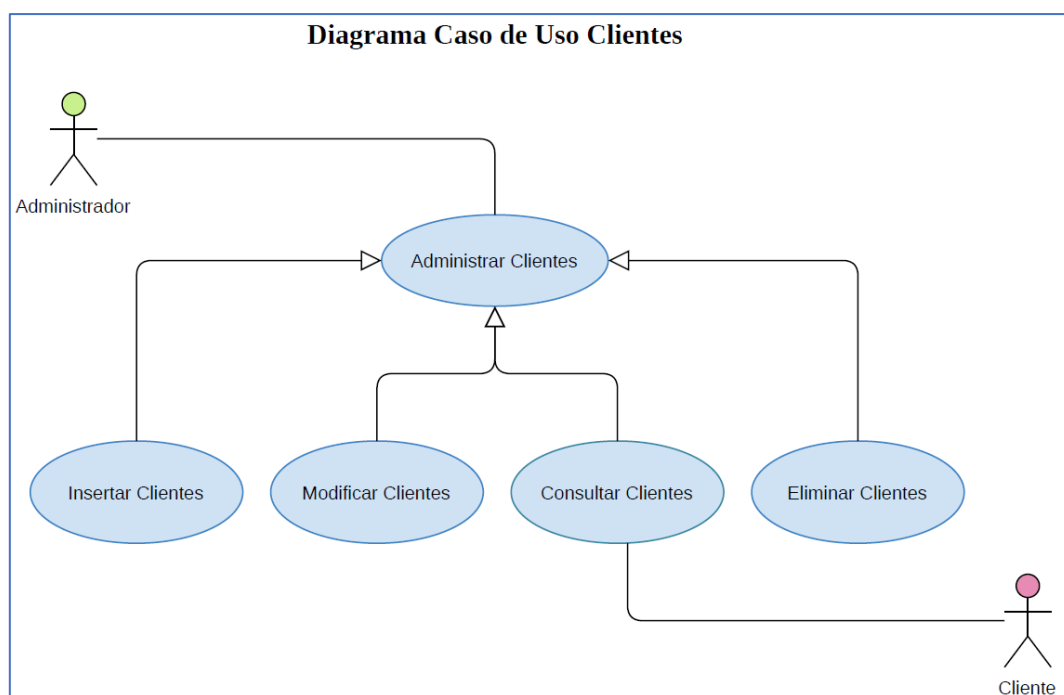


Figura 12-03. Caso de uso clientes. (Cisneros & Vaca, 2022)

La figura a continuación muestra el caso de uso al siguiente nivel del submódulo proveedores, perteneciente al módulo principal de compras. En este, el actor posee permisos para realizar una completa administración de los proveedores, es decir, puede consultar datos, insertar nuevos registros, modificar los existentes y eliminarlos si es necesario.

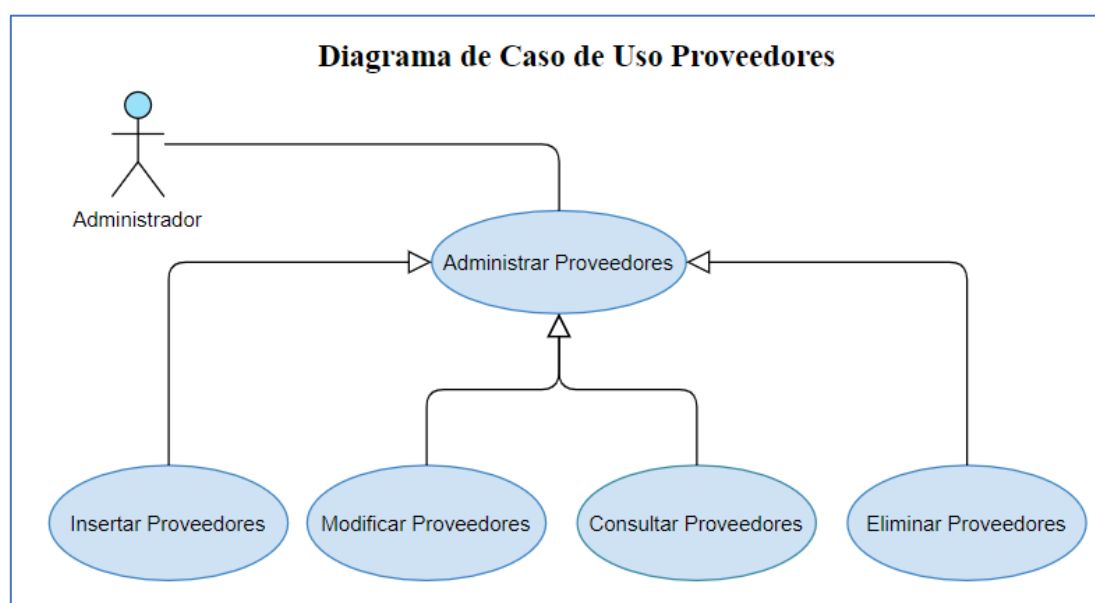


Figura 13-03. Caso de uso submódulo proveedores. (Cisneros & Vaca, 2022)

Para el resto de los submódulos, este actor con rol de administrador presenta el mismo comportamiento debido a que dispone de los permisos totales para el manejo del sistema.

Los diagramas de caso de uso al siguiente nivel completos se encuentran en los anexos del proyecto.

3.7 Diseño de interfaz

El diseño de todas las interfaces gráficas tanto del sistema como del e-commerce fue elaborado con la aplicación colaborativa **Figma**, la misma que permite realizar prototipado web de forma rápida y sencilla. Esta ofrece distintas funciones y herramientas que posibilitan la creación desde simples bocetos hasta un prototipo de alta fidelidad. Una de sus grandes ventajas, es que está basada en la web, es decir, que toda su información se guarda en la nube pudiéndose acceder a esta desde cualquier dispositivo en el que se tenga conexión a internet y a su vez, se puede colaborar con otros en tiempo real.

En primer lugar, para este proyecto se comenzó con el diseño de lo que se conoce como **wireframes**, que son una representación gráfica del esquema de una página. Estos permiten tener una idea general de cómo será y los elementos que contendrá cada pantalla, sin entrar en cuestiones de color o tipografías. A partir de esta etapa, se fueron agregando más estilos y detalles en cada componente de la interfaz hasta conseguir lo que sería el diseño final de los dos sistemas, los mismos que fueron aprobados por la empresa.

A continuación, se muestran los diseños de las pantallas principales de ambos sistemas, mientras que los **wireframes** completos se adjuntan en los anexos del proyecto.

3.7.1 Sistema de Información

El sistema al estar dividido en módulos en los que se realizan las mismas operaciones simplifica el diseño de nuevas y distintas pantallas. De manera que, se presentan cinco pantallas generales de la cuales tres son reutilizadas en cada submódulo para las operaciones de leer, agregar, editar y eliminar un registro. Por otro lado, las dos pantallas restantes corresponden a las interfaces de inicio de sesión y página principal, las mismas que serán explicadas con más detalle a continuación.

Interfaz Inicio de Sesión

Como se mencionó al comienzo del capítulo, el ingreso al sistema está controlado por la siguiente pantalla de inicio de sesión en la que un determinado usuario debe colocar sus credenciales de acceso, siendo estas su nombre de usuario y contraseña correspondiente. Si dichas credenciales son válidas, continuará a la siguiente pantalla principal.

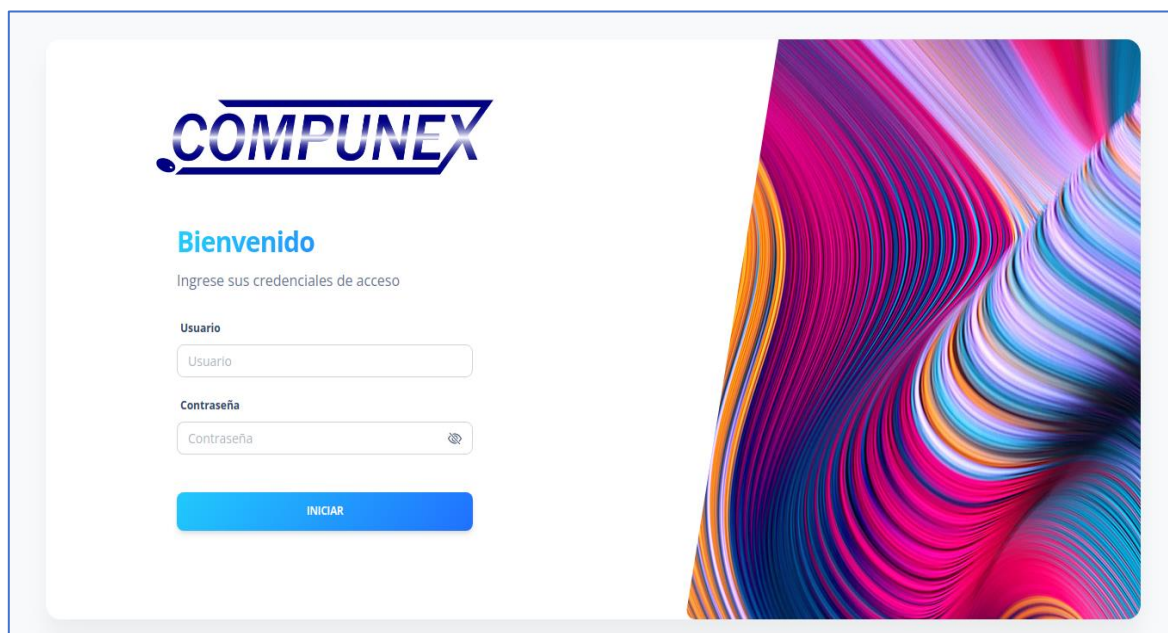


Figura 14-03. Interfaz Inicio Sesión. (Cisneros & Vaca, 2022)

Interfaz Página Principal y Dashboard

La siguiente interfaz corresponde a la página principal del sistema con 3 secciones importantes. En la parte central se encuentra el **dashboard**, el mismo que presenta en forma de resumen el dinero mensual proveniente de las transacciones principales de la empresa junto con una tabla que muestra las últimas ventas guardadas tanto del sistema como del **e-commerce**. Por otro lado, en la parte superior de la página se encuentra la barra de navegación que indica el módulo y submódulo en el que se encuentra, el nombre del usuario y un botón para cerrar sesión. Seguido, en la parte lateral izquierda está el menú del sistema que cuenta con una lista de todos los módulos y submódulos disponibles. Cabe mencionar que esta lista solo mostrará aquellos módulos a los que tenga acceso el rol que le fue asignado al usuario ingresado. Estas dos secciones mencionadas anteriormente, aparecerán en todas las páginas del sistema a excepción del inicio de sesión.

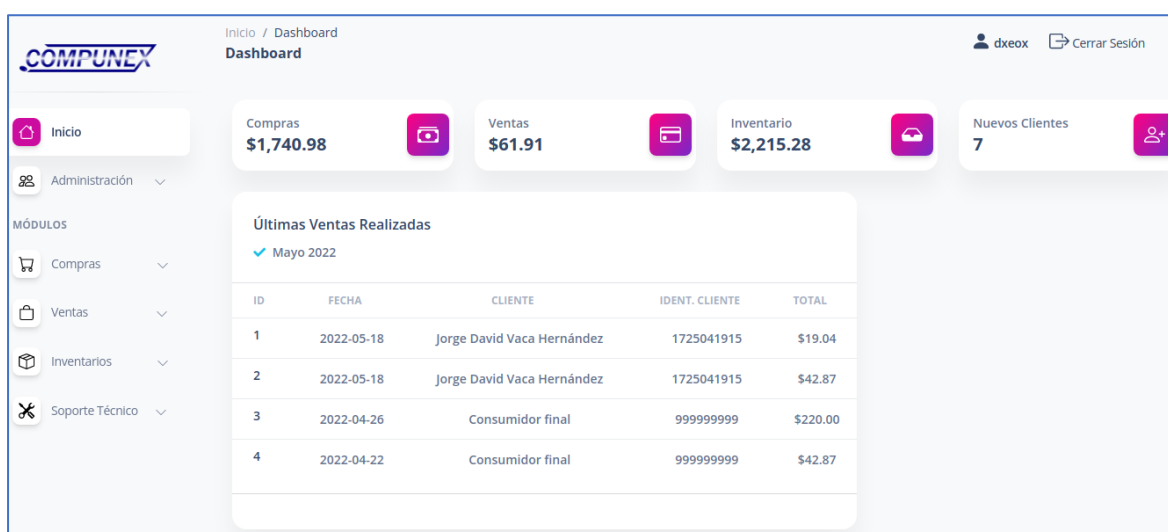


Figura 15-03. Interfaz página principal con Dashboard. (Cisneros & Vaca, 2022)

Interfaz Visualización de Registros

Al acceder a cualquiera de los submódulos, se podrá observar una tabla cargada con la información de todos los registros almacenados en el sistema pertenecientes al submódulo

seleccionado. En la parte superior se presenta el título del submódulo, seguido con un botón para seleccionar el número de registros que se desea visualizar por página y también otro botón que brinda la opción de agregar un nuevo registro, ya sea mediante un modal o en otra página.

La parte central, es decir la tabla de datos, contiene la información de los registros presentada en filas y columnas según la cantidad existente de estos con sus respectivos atributos. Además, la última columna posee dos íconos que simbolizan las acciones de editar y eliminar un registro, por lo que a través de estas se puede realizar dichas operaciones en la misma o en otra página.

Finalmente, en la parte inferior derecha se encuentran los botones para la paginación de la tabla, que permitan navegar entre las diferentes páginas según el número de registros seleccionados.

La siguiente figura muestra un ejemplo de la interfaz mencionada aplicada al submódulo de usuarios.

ID	NOMBRE	MÓDULO AL QUE PERTENECE		
1	Cientes	Ventas		
2	Proveedores	Compras		
3	Ventas	Ventas		
4	Autorizaciones	Administración		
5	Usuarios	Administración		
6	Roles	Administración		
7	Módulos	Administración		
8	Submódulos	Administración		
9	Permisos	Administración		
10	Productos	Inventario		

Figura 16-03. Interfaz visualización de registros. (Cisneros & Vaca, 2022)

Interfaz Crear/ Editar un registro

A continuación, se muestra el diseño de la interfaz para crear un nuevo registro. Esta contiene a manera de un formulario todos los campos correspondientes a los atributos necesarios para agregar nuevos registros según el modelo de la base de datos. Dichos atributos varían según el submódulo en el que se encuentre el usuario, pudiendo ser menos o más campos. Además, presenta un título descriptivo de la acción a realizar y dos botones que permiten guardar los cambios o cancelar la acción y regresa a la página anterior.



La imagen muestra una interfaz de usuario para añadir un nuevo rol. El formulario tiene un título "Añadir Nuevo Rol" en la parte superior izquierda y un botón de cerrar "x" en la parte superior derecha. Hay dos campos de entrada de texto: "Nombre" y "Descripción". En la parte inferior derecha del formulario, hay dos botones: "CANCELAR" (gris) y "AGREGAR" (rojo).

Figura 17-03. Interfaz para añadir un registro. (Cisneros & Vaca, 2022)

La interfaz para editar un registro es la misma que la de agregar, con la única diferencia de que esta presenta los datos del registro cargados en el formulario de manera que sean más fáciles para actualizarlos. Este cambio en el diseño se observa en la siguiente figura.



The image shows a modal window titled "Editar Rol" with a close button (X) in the top right corner. It contains two text input fields: "Nombre" with the value "Comprador" and "Descripción" with the value "Puede registrar compras". At the bottom, there are two buttons: "CANCELAR" (grey) and "ACTUALIZAR" (pink).

Figura 18-03. Interfaz para editar un registro. (Cisneros & Vaca, 2022)

Interfaz Eliminar un registro

Como se mencionó, desde la tabla de visualización se puede acceder al ícono para eliminar un registro. Al ser una acción con cambios irrevertibles, se añadió una ventana modal de confirmación con dos botones, uno para continuar con la eliminación y otro para cancelar. De esta manera, se evita que se realicen eliminaciones accidentales y la pérdida del dato.

A continuación se muestra el diseño de la ventana emergente para confirmar la acción.



The image shows a modal window titled "Confirmar" with a close button (X) in the top right corner. It contains a text prompt: "¿Está seguro que desea eliminar este registro?". At the bottom, there are two buttons: "NO" (grey) and "SI" (red).

Figura 19-03. Interfaz modal confirmación de eliminación. (Cisneros & Vaca, 2022)

3.7.2 E-Commerce

Las interfaces del comercio electrónico fueron realizadas tomando en cuenta los componentes fundamentales que debe poseer una tienda en línea como lo son un catálogo de productos y un

carrito de compras. De igual manera, estas presentan un diseño moderno, intuitivo y fácil de navegar para cualquier tipo de usuario. En conjunto proporcionan una experiencia de compra en línea rápida y agradable.

Inicio y registro de clientes

La siguiente interfaz corresponde al registro de un nuevo cliente, en el que se solicita al usuario completar un pequeño formulario con los datos correspondientes a su correo electrónico, nombre de usuario y contraseña para crear una nueva cuenta. En caso de ser un usuario ya registrado, únicamente aparecerán los campos de usuario y contraseña, los cuales se encargan de verificar el inicio de sesión generando un token de acceso válido.

Cabe mencionar que, estas dos interfaces solo aparecerán en caso de que un usuario desea realizar una compra ya que para el resto de la navegación en la tienda en línea se lo puedo hacer como invitado es decir sin identificarse.

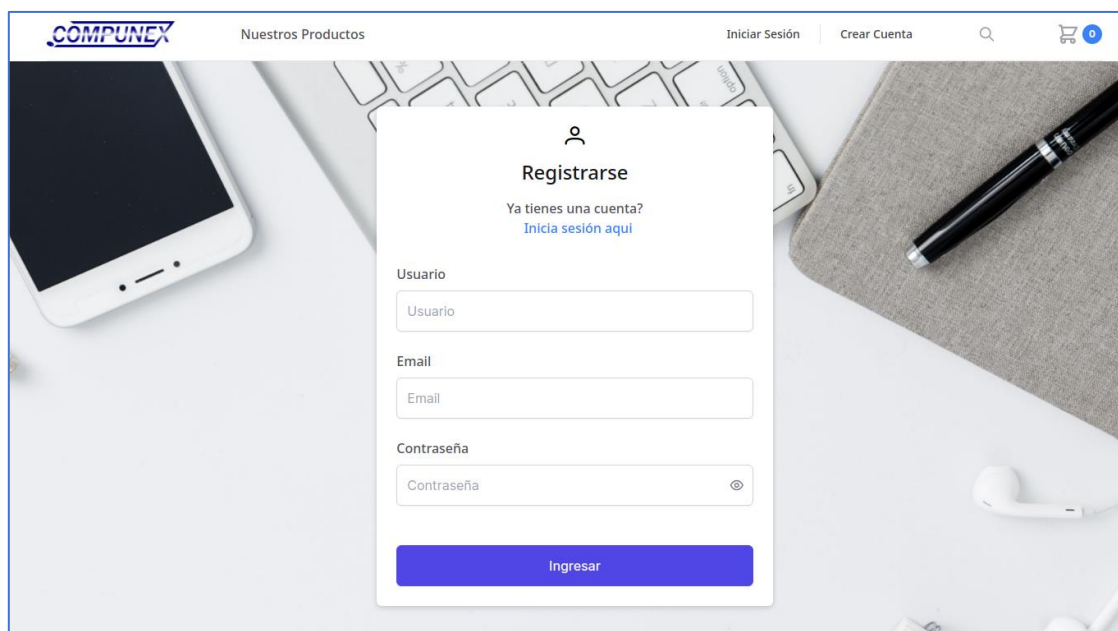


Figura 20-03. Interfaz registro de usuarios E-commerce. (Cisneros & Vaca, 2022)

Catálogo de productos

Para desplegar todos los productos de forma que sean atractivos para los clientes, se realizó el diseño de varias tarjetas contenedoras. Estas se encargan de mostrar cierta información de un producto en específico como su nombre, precio y una imagen que lo ilustre. Además, contiene un botón que permite agregar dicho producto al carrito de compras. Por lo tanto, el conjunto de varias de estas tarjetas posibilita un catálogo de los productos ofertados por la empresa y que están disponibles para una compra. La figura a continuación muestra el diseño estas tarjetas.

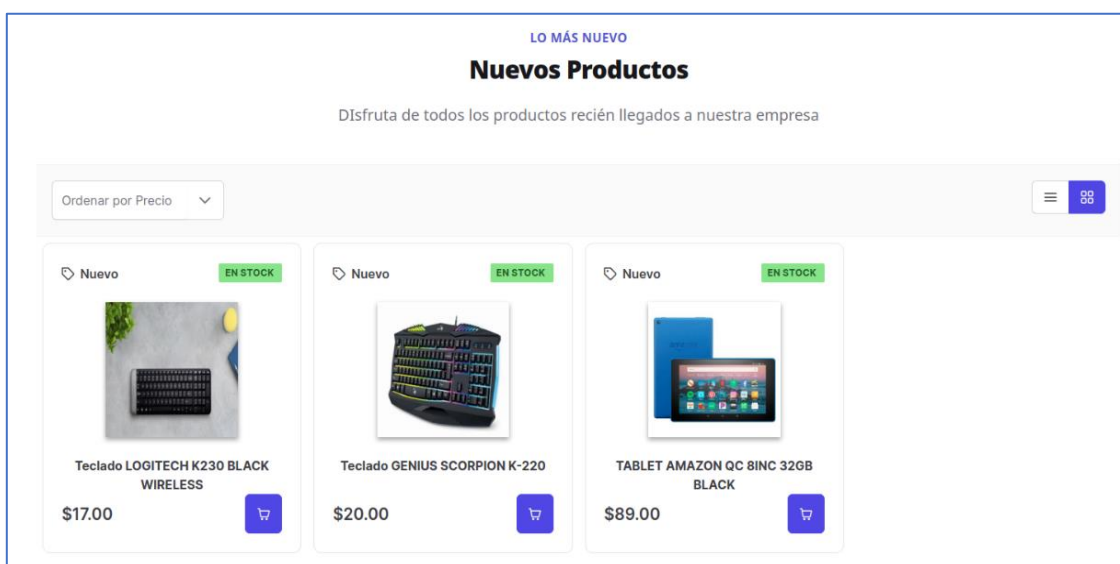


Figura 21-03. *Interfaz catálogo en forma de grid. (Cisneros & Vaca, 2022)*

De la misma manera, se agregó otro distinto diseño de estas mismas tarjetas para que se desplieguen en forma de lista los mismos componentes dándole al usuario otra opción de visualización de los productos. Tanto esta vista como la anterior en forma de tabla, poseen un botón para ordenar los productos de mayor a menor precio y viceversa. A continuación, se muestra la opción de la lista de productos.



Figura 22-03. Interfaz catálogo en forma de lista. (Cisneros & Vaca, 2022)

Carrito de compras

Una parte fundamental de la tienda en línea es el carrito de compras, por esta razón se procuró diseñarlo de forma simple, intuitivo pero sobre todo útil. Además, para que el usuario pueda acceder a este en cualquier momento se lo colocó dentro de un panel colapsable, el mismo que contiene a manera de resumen todos los productos añadidos con el valor total de la compra hasta el momento. De igual manera, posee dos botones que permiten, ya sea, a proceder a la confirmación de la compra o a seguir navegando en la tienda para seguir agregando más productos. A continuación se visualiza la interfaz final del carrito de compras.

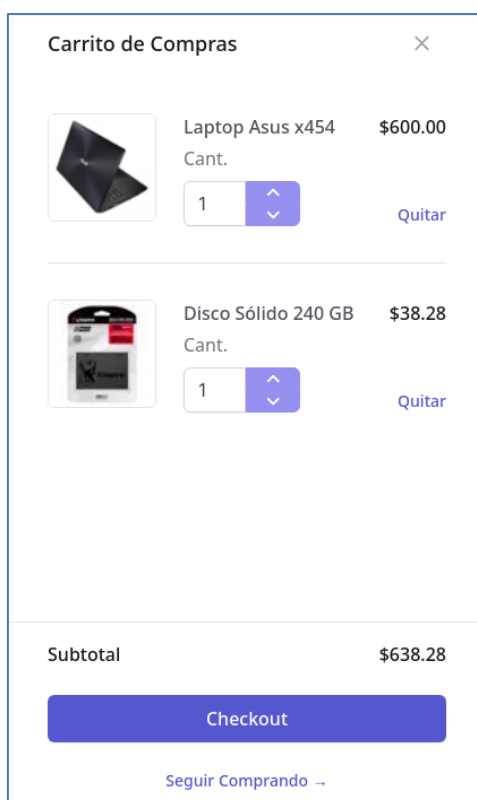


Figura 23-03. Interfaz carrito de compras. (Cisneros & Vaca, 2022)

Orden de compra

Al finalizar todo el proceso de compra, al usuario se le presentará la siguiente interfaz que contiene la orden de su compra para ser confirmada. En esta se encuentran listados todos los productos comprados con sus respectivos precios y cantidades, el total de la compra después de calculados el descuento e IVA correspondiente y finalmente los datos del cliente a quién estará asociada dicha compra. Además se le permite al tal usuario escoger entre obtener su orden de compra como consumidor final o con datos, pudiendo en esta última opción asociarla a un cliente ya registrado o a uno nuevo.







Orden de Compra		Resumen	
<input checked="" type="radio"/> Con Datos	<input type="radio"/> Consumidor Final		Laptop Asus x454 Cant. 1 \$ 600.00
Datos Personales			
Tipo de Identificación	Identificación		
Cédula	1717431942		
Nombre: Daniela Estefania Cisneros Jácome			
Correo: danicisneros2@hotmail.com			
<input type="button" value="Confirmar Compra"/>			
		Subtotal: \$600.00	
		Descuento: -\$0.00	
		IVA %12: \$72.00	
		Total: \$672.00	

Figura 24-03. Interfaz confirmación de compra. (Cisneros & Vaca, 2022)

Historial de compras del usuario

Cada usuario cuenta con la posibilidad de visualizar todas sus compras pasadas en cualquier momento con tan solo ingresar la tienda en línea con sus credenciales de acceso. A continuación, se muestra la siguiente interfaz que corresponde al listado de las ordenes de compras de un usuario.

# Comprobante	Fecha	Nombre Cliente	Total	
8	2022-04-26	Consumidor final	\$220.00	
7	2022-04-22	Consumidor final	\$42.87	
16	2022-06-06	Daniela Estefania Cisneros Jácome	\$19.04	
14	2022-05-25	Daniela Estefania Cisneros Jácome	\$691.04	
13	2022-05-24	Eddy Patricio Cisneros Tamayo	\$22.40	


Mostrando 1 a 5 registros de 5 << < 1 > >> 10

Figura 25-03. Interfaz historial de compras. (Cisneros & Vaca, 2022)

De la misma manera, se le permite al usuario ver detalladamente cada una de estas compras en una nueva ventana. Aquí se encuentran desglosados todos los productos adquiridos con sus respectivos valores, tal como se muestra en la siguiente interfaz.


The screenshot displays a web interface for a purchase order. At the top left, there is a back arrow and the text 'Regresar'. The main heading is 'Orden de Compra No.7' with the date '2022-04-22' below it. The interface is divided into three main sections: 'Detalles de Compra', 'Datos Cliente', and 'Resumen'.

Detalles de Compra

	Disco Sólido 240 GB	Cant.1	\$38.28
---	----------------------------	--------	----------------

Datos Cliente

Identificación
999999999

 Consumidor final

Resumen

Subtotal	\$38.28
Descuento	-\$0.00
IVA (12%)	\$4.59
Total	\$42.87

Figura 26-03. Interfaz detalle orden de compra. (Cisneros & Vaca, 2022)

4. CAPÍTULO 4: Desarrollo y Pruebas

El desarrollo tanto del sistema de información como del *e-commerce* fueron realizados bajo la metodología ágil Scrum. Esta permite proporcionar al cliente entregas tempranas del producto con la finalidad de irlo perfeccionando a través de iteraciones incrementales. De esta manera, se logra involucrar directamente al cliente en el proceso de desarrollo y que sus requerimientos sean satisfechos en todo momento. Para esto, el desarrollo completo del proyecto fue dividido en múltiples *sprints* que contemplan los requerimientos funcionales y no funcionales establecidos en el capítulo anterior.

Especificación de roles

La tabla a continuación muestra los roles del equipo Scrum con sus respectivos responsables asignados para las distintas actividades del desarrollo del proyecto.

Rol	Responsable
Product Owner	David Vaca
Scrum Master	Daniela Cisneros
Development Team	Daniela Cisneros David Vaca
Focus Users	Usuario con rol administrador Usuario con rol cliente Usuario con conocimiento técnico

Tabla 9-04. Roles de Scrum. (Cisneros & Vaca, 2022)

Historias de Usuario

Concluida una reunión con todos los involucrados del proyecto, se lograron identificar y documentar 16 historias de usuario correspondientes al rol de administrador y 7 al del cliente. Estas historias describen de manera coloquial todas las funcionalidades que debe tener el

producto final desde el punto de vista de un usuario en particular. En la siguiente tabla se muestran algunas de las historias de usuario con prioridad más alta dentro del sistema.

#	Enunciado de la Historia			Prioridad
	Rol	Funcionalidad	Razón	
1	Como administrador	Quiero tener un inicio de sesión mediante usuario y contraseña	Para controlar el ingreso al sistema	Alta
2	Como administrador	Quiero poder gestionar el sistema	Para consultar, agregar, editar, eliminar usuarios del sistema	Alta
3	Como administrador	Quiero acceder al módulo de compras	Para consultar, agregar, editar, eliminar las compras de la empresa	Alta
4	Como administrador	Quiero acceder al módulo de ventas	Para consultar, agregar, editar, eliminar las ventas de la empresa	Alta
5	Como administrador	Quiero acceder al módulo de inventario	Para consultar, agregar, editar, eliminar los productos de la empresa	Alta
6	Como cliente	Quiero acceder a una tienda en línea	Para realizar compras sin necesidad de ir a un establecimiento	Alta

Tabla 10-04. *Historias de Usuario.* (Cisneros & Vaca, 2022)

La recopilación completa de estas historias de usuario se encuentra en los anexos del proyecto.

Lista de funcionalidades

Las funcionalidades que deben cumplir los dos sistemas se encuentran detalladas en el **product backlog**, mismas que surgen a partir de las historias de usuario documentadas previamente. Dichas funcionalidades están clasificadas principalmente según el rol que lo utilizará y también en el número estimado de **sprint** en el que serán implementadas. A continuación se presenta una tabla con ciertas funcionalidades de alta prioridad.

#	Descripción	Rol	Prioridad	Sprint	Criterio de Aceptación
1	Crear una pantalla para el inicio de sesión al sistema	Administrador	Alta	1	Solo se ingresa al sistema si el usuario y contraseña son correctos, caso contrario no se permite el acceso
2	Crear una pantalla para la administración de usuarios	Administrador	Alta	1	Se puede visualizar en una tabla todos los usuarios y a través de distintos botones crear y editar un registro
3	Crear una pantalla para la administración de productos	Administrador	Alta	2	Se puede visualizar en una tabla todos los productos y a través de distintos botones crear y editar un registro
4	Crear una pantalla para la administración de compras	Administrador	Alta	3	Se puede visualizar en una tabla todas las compras y a través de distintos botones crear y editar un registro
5	Implementar un carrito de compras	Cliente	Alta	4	Permite agregar múltiples productos, eliminarlos, modificar sus cantidades y calcular un precio total de la compra

Tabla 11-04. *Product Backlog.* (Cisneros & Vaca, 2022)

En los anexos del proyecto se encuentra toda la lista completa de funcionalidades.

Configuración previa al primer sprint

Antes de comenzar con el desarrollo de los sistemas, se realizó la conexión en red de dos equipos montados sobre un software de virtualización. Cada uno de estos equipos cumple un propósito en específico y cuenta con su propio ambiente de desarrollo sobre el sistema operativo Linux, además de incluir sus respectivas herramientas de *IDE's*¹³, *frameworks* y servidores de desarrollo.

¹³ **IDE:** Integrated Development Environment

En el primer equipo o “máquina 1”, se implementó tanto el sistema de gestión de la base de datos PostgreSQL como el servidor de desarrollo Uvicorn con su respectiva arquitectura basada en microservicios, los cuales en conjunto componen el **backend** de la aplicación. Por otro lado, en el equipo o “máquina 2” se realizó el consumo del API para el desarrollo del **frontend** del sistema, el mismo que cuenta con un servidor de renderizado (SSR) para los componentes con los que interactuará el usuario.

La siguiente ilustración explica la integración de los dos equipos mencionados anteriormente con sus respectivas funciones.

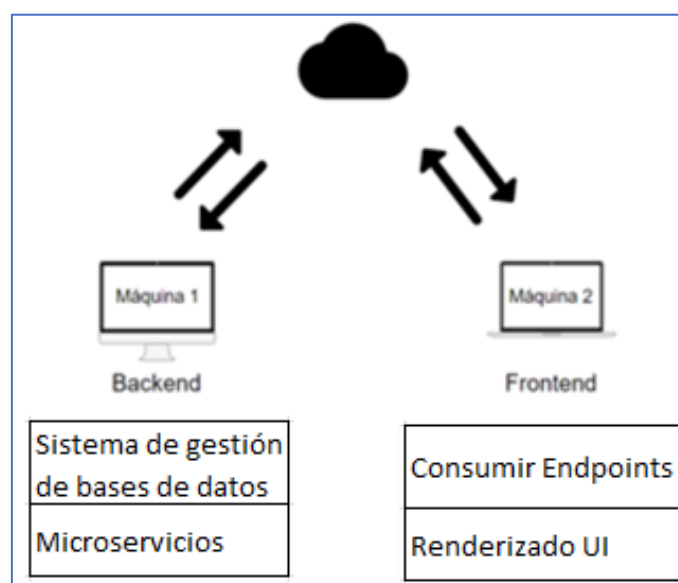


Figura 27-04. Sistema de virtualización (Cisneros & Vaca, 2022)

4.1 Administración del sistema

La administración del sistema de información contempla los requerimientos relacionados con la gestión de usuarios, roles, permisos, módulos y submódulos. Además, incluye todas aquellas pantallas y funcionalidades detalladas en el **product backlog** pertenecientes al **sprint 1**, siendo la más importante la interfaz para el inicio de sesión.

4.1.1 Sprint 1

Este *sprint* tiene como objetivo principal la entrega del primer incremento del producto que corresponde a la administración del sistema. Para esto, se presenta en la siguiente tabla la lista de funcionalidades determinadas para este primer *sprint* con sus tareas asociadas.

#	Descripción	Tareas	Estatus
1	Crear una pantalla para el inicio de sesión	Conectar al microservicio de Login Implementar alertas informativas	Por iniciar
2	Crear la página principal con un dashboard	Conectar al microservicio de Administración Crear un menú para acceder a los módulos Crear paneles con resúmenes de los módulos	Por iniciar
3	Crear una pantalla para la administración de usuarios	Implementar una tabla con paginación para mostrar registros Implementar formulario para agregar y editar un registro Implementar una ventana modal para confirmar el cambio de estado de un registro	Por iniciar
4	Crear una pantalla para la administración de roles	Implementar una tabla con paginación para mostrar registros Implementar formulario para agregar y editar un registro Implementar una ventana modal para confirmar la eliminación de un registro	Por iniciar
5	Crear una pantalla para la administración de permisos	Implementar una tabla con paginación para mostrar registros Implementar formulario para agregar y editar un registro Implementar una ventana modal para confirmar la eliminación de un registro	Por iniciar
6	Crear una pantalla para la administración de módulos	Implementar una tabla con paginación para mostrar registros Implementar formulario para agregar y editar un registro Implementar una ventana modal para confirmar la eliminación de un registro	Por iniciar
7	Crear una pantalla para la administración de submódulos	Implementar una tabla con paginación para mostrar registros Implementar formulario para agregar y editar un registro Implementar una ventana modal para confirmar la eliminación de un registro	Por iniciar

Tabla 12-04. *Sprint Backlog 1 Administración-1. (Cisneros & Vaca, 2022)*

Para completar todas las funcionalidades acordadas anteriormente, se implementó en el lado del *backend* los microservicios y *endpoints* necesarios para el manejo de la información de todos los submódulos. Mientras que en el *frontend*, se presentó todos los datos al usuario en distintos componentes como formularios y tablas, en los que también se realizaron sus respectivas validaciones.

A continuación, se explicará más a detalle los elementos más importantes que fueron implementados durante este primer *sprint*.

Microservicio

Todo microservicio creado se basa en la siguiente estructura de archivos:

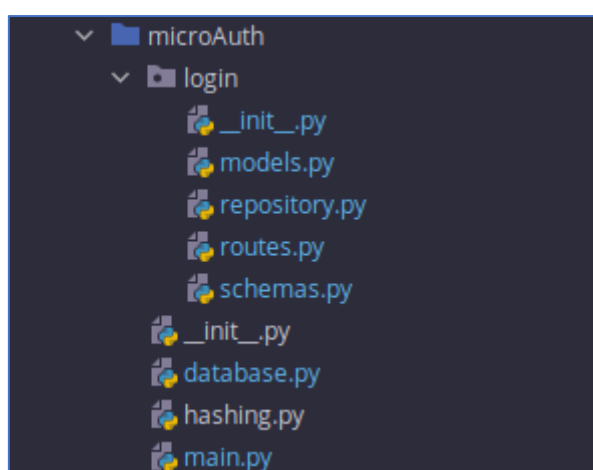


Figura 28-04. Estructura del microservicio. (Cisneros & Vaca, 2022)

Como se observa, la carpeta raíz consta de la palabra “micro” más el nombre del microservicio. El archivo `main.py` dentro de la carpeta permite registrar los módulos que contendrá el microservicio, así como inicializar el servidor Uvicorn. Además, existen los archivos `database.py` que contiene toda la información para realizar la conexión a la base de datos mediante el ORM¹⁴ SQLAlchemy permitiendo ejecutar las distintas operaciones en la base

¹⁴ **ORM:** Object-Relational mapping

desde el *backend*, y el archivo `hashing.py` que posee el algoritmo de encriptación y desencriptación para el correcto manejo de la contraseña.

En la estructura que se maneja, cada directorio corresponde a un submódulo, el mismo que contiene los siguientes archivos:

- `Routes.py`: donde se crean los *endpoints* o puntos de entrada junto con las HTTP *request* según la lógica de cada submódulo.
- `Repository.py`: donde se realiza todas las operaciones a la base de datos, junto a un manejo y validación de la información.
- `Schemas.py`: contiene la estructura de cómo se gestiona los diccionarios con su respectiva clave-valor, manejando el cuerpo de las solicitudes HTTP.
- `Models.py`: que permite manejar las tablas de la base de datos como objetos mediante el ORM SQLAlchemy para hacer consultas con funciones ya elaboradas.

Esta estructura con sus respectivos archivos mencionados son la base de todos los siguientes microservicios, con la única diferencia en el código que se encuentra dentro de cada uno de los archivos. De esta manera, se permite una fácil detección y solución de errores.

El microservicio Login es utilizado para la autenticación y autorización de cada usuario al iniciar sesión en el sistema, ya que solicita su nombre de usuario y contraseña. Si los datos ingresados son correctos se devuelve un JWTToken ¹⁵ que contiene toda la información del usuario, tanto rol como permisos de cada submódulo correspondiente a una autorización. En caso de que las credenciales ingresadas sean incorrectas, se devuelve un mensaje informando el error.

¹⁵ **JWT**: JSON Web Token

El microservicio Administración contiene los *endpoints* que permiten controlar los permisos y autorizaciones que se le asigna a cada rol dependiendo del submódulo. A diferencia del microservicio anterior, este contiene un archivo llamado security.py que es el encargado de consultar los permisos que cuenta el rol asignado a ese usuario, siendo el que verifica si tiene permitido el acceso a un submódulo y qué acciones puede realizar.

Endpoints para el microservicio de Administración

Para mostrar la información almacenada en la base de datos correspondiente a cada submódulo, se crearon los siguientes puntos de entrada que posteriormente serán consumidos desde el *frontend* mediante peticiones.

Microservicio	HTTP Request	Ruta Endpoint	Acción
Login	POST	/login	
Administración	GET	/roles	Leer Roles
Administración	GET	/roles/{id_rol}	Leer Rol
Administración	POST	/roles	Registrar Rol
Administración	PUT	/roles/{id_rol}	Editar Rol
Administración	DELETE	/roles/{id_rol}	Eliminar Rol
Administración	GET	/submodulos	Leer Submódulos
Administración	GET	/submodulos/{id_submodulo}	Leer Submódulo
Administración	POST	/submodulos	Registrar Submódulo
Administración	PUT	/submodulos/{id_submodulo}	Editar Submódulo
Administración	DELETE	/submodulos/{id_submodulo}	Eliminar Submódulo
Administración	GET	/permisos/{id_rol}	Leer Permisos
Administración	GET	/permiso/{id_permiso}	Leer Permiso
Administración	POST	/permiso	Registrar Permiso
Administración	PUT	/permiso/{id_permiso}	Editar Permiso
Administración	DELETE	/permiso/{id_permiso}	Eliminar Permiso
Administración	GET	/usuarios	Leer Usuarios
Administración	GET	/usuario/{id_usuario}	Leer Usuario

Administración	POST	/usuarios	Registrar Usuario
Administración	PUT	/usuario/{id_usuario}/editarPassword	Editar Password Usuario
Administración	PUT	/usuario/{id_usuario}/editarRol	Editar Rol Usuario
Administración	DELETE	/usuario/{id_usuario}	Cambiar Estado Usuario
Administración	GET	/modulos	Leer Módulos
Administración	GET	/modulos/{id_modulo}	Leer Módulo
Administración	POST	/modulos	Registrar Módulo
Administración	PUT	/modulos/{id_modulo}	Editar Módulo
Administración	DELETE	/modulos/{id_modulo}	Eliminar Módulo

Tabla 13-04. Endpoints para el primer sprint. (Cisneros & Vaca, 2022)

Rutas y Componentes

Nuxt.js permitió crear todas las páginas del módulo de forma sencilla sin tener que enrutarlas manualmente. De esta manera, se añadió una página por cada submódulo existente dentro del módulo de administración tal como se observa a continuación.

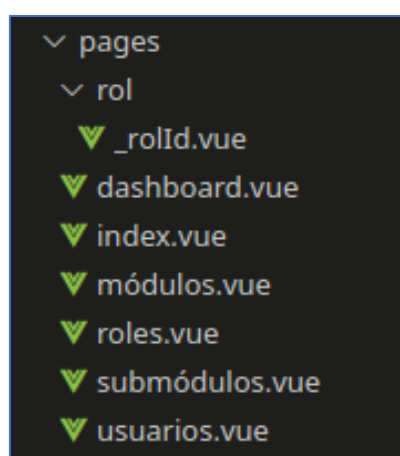


Figura 29-04. Rutas del módulo. (Cisneros & Vaca, 2022)

En caso de requerir pasar un parámetro de una página a otra como en el caso de roles, se creó una carpeta con el archivo del nombre del parámetro precedido por un guion bajo según se muestra en la figura anterior. Con esta sintaxis *Nuxt* entiende la instrucción y se encarga de almacenar por sí mismo dicho dato en la ruta para su posterior uso en una nueva página.

Por otro lado, también se crearon ciertos componentes para que sean reutilizables en todas las páginas del proyecto. Como se observa en la siguiente figura, estos corresponden al menú lateral, la barra de navegación, y un modal informativo en caso de haber expirado el tiempo de validez del token de autenticación. Estos serán llamados para renderizarse solamente cuando sean requeridos.

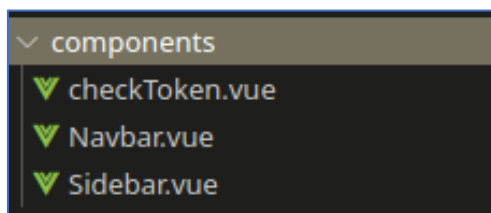


Figura 30-04. Componentes. (Cisneros & Vaca, 2022)

El componente de la barra de navegación es el único que recibe dos propiedades al momento de ser llamado, estas ayudan a cambiar los valores de su contenido según lo enviado. En este caso, permiten indicar el módulo y submódulo en el que se encuentra un usuario lo que siempre cambia de acuerdo con la página a la que este acceda. Un ejemplo de estas propiedades se muestra a continuación.

```
<div class="g-sidenav-show bg-gray-100 vh-completa" id='mainDashboard'>
  <Sidebar/>
  <Navbar :Modulo='Inicio' :Tabla='Dashboard' />
  <main class="main-content position-relative max-height-vh-">...
  </main>
</div>
```

Figura 31-04. Propiedades de un componente. (Cisneros & Vaca, 2022)

Interfaces

Para que la implementación de las interfaces del sistema sea más sencilla, se utilizó como base una plantilla gratuita de Creative Tim llamada Soft Ui Dashboard. Esta proporciona un diseño moderno que puede ser modificado y ajustado a las ideas de interfaces que fueron establecidas

previamente, y lo más importante es que permite que todas estas pantallas sean adaptables a distintos tipos de dispositivos.

Debido a que su estructura está hecha con componentes de HTML¹⁶ y Bootstrap 5, no requería de grandes cambios para su integración al proyecto. Simplemente se copiaron todos estos componentes en las páginas que corresponden según su uso y se añadió el respectivo archivo CSS dentro de la carpeta *assets* del proyecto junto con las imágenes necesarias.

La figura a continuación muestra la nueva estructura de la carpeta *assets* después de añadir los recursos necesarios.

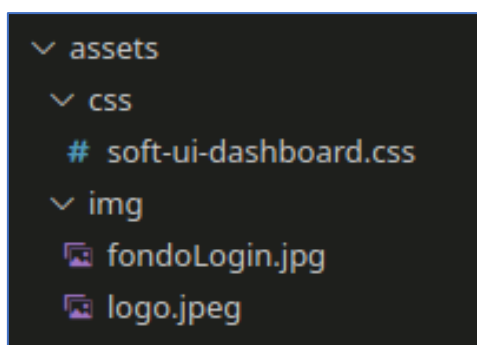


Figura 32-04. *Activos del proyecto.* (Cisneros & Vaca, 2022)

Además de estos componentes, también se hizo uso de algunos otros provenientes de la dependencia instalada de Bootstrap Vue.

Token de autenticación

Cómo se mencionó anteriormente, al momento de tener un inicio de sesión satisfactorio se nos devuelve un JW token. Dentro de este token se almacenó toda la información del usuario, tanto su nombre como su id, junto con sus permisos correspondientes a cada módulo. Dicho token presenta la siguiente estructura:

¹⁶ **HTML:** HyperText Markup Language

```
{
  "id_usuario": 8,
  "nombre_usuario": "David",
  "autorizaciones": {
    "rol": "Comprador",
    "modulos": {
      "Compras": {
        "Proveedores": {
          "crear": true,
          "leer": true,
          "editar": true,
          "eliminar": true
        }
      }
    }
  },
  "exp": 1648688565
}
```

Figura 33-04. Estructura del JW Token. (Cisneros & Vaca, 2022)

Como se observa en la figura, el token contiene solamente los nombres de los módulos y submódulos a los que tiene acceso el rol asignado a dicho usuario. En caso de no tener permiso de realizar una acción, esta no es enviada. Además, por seguridad, la información está firmada mediante el algoritmo de encriptación HS256 y con una clave privada para comprobar que el token sea válido. Y cuenta con un tiempo de expiración que en este caso es de 4h.

En el lado del *frontend*, gracias al propio módulo de autenticación de Nuxt.js, este token es guardado automáticamente en el almacenamiento local del navegador, y permanece ahí mientras su tiempo de expiración está vigente; de lo contrario, se lo elimina. A continuación se muestra el almacenamiento descrito.

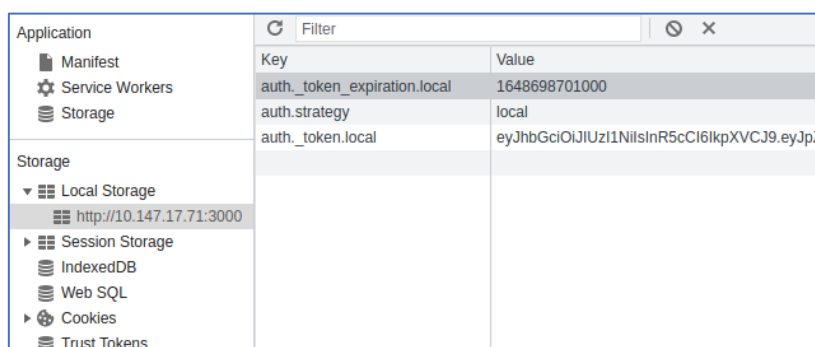


Figura 34-04. Almacenamiento local. (Cisneros & Vaca, 2022)

Con el token guardado, se procede a decodificarlo con la ayuda de la librería JWT-decode para poder leer la información que contiene. De este se extraen tanto el nombre del usuario registrado así como también todos los permisos que posee para acceder a los distintos módulos y submódulos. La siguiente figura ilustra las funciones que se utilizaron para realizar las acciones descritas previamente.

```
import jwt_decode from "jwt-decode";

export function getAccessToken(){
  return localStorage.getItem('auth._token.local')
}
export function getDecoded(){
  if(getAccessToken()){
    let token_decoded = jwt_decode(getAccessToken())
    return token_decoded
  }
}
export function getUser(){
  let decoded = getDecoded()
  let user = decoded.nombre_usuario
  return user
}
export function getMenuAuth(){
  let decoded = getDecoded()
  let menu = decoded.autorizaciones.modulos
  return menu
}
```

Figura 35-04. *Funciones de Autenticación.* (Cisneros & Vaca, 2022)

Ya con los permisos del usuario se modificó el menú lateral del sistema para que sea dinámico y pueda mostrar solamente los módulos a los que dicho usuario tiene acceso. Además, se controla las acciones correspondientes a lectura, creación, edición y modificación de un registro en un submódulo en específico. Para esto, se oculta y des oculta botones relacionados con las acciones según los permisos obtenidos del token.

A continuación se muestra otra función importante que se realiza con la ayuda del módulo de autenticación de Nuxt.js. Esta se encuentra dentro de la carpeta Middleware del proyecto para que se ejecute antes de renderizar la página en la que es llamada y permite verificar que se haya iniciado una sesión. Es decir, solo se podrá acceder a las páginas después de que un usuario

inicie sesión correctamente, caso contrario siempre lo redireccionará a la pantalla de inicio para autenticarse.

```
export default function ({ store, redirect }) {  
  if (!store.state.auth.loggedIn) {  
    return redirect('/')  
  }  
}
```

Figura 36-04. Middleware de autenticación. (Cisneros & Vaca, 2022)

De igual manera, se implementó otra función para verificar la expiración del token la cual arroja un modal después de exceder el tiempo de validez. Este modal presenta un mensaje al usuario de que su tiempo de sesión ha expirado y que debe volver a iniciar sesión. Después de confirmar el mensaje, lo redirecciona automáticamente a la página de inicio. Para esto, se extrajo del token decodificado el valor de expiración y se lo convirtió en fecha para saber en qué tiempo este perdería su validez. Finalmente, a este valor se le resta la fecha actual para descontar el tiempo que ha pasado desde que se generó el token y así obtener el tiempo que queda validez. La figura a continuación muestra el código implementado para realizar la función descrita, la misma que es llamada en todas las páginas del sistema.

```
async mounted(){  
  var decodedToken = getDecoded()  
  var date = new Date( decodedToken.exp *1000)  
  setTimeout(()=>{  
    this.showModalToken()  
  }, date - Date.now());  
},
```

Figura 37-04. Verificación expiración de Token JW. (Cisneros & Vaca, 2022)

Gestión de los datos

Para poder presentar al cliente la información proveniente de la base de datos, se hizo uso del módulo Axios de Nuxt.js que es un cliente HTTP que permite realizar solicitudes del lado del

servidor. Para esto, primero se configuró la URL a través de la cual se obtendrá los datos, la misma que corresponde a los *endpoints* explicados anteriormente pudiendo variar en el tipo de solicitud dependiendo de la acción a ejecutar. Además, en esta petición es necesario incluir el token de acceso dentro de la cabecera ya que todo es controlado con permisos y autenticaciones.

La siguiente figura muestra un ejemplo de una función asíncrona que usa Axios en este caso para una petición de tipo GET que devuelve la información de los usuarios del sistema. Esta contiene dos partes principales, el *then* donde se colocan todas las acciones a realizarse cuando se ha obtenido una respuesta exitosa y el *catch*, para capturar cualquier tipo de error que ha surgido en la solicitud.

```
async getUsuarios(){
  await axios.get('/usuarios',{headers:{Authorization:'Bearer ' + getAccessToken()}})
  .then(response => {
    if(response.data !==null)
      this.usuarios = response.data
    else
      this.error = true
  }) .catch(e => {
    this.$toast.error(e.response.data.detail)
  })
},
```

Figura 38-04. Ejemplo solicitud con Axios. (Cisneros & Vaca, 2022)

Siguiendo esta misma estructura, se realizaron las distintas solicitudes al servidor para obtener toda la información correspondiente al módulo de administración y sus múltiples submódulos. Con la diferencia en los tipos de solicitud, ya sean *post*, *get*, *put* o *delete*, y también de la url asociada a cada *endpoint*.

El código completo implementando para este *sprint* se encuentra en los anexos del proyecto.

4.1.2 Revisión del Sprint 1

Una vez finalizado el *sprint* se tuvo la primera reunión con todos los interesados del proyecto junto con el usuario funcional con conocimiento técnico. Esta reunión consistió principalmente en la revisión del funcionamiento del producto entregado, el mismo que correspondía a la administración del sistema involucrando el manejo de usuarios, roles, permisos, módulos y submódulos. A continuación, se presenta un resumen de las observaciones obtenidas, las cuales serán explicadas con mayor profundidad más adelante.

¿Qué salió bien en la iteración?	¿Qué no salió bien en la iteración?	¿Qué mejoras vamos a implementar en la próxima iteración?
Inicio de sesión	Cambiar el estado de un usuario	Listar los usuarios activos e inactivos del sistema
Registro y edición de usuarios	Visualización de los usuarios del sistema	Cambiar el ícono de eliminar un usuario
Registro y edición de permisos	Identificación de los permisos	Cambiar el nombre de la acción asociada a un permiso
Administración de roles		
Administración de módulos		
Administración de submódulos		

Tabla 14-04. *Sprint Restrospective 1. (Cisneros & Vaca, 2022)*

En primer lugar, el usuario final verificó la funcionalidad del inicio de sesión al sistema con múltiples casos considerando datos incorrectos, incompletos y por supuesto, con credenciales válidas. Al no encontrarse inconvenientes, continuó con la administración de los usuarios. En este caso, se presentó la única novedad al momento de eliminar un usuario, ya que este no debe eliminarse como tal sino simplemente cambiar su estado actual a inactivo.

Por otro lado, en la gestión de roles también se comprobó que esta funcionara correctamente, en especial que controlara las eliminaciones de un rol al que estuviera asociado un usuario. Sin encontrar ninguna observación, se procedió a verificar los permisos de un rol. Estos se comportan de manera correcta, sin embargo, el usuario funcional requirió que los nombres de cada uno de ellos estuvieran mejor detallados. Finalmente se realizaron las pruebas en la administración tanto de módulos como submódulos en las que los resultados fueron satisfactorios.

4.1.3 Sprint 2

El segundo *sprint* de la administración del sistema contempló el desarrollo de las correcciones presentadas por el usuario final, las cuales fueron detalladas en el apartado anterior. Para esto, se presenta a continuación el correspondiente *product backlog* del *sprint* con el estado de los requerimientos actualizado.

#	Descripción	Tareas	Estatus
1	Crear una pantalla para el inicio de sesión	Conectar al microservicio de Login Implementar alertas informativas	Completado
2	Crear la página principal con un dashboard	Conectar al microservicio de Administración Crear un menú para acceder a los módulos Crear paneles con resúmenes de los módulos	En proceso
3	Crear una pantalla para la administración de usuarios	Implementar una tabla con paginación para mostrar registros Implementar formulario para agregar y editar un registro Implementar una ventana modal para confirmar la eliminación de un registro	En proceso
4	Crear una pantalla para la administración de roles	Implementar una tabla con paginación para mostrar registros Implementar formulario para agregar y editar un registro Implementar una ventana modal para confirmar la eliminación de un registro	En proceso
5		Implementar una tabla con paginación para mostrar registros	En proceso

	Crear una pantalla para la administración de permisos	Implementar formulario para agregar y editar un registro Implementar una ventana modal para confirmar la eliminación de un registro	
6	Crear una pantalla para la administración de módulos	Implementar una tabla con paginación para mostrar registros Implementar formulario para agregar y editar un registro Implementar una ventana modal para confirmar la eliminación de un registro	Completado
7	Crear una pantalla para la administración de submódulos	Implementar una tabla con paginación para mostrar registros Implementar formulario para agregar y editar un registro Implementar una ventana modal para confirmar la eliminación de un registro	Completado

Tabla 15-04. *Sprint Backlog 1 Administración-2. (Cisneros & Vaca, 2022)*

Entre las modificaciones realizadas, la más importante consistió en corregir la función de eliminar un usuario por la de cambiar su estado de activo a inactivo o viceversa. Para esto, se editó el ícono actual por otro más representativo y se aumentó al momento de listar para que aparezcan todos los usuarios sin importar su estado. La figura siguiente muestra el resultado final de esta modificación.

ID	NOMBRE	ROL	ESTADO	
1	dxeox	Administrador	ACTIVO	✎ ↓↑
2	Rolsito	Bodega	ACTIVO	✎ ↓↑
3	VendedorUser	Vendedor	ACTIVO	✎ ↓↑
4	Vendedorsito	Vendedor	ACTIVO	✎ ↓↑
5	admin	Administrador	ACTIVO	✎ ↓↑
8	David	Comprador	ACTIVO	✎ ↓↑
9	dani	ContadorEditado	ACTIVO	✎ ↓↑
11	Prueba1	ContadorEditado	INACTIVO	✎ ↓↑
12	silvana	Administrador	INACTIVO	✎ ↓↑
13	Alberto	Vendedor	ACTIVO	✎ ↓↑

Figura 39-04. *Usuarios activos e inactivos del sistema. (Cisneros & Vaca, 2022)*

La siguiente corrección corresponde únicamente al cambio en el nombre de los permisos asociados a un determinado rol. Como se observa en la figura más adelante, cada permiso de un submódulo representa una de las acciones para registrar, ver, modificar y eliminar un registro en particular.

ID	SUBMÓDULO	PERMISOS					
3	Clientes	<input checked="" type="checkbox"/> Registrar	<input checked="" type="checkbox"/> Ver	<input checked="" type="checkbox"/> Modificar	<input checked="" type="checkbox"/> Eliminar		
4	Proveedores	<input checked="" type="checkbox"/> Registrar	<input checked="" type="checkbox"/> Ver	<input checked="" type="checkbox"/> Modificar	<input checked="" type="checkbox"/> Eliminar		
18	Ventas	<input checked="" type="checkbox"/> Registrar	<input checked="" type="checkbox"/> Ver	<input checked="" type="checkbox"/> Modificar	<input checked="" type="checkbox"/> Eliminar		
5	Usuarios	<input checked="" type="checkbox"/> Registrar	<input checked="" type="checkbox"/> Ver	<input checked="" type="checkbox"/> Modificar	<input checked="" type="checkbox"/> Eliminar		
6	Roles	<input checked="" type="checkbox"/> Registrar	<input checked="" type="checkbox"/> Ver	<input checked="" type="checkbox"/> Modificar	<input checked="" type="checkbox"/> Eliminar		
7	Módulos	<input checked="" type="checkbox"/> Registrar	<input checked="" type="checkbox"/> Ver	<input checked="" type="checkbox"/> Modificar	<input checked="" type="checkbox"/> Eliminar		
8	Submódulos	<input checked="" type="checkbox"/> Registrar	<input checked="" type="checkbox"/> Ver	<input checked="" type="checkbox"/> Modificar	<input checked="" type="checkbox"/> Eliminar		
17	Permisos	<input checked="" type="checkbox"/> Registrar	<input checked="" type="checkbox"/> Ver	<input checked="" type="checkbox"/> Modificar	<input checked="" type="checkbox"/> Eliminar		
20	Productos	<input checked="" type="checkbox"/> Registrar	<input checked="" type="checkbox"/> Ver	<input checked="" type="checkbox"/> Modificar	<input checked="" type="checkbox"/> Eliminar		
21	Bodega	<input checked="" type="checkbox"/> Registrar	<input checked="" type="checkbox"/> Ver	<input checked="" type="checkbox"/> Modificar	<input checked="" type="checkbox"/> Eliminar		

Figura 40-04. *Permisos de un rol.* (Cisneros & Vaca, 2022)

4.1.4 Revisión del Sprint 2

Con un nuevo y mejorado producto que fue entregado al finalizar el segundo *sprint* correspondiente a la administración del sistema, se procedió a realizar una nueva reunión con el usuario funcional. Afortunadamente, no se presentó ninguna observación adicional por parte de este usuario; por lo que se concluye que se han cumplido satisfactoriamente todos los requerimientos de administración y se puede continuar con el desarrollo del siguiente módulo.

4.2 Módulo Inventario

Como se ha mencionado anteriormente, el módulo inventario comprende el manejo de todos los productos e ítems de la empresa. Este permite llevar un control y registro de todo lo que se encuentra en la bodega junto con la información detallada de cada uno de los productos. Es

fundamental la realización de este módulo antes de los de compras y ventas, ya que estas transacciones dependen directamente de los productos, los cuales pueden ser sumados o restados del inventario de la empresa.

4.2.1 Sprint 2

El segundo *sprint* continuó con el desarrollo del módulo de inventario tomando como base las funcionalidades determinadas a cumplirse en esta iteración, las cuales serán detalladas en la tabla a continuación.

#	Descripción	Tareas	Estatus
1	Crear una pantalla para la administración de ítems	Conectarse al microservicio de Inventario	Por iniciar
		Implementar una tabla con paginación para mostrar registros	
		Implementar formulario para editar un registro	
2	Crear una pantalla para la administración de productos	Implementar una tabla con paginación para mostrar registros	Por iniciar
		Implementar formulario para agregar y editar un registro	
		Implementar un buscador de productos	

Tabla 16-04. *Sprint Backlog 2 Inventario-1. (Cisneros & Vaca, 2022)*

Para cumplir todas estas funcionalidades se utilizaron en su gran mayoría los mismos componentes del *sprint* 1 con sus respectivos cambios para adaptarse a este nuevo módulo. Agregándole a estos cambios, la creación de un nuevo microservicio para el manejo únicamente de la información del inventario de la empresa junto con sus *endpoints* respectivos.

La siguiente tabla muestra los distintos puntos de entrada que fueron creados en este nuevo *sprint* para que sean consumidos en el *frontend* según sean requeridos.

Microservicio	HTTP Request	Ruta Endpoint	Acción
Inventario	GET	/productos	Leer Productos
Inventario	GET	/producto/{id_producto}	Leer Producto
Inventario	GET	/productos/findByWord/{texto}	Buscar Producto
Inventario	GET	/productosExistentes	Obtener Productos Existentes
Inventario	GET	/productosNombres	Obtener Nombres de Productos
Inventario	GET	/detalleProducto/venta/findByNombre/{nombre_producto}	Obtener Detalle de Productos
Inventario	POST	/producto	Registrar Producto
Inventario	PUT	/producto/{id_producto}	Editar Producto
Inventario	GET	/items	Leer Ítems
Inventario	GET	/item/{id_item}	Leer Ítem
Inventario	PUT	/item/{id_item}	Editar Ítem

Tabla 17-04. Endpoints módulo inventario. (Cisneros & Vaca, 2022)

Manejo de Imágenes

A diferencia del anterior *sprint*, este módulo presenta un nuevo tipo de dato el mismo que requería ser manejado de distinta manera. Se trata del campo para mostrar imágenes en el submódulo de productos. Este permite identificar de manera visual a cada uno de ellos, por lo que fue importante poder guardar no solo una sino varias imágenes asociadas a un mismo producto. Además, estas imágenes también servirán en el comercio electrónico para que un cliente pueda observar múltiples fotografías del producto que desea comprar.

Cabe recalcar que estas imágenes no fueron guardadas directamente en la base de datos, sino solo la ruta de su ubicación. Es decir, el nombre de la carpeta que en este caso corresponde a la identificación del producto junto al nombre de la imagen. Se muestra a continuación un ejemplo de una carpeta asociada a un producto con sus respectivas imágenes y diferentes dimensiones.

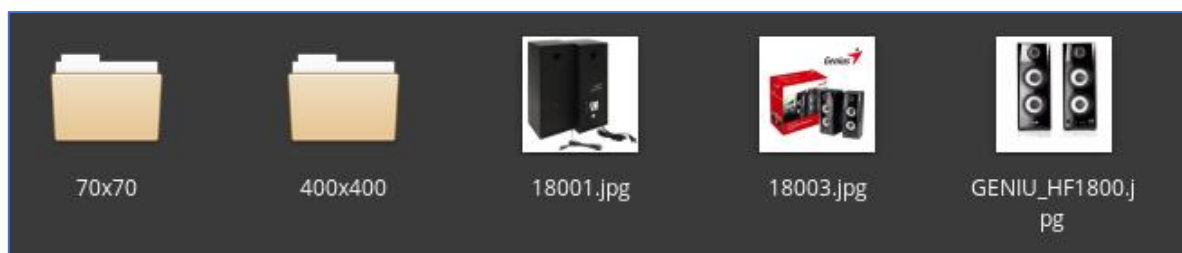


Figura 41-04. Imágenes de un producto. (Cisneros & Vaca, 2022)

Para redimensionar las imágenes, se utilizó en el lado del **backend** de la librería de **python** llamada “PIL”, la misma que permite modificar el tamaño actual de una imagen a un otro deseado. En este caso, se implementó la siguiente función detallada en la figura a continuación para poder obtener imágenes de un mismo tamaño siendo de 70 y 400 pixeles.

```
for file in files:
    arreglo_nombresImagenes.append(file.filename)
    with open(f"{PATH_IMAGES}/{secuencia}/{file.filename}", "wb+") as file_object:
        shutil.copyfileobj(file.file, file_object)
    imagen = Image.open(f'{PATH_IMAGES}/{secuencia}/{file.filename}')
    imagen70x70 = imagen.resize((70, 70))
    imagen70x70.save(f'{PATH_IMAGES}/{secuencia}/70x70/{file.filename}')
    imagen400x400 = imagen.resize((200, 200))
    imagen400x400.save(f'{PATH_IMAGES}/{secuencia}/400x400/{file.filename}')
```

Figura 42-04. Función para redimensionar imágenes. (Cisneros & Vaca, 2022)

Por otro lado, en el **frontend**, se hizo uso del objeto **formData** para enviar todo el contenido del formulario, en especial lo de tipo archivo. Gracias a la ayuda del evento **onChange** aplicado, se lograron capturar múltiples archivos, los mismos que fueron limitados para recibir solo aquellos de tipo imagen. De esta manera, se envían al **backend** en un arreglo todas las imágenes seleccionadas junto con el nombre del archivo. La función mostrada en la siguiente figura permite realizar el proceso descrito anteriormente.

```

onChange(e){
  this.imagen = e.target.files
  const formData = new FormData()
  for (let i = 0; i < this.imagen.length; i++){
    formData.append('files',this.imagen[i])
  }
},

```

Figura 43-04. Función para guardar archivos. (Cisneros & Vaca, 2022)

Para el resto de los datos involucrados, se utilizaron las mismas funciones para hacer las distintas solicitudes al servidor que fueron detalladas en el apartado 4.1.1. Estas permiten visualizar, crear y editar un registro de los submódulos de productos e ítems de la bodega pertenecientes al módulo inventario.

El código completo implementando para este *sprint* se encuentra en los anexos del proyecto.

4.2.2 Revisión del Sprint 2

Durante la revisión del producto entregado correspondiente al segundo *sprint*, que contó además con la presencia del usuario funcional, se obtuvieron ciertas observaciones únicamente en el submódulo de productos. Estas se encuentran resumidas en la siguiente tabla.

¿Qué salió bien en la iteración?	¿Qué no salió bien en la iteración?	¿Qué mejoras vamos a implementar en la próxima iteración?
Listado de ítems	Listar productos	Estandarizar el tamaño de las imágenes
Edición de un ítem	Visualizar imágenes de un producto	Agregar imágenes de baja y alta calidad
Registro de un producto		Cambiar la forma de listar las imágenes al editar
Edición de un producto		
Buscar un producto		

Tabla 18-04. Sprint Restrospective 2. (Cisneros & Vaca, 2022)

Al momento de listar todos los productos de la empresa cada uno con su respectiva imagen, estas eran mostradas en su tamaño original por lo que no existía uniformidad en la presentación. Por esta razón, se precisó que se estandarizara el tamaño de las imágenes mostradas y de ser posible, tener dos de ellas. Una imagen pequeña y de menor calidad para mostrar en la tabla de registros y otra con mayor calidad para utilizarla al momento de editar dicho producto. En esta edición, también se sugirió cambiar la presentación actual de las imágenes de un mismo producto por algún componente extra que permitiera visualizarlas todas sin tener que desplazarse a lo largo de la página.

4.2.3 Sprint 3

El siguiente *sprint* para este módulo estuvo enfocado en las correcciones de las observaciones presentadas en la última reunión. Por lo tanto, se muestra en la tabla a continuación los requerimientos con sus estados actualizados para esta nueva iteración.

#	Descripción	Tareas	Estatus
1	Crear una pantalla para la administración de ítems	Conectarse al microservicio de Inventario Implementar una tabla con paginación para mostrar registros Implementar formulario para editar un registro	Completado
2	Crear una pantalla para la administración de productos	Implementar una tabla con paginación para mostrar registros Implementar formulario para agregar y editar un registro Implementar un buscador de productos	En proceso

Tabla 19-04. *Sprint Backlog 2 Inventario-2. (Cisneros & Vaca, 2022)*

Tal como fue solicitado, se mejoró la forma de presentar el listado de los productos con imágenes de un mismo tamaño y de menor calidad. Para realizar esta estandarización se lo hizo desde el *backend* manejando copias a distintos tamaños. Los tamaños escogidos son de 70x70 y 400x400, además de guardar una copia de la imagen original

El resultado final de esta modificación se muestra a continuación.

ID	IMAGEN	NOMBRE	DETALLE	MARCA	CANTIDAD
79		Laptop Asus x454	PROCESADOR CORE I5 10400u/Memoria 8gb/ SSD 240GB/Pantalla 14'	Asus	0
80		MOUSE USB GENIUS DX-110	NEGRO, 3 BOTONES	GENIUS	0
81		PARLANTES GENIUS SP-U115	SONIDO 2.1 DE 3W PARA PC, NOTEBOOK, IPOD, MP3 / MP4	GENIUS	0

Figura 44-04. Listado de productos. (Cisneros & Vaca, 2022)

Por otro lado, al momento de editar un producto en específico también se modificó la visualización de sus imágenes asociadas. Para esto, se optó por usar un componente externo de carrusel, denominado vueper-slides, para mostrar todas las imágenes sin tener que desplazarnos de la parte central del formulario de edición. El código a continuación muestra cómo fue incorporado este componente a la página del producto, en el que además se le añadieron las funcionalidades para poder interactuar con cualquier imagen que presentara.

```
<div class="col-12 col-lg-6 mt-3 mt-lg-0">
  <p>Imágenes</p>
  <vueper-slides class="no-shadow ex--center-mode" fixed-height="350px" :touchable="false"
    arrows-outside bullets-outside fractions>
    <vueper-slide
      v-for="imagen in this.imagenes"
      :key="imagen"
      :image="`http://10.147.17.173:5002/productos/imagenes/${form.id}/${imagen}`">
      <template #content>
        <a class="trash cursor-pointer vueperslide__content-wrapper" @click="showModalDelete(imagen)">
          <b-icon class="icon" icon='trash' ></b-icon>
        </a>
      </template>
    </vueper-slide>
  </vueper-slides>
</div>
```

Figura 45-04. Componente Vueper slides. (Cisneros & Vaca, 2022)

Este componente permitió que se pudiera seleccionar la imagen que se desea eliminar de entre todas las presentadas en el carrusel y también navegar entre estas con las flechas de ayuda laterales o las viñetas inferiores. También incluye una etiqueta en la parte superior que muestra el número actual de la imagen en relación con el número total de las imágenes cargadas. La figura siguiente ilustra el diseño final de la interfaz para editar un producto con la incorporación del carrusel implementado para la visualización de sus imágenes.

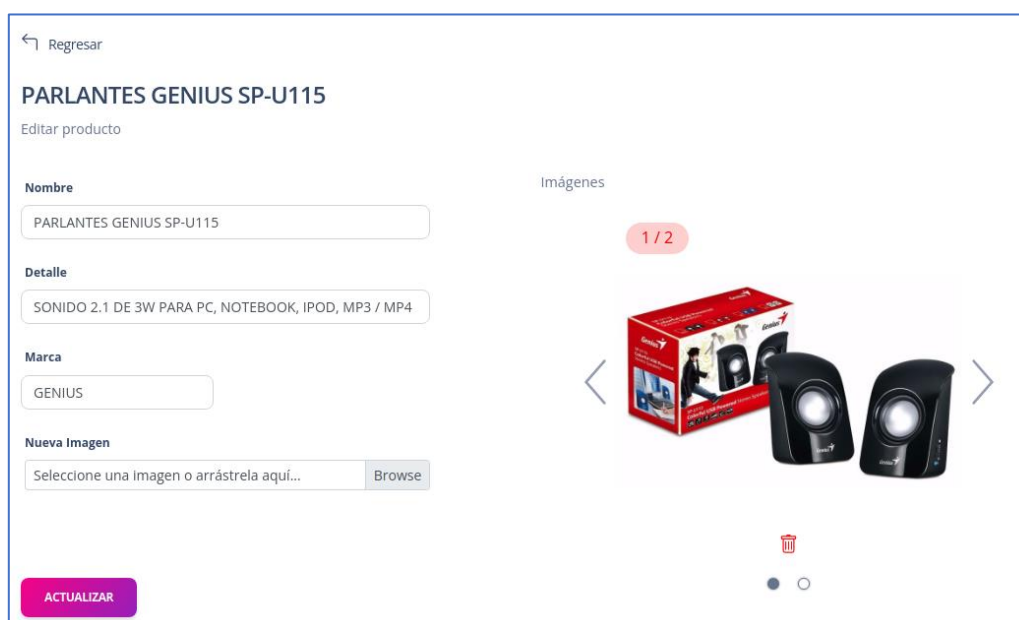


Figura 46-04. Carrusel de imágenes. (Cisneros & Vaca, 2022)

El código completo implementando para este *sprint* se encuentra en los anexos del proyecto.

4.2.4 Revisión del Sprint 3

Con la presencia del usuario funcional, se realizó la siguiente reunión para la revisión del nuevo producto entregado del módulo de inventario. En esta se verificó el cumplimiento de aquellos requerimientos faltantes junto con las modificaciones que fueron solicitadas previamente y al no presentarse ningún inconveniente en estos, se consideró al módulo actual como terminado pudiendo así continuar con la implementación del siguiente modulo.

4.3 Módulo Compras

El presente módulo permite llevar la administración de todas las órdenes de compras de la empresa junto con sus proveedores asociados. Además, este posibilita el ingreso de nuevos ítems, provenientes de dichas compras, al inventario actual para su posterior venta al público. Estos ítems se encuentran relacionados con un producto específico, logrando diferenciarse de otros por su único número de serie.

4.3.1 Sprint 3

La continuación del tercer *sprint* se centró en el desarrollo del módulo de compras de la empresa, el mismo que comprende las distintas funcionalidades que serán detalladas en la siguiente tabla.

#	Descripción	Tareas	Estatus
1	Crear una pantalla para la administración de proveedores	Conectarse al microservicio de compras Implementar una tabla con paginación para mostrar registros Implementar formulario para agregar y editar un registro Implementar una ventana modal para confirmar el cambio de estado de un registro	Por iniciar

		Implementar una tabla con paginación para mostrar registros	
2	Crear una pantalla para la administración de órdenes de compras	Implementar formulario para agregar y editar un registro Implementar interfaz para visualizar los detalles del registro	Por iniciar

Tabla 20-04. *Sprint Backlog 3 Compras-1. (Cisneros & Vaca, 2022)*

Al igual que en el *sprint* anterior, se reutilizaron varios de los componentes que fueron explicados en el apartado 4.1.1 correspondiente al primer *sprint*, los cuales fueron adaptados para completar la lista de requerimientos del presente módulo. De la misma manera, se creó otro microservicio destinado a la gestión de la información de los proveedores y compras de la empresa. Dentro de este, se encuentran todos los nuevos puntos de entrada que fueron creados para el manejo de dicha información en el lado del cliente, los mismos que serán detallados en la tabla a continuación.

Microservicio	HTTP Request	Ruta Endpoint	Acción
Compras	GET	/proveedores	Leer Proveedores
Compras	GET	/proveedor/{id_proveedor}	Leer Proveedor
Compras	GET	/proveedoresHabilitados	Leer Proveedores Habilitados
Compras	POST	/proveedor	Registrar Proveedor
Compras	PUT	/proveedor/{id_proveedor}	Editar Proveedor
Compras	DELETE	/proveedor/{id_proveedor}	Cambiar Estado Proveedor
Compras	GET	/compras	Leer Compras
Compras	GET	/compraDetalle/{id_compra}	Leer Compra con Detalle
Compras	GET	/detalle_compra/{id_detalleCompra}	Leer Detalle Compra
Compras	POST	/compra	Registrar Compra
Compras	PUT	/compra/{id_compra}	Editar Compra

Tabla 21-04. *Endpoints módulo compras. (Cisneros & Vaca, 2022)*

Relación de múltiples tablas

Otro cambio importante implementado que difiere de los *sprints* anteriores se encuentra en las consultas y operaciones realizadas en la base de datos, ya que en esta ocasión se trabajó con tablas que están relacionadas con otras. La siguiente figura muestra un ejemplo de una función que permite unir las tablas compras y detalle de compras para mostrar al usuario los datos de una compra en específica con todos sus detalles asociados, es decir, los productos involucrados.

```
def listar_compra_y_detalle(db: Session, id_compra: int):
    datos = db.query(models.Compras,
                     modelsProveedores.Proveedores.nombre_proveedor) \
        .join(modelsProveedores.Proveedores) \
        .filter(models.Compras.id_compra == id_compra) \
        .options(joinedload(models.Compras.detalle_compra)
                 .joinedload(modelsDetalleCompras.DetalleCompras.productos)
                 .load_only('nombre_producto')) \
        .all()
    verificar_existencia(datos)
    return datos
```

Figura 47-04. Código para listar compras con detalle. (Cisneros & Vaca, 2022)

Adicionalmente, se incorporó otra operación de manipulación de datos para insertar nuevos ítems cuando se efectúa una nueva compra. Por lo tanto, se requirió también del microservicio de inventario para poder implementar los cambios en sus tablas asociadas que en este caso corresponden a productos y bodega. A continuación, se presenta el código utilizado para ejecutar el proceso explicado anteriormente.

```
for item in detalle_compra:
    items.append({
        'precioCompra_item': str(item['precioUnitario_detalleCompra']),
        'productos_id_producto': item['productos_id_producto'],
        'compras_id_compra': secuencia,
        'pvp_item': str(item['pvp_item']),
        'pvd_item': str(item['pvd_item']),
        'descuentoPorcentaje_item': item['descuentoPorcentaje_item'],
        'numeroSerie_item': item['numeroSerie_item']})
verificar_item_inventario(items)
for i in range(len(detalle_compra)):
    if not detalle_compra[i]['cantidad_detalleCompra'] == len(detalle_compra[i]['numeroSerie_item']):
        raise HTTPException(status_code=status.HTTP_409_CONFLICT,
            detail="No concuerda los numeros de serie con la cantidad.")
detalle_compra[i]['compras_id_compra'] = secuencia
detalle_compra[i].pop('pvp_item')
detalle_compra[i].pop('pvd_item')
detalle_compra[i].pop('numeroSerie_item')
detalle_compra[i].pop('descuentoPorcentaje_item')
```

Figura 48-04. Función agregar productos a inventario. (Cisneros & Vaca, 2022)

Estructura Cabecera – Detalle

Este módulo se diferencia de los anteriores, ya que para la generación de una orden de compra se requiere de la implementación de una estructura que se conoce como cabecera detalle. Esta permite organizar de mejor manera toda la información que engloba una orden al separarla en dos secciones importantes.

La parte superior o cabecera contiene la información que permite identificar la compra como el número de factura asociado y el proveedor al que se le fueron comprados los productos. Seguido, continúa el detalle que lista toda la información relacionada con los productos adquiridos, como su precio unitario, su descuento, y la cantidad. Esta sección puede presentar uno o varios detalles asociados a una misma compra, ya que depende del número de elementos adquiridos. Al final, en la parte inferior, se muestran los respectivos valores correspondientes a subtotal, descuento, IVA y total de la compra que fueron calculados con las funciones que se encuentran en la siguiente figura.

A continuación, se muestra el diseño de la interfaz implementada para una orden de compra con la estructura cabecera detalle explicada anteriormente.

← Regresar					
Orden de Compra #6					
Detalles					
Fecha	2022-04-19	Número de Factura	002-001-021048477	Proveedor	Tecnomega
#	PRODUCTO	CANT.	PRECIO UNIT	DESCT.	PRECIO TOTAL
3	Laptop Asus x454	3	\$454.00	%1	\$1,362.00
Subtotal					\$1,362.00
Descuento					\$13.62
IVA %12					\$161.81
Total					\$1,510.19

Figura 49-04. Orden de compra. (Cisneros & Vaca, 2022)

El código completo implementando para este *sprint* se encuentra en los anexos del proyecto.

4.3.2 Revisión del Sprint 3

Con el nuevo incremento del producto entregado correspondiente al tercer *sprint*, se procedió con la revisión de este junto con la presencia del usuario final. En la reunión se presentaron ciertas observaciones las mismas que son resumidas en la siguiente tabla y que serán explicadas más adelante.

¿Qué salió bien en la iteración?	¿Qué no salió bien en la iteración?	¿Qué mejoras vamos a implementar en la próxima iteración?
Listado de proveedores	Validar campo teléfono al registrar un nuevo proveedor	Limitar el campo a 13 dígitos y solo números
Edición de proveedores	Texto del botón para registrar una nueva orden de compra	Cambiar el texto del botón a guardar
Listado de órdenes de compra	Actualizar cantidad de ítems asociados a un producto	Aumentar la cantidad correspondiente al producto en el inventario
Edición de una orden de compra		

Ver detalles de una orden de compra

Tabla 22-04. *Sprint Retrospective 3. (Cisneros & Vaca, 2022)*

En cuanto a la verificación de la administración de proveedores, surgió el único inconveniente en el campo teléfono al momento de registrar un nuevo proveedor. Este campo permitía introducir todo tipo de carácter y no solamente números como debe ser. Además, existía un conflicto al superar los 13 dígitos ya que la base no está diseñada para aceptar más caracteres por lo que ocurría un error y no se permitía la creación del nuevo registro.

Por otro lado, con lo referente a las órdenes de compra también resultaron ser satisfactorias sin embargo el usuario sugirió que, al momento de registrar una nueva orden se cambiara el texto del botón ya que la acción a realizar no estaba clara. Además, se presentó el problema de que la cantidad de ítems asociada a un producto no es cambiada en el inventario según se registra en el detalle de la orden.

4.3.3 Sprint 4

Este nuevo *sprint* del módulo de compras se enfocó solamente en corregir los problemas encontrados en la revisión previa. De tal manera que, se presenta a continuación, la tabla de funcionalidades del módulo con sus estados actualizados las mismas que serán cumplidas al finalizar la iteración.

#	Descripción	Tareas	Estatus
1	Crear una pantalla para la administración de proveedores	Conectarse al microservicio de compras Implementar una tabla con paginación para mostrar registros Implementar formulario para agregar y editar un registro	En proceso

		Implementar una ventana modal para confirmar el cambio de estado de un registro	
		Implementar una tabla con paginación para mostrar registros	
2	Crear una pantalla para la administración de una orden de compra	Implementar formulario para agregar y editar un registro Implementar interfaz para visualizar los detalles del registro	En proceso

Tabla 23-04. *Sprint Backlog 3 Compras-2. (Cisneros & Vaca, 2022)*

Tal como fue requerido, se implementó una validación en el lado del *frontend* para verificar que el campo de teléfono del proveedor solo permitiera ingresar números y que la cantidad de caracteres esté limitada para aceptar únicamente hasta 13 dígitos. En caso de superar el número permitido, se presenta al usuario una notificación de error y no se permite agregar el registro hasta que cumpla con las especificaciones requeridas. Dichas validaciones se observan en la siguiente figura que muestra el mensaje de error correspondiente.

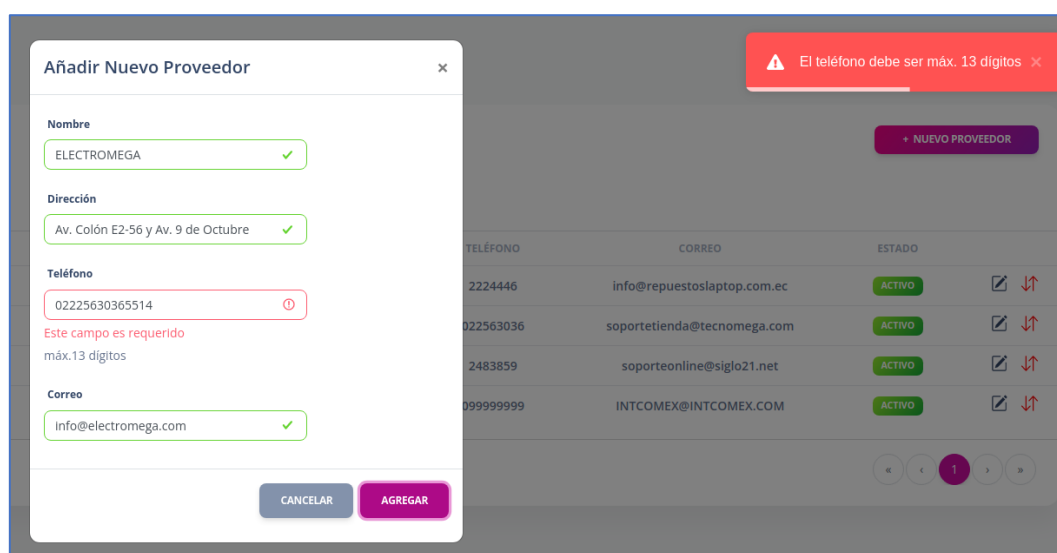


Figura 50-04. *Verificación número de teléfono. (Cisneros & Vaca, 2022)*

Por otro lado, para actualizar la cantidad de un producto en el inventario se realizó una operación en la base de datos que permitió modificar el campo según el número de ítems que

fueron adquiridos. En la figura a continuación se encuentra el código que fue el que permitió satisfacer este requerimiento, simplemente tomando la cantidad actual del producto y sumándole la cantidad ingresada en el detalle de la compra.

```
edit_producto = db.query(modelsDetalleCompras.Productos)\
    .filter(modelsDetalleCompras.Productos.id_producto == producto['productos_id_producto']).first()
edit_producto.cantidad_producto = edit_producto.cantidad_producto + producto['cantidad_detalleCompra']
```

Figura 51-04. Código actualizar cantidad producto. (Cisneros & Vaca, 2022)

4.3.4 Revisión del Sprint 4

Al finalizar el *sprint*, se realizó una nueva reunión con el usuario final para poder verificar el cumplimiento de los requerimientos faltantes. Después de ejecutarse varias pruebas en el módulo de compras del sistema, se concluyó que el producto entregado cumple con lo solicitado y de ser posible, ya se puede continuar con el desarrollo de los siguientes módulos.

4.4 Módulo Ventas y Garantías

El módulo ventas corresponde a una parte fundamental del sistema ya que permite gestionar todas las órdenes de venta emitidas por la empresa a sus clientes, pudiendo estas ser en línea o de forma presencial en el establecimiento del negocio. Por otro lado, el módulo de soporte técnico comprende el manejo de las garantías asociadas a aquellos productos que ya fueron vendidos y por alguna razón requieren de este servicio.

4.4.1 Sprint 4

Este cuarto *sprint* continuó con el desarrollo de los módulos tanto de ventas como de garantías, los mismos que contemplan los requerimientos detallados en la tabla a continuación y que serán cumplidos a lo largo de esta nueva iteración.

#	Descripción	Tareas	Estatus
1	Crear una pantalla para la administración de clientes	Conectarse al microservicio de ventas	Por iniciar
		Implementar una tabla con paginación para mostrar registros	
		Implementar formulario para agregar y editar un registro	
		Implementar una ventana modal para confirmar la eliminación de un registro	
2	Crear una pantalla para la administración de orden de ventas	Implementar una tabla con paginación para mostrar registros	Por iniciar
		Implementar formulario para agregar un registro	
		Implementar interfaz para visualizar los detalles del registro	
3	Crear una pantalla para la administración de garantías	Implementar una tabla con paginación para mostrar registros	Por iniciar
		Implementar formulario para agregar un registro	
		Implementar interfaz para visualizar los detalles del registro	
		Implementar una ventana modal para confirmar la eliminación de un registro	

Tabla 24-04. *Sprint Backlog 4 Ventas-1. (Cisneros & Vaca, 2022)*

De la misma manera que en los *sprint* previos, en la parte del *backend* del sistema se creó un nuevo microservicio con sus *endpoints* respectivos y en el lado del *frontend*, se agregaron las nuevas interfaces que permiten la administración de los clientes, el registro y visualización de ventas y también la gestión de las garantías de los ítems de la empresa. El desarrollo de estos dos módulos fue relativamente similar a los módulos anteriores ya que presentan la misma

estructura que se ha venido trabajando a lo largo de todo el sistema; por lo tanto, todos los componentes detallados anteriormente, en especial en el apartado 4.3.1, fueron reutilizados en esta iteración con ciertas modificaciones.

Entre las principales diferencias, se encuentran en los puntos de entrada listados en la tabla a continuación que fueron implementados en este nuevo *sprint* para que sean consumidos del lado del cliente y permitan manejar la información respectiva de cada módulo.

Microservicio	HTTP Request	Ruta Endpoint	Acción
Ventas	GET	/clientes	Leer Clientes
Ventas	GET	/clientes/viewNombres	Obtener Nombres Clientes
Ventas	GET	/clientes/findByIdentificacion/{identificacion_cliente}	Buscar Cliente por Identificación
Ventas	GET	/clientes/findByNombre/{nombre_cliente}	Buscar Cliente por Nombre
Ventas	GET	/clientes/{id_cliente}	Leer Cliente
Ventas	POST	/clientes	Registrar Cliente
Ventas	PUT	/clientes/{id_cliente}	Editar Cliente
Ventas	DELETE	clientes/{id_cliente}	Eliminar Cliente
Ventas	GET	/ventas	Leer Ventas
Ventas	GET	/ventaDetalle/{id_venta}	Leer Venta con Detalle
Ventas	POST	/ventas	Registrar Venta
Ventas	GET	/garantias	Leer Garantías
Ventas	GET	/garantias/{id_garantia}	Leer Garantía
Ventas	POST	/garantias	Registrar Garantía
Ventas	DELETE	/garantias/{id_garantia}	Eliminar Garantía

Tabla 25-04. Endpoints módulo ventas y garantías. (Cisneros & Vaca, 2022)

Al igual que en el *sprint* 3, para el módulo de ventas se utilizó la misma estructura de cabecera detalle con la única diferencia de que al detallar cada producto adquirido también se muestran los números de serie asociados. Este cambio se visualiza en la siguiente interfaz.

← Regresar

Orden de Venta #14

Detalles

Fecha	2022-05-25	# Comprobante	14	Cliente	Daniela Estefania Cisneros Jácome	Ident. Cliente	1717431942
--------------	------------	----------------------	----	----------------	-----------------------------------	-----------------------	------------

ID	PRODUCTO	NUM SERIE	DESCT.	CANT.	PRECIO UNIT	PRECIO TOTAL
1	Laptop Asus x454	212651LAP003	%0	1	\$600.00	\$600.00
2	Teclado LOGITECH K230 BLACK WIRELESS	21265C202F2	%0	1	\$17.00	\$17.00

Subtotal	\$617.00
Descuento	\$0.00
IVA %12	\$74.04
Total	\$691.04

Figura 52-04. Orden de venta. (Cisneros & Vaca, 2022)

Debido a que para registrar una nueva venta esta debe estar asociada a un cliente, se añadieron distintas formas para facilitar este proceso. La primera consiste en la búsqueda del cliente ya sea por su número de identificación o por su nombre, y al encontrarlo en el sistema se presentan el resto de sus datos cargados tal como se observa en la siguiente figura.

Nueva Orden de Venta

Agregar

Fecha	2022-06-03	Nombre	Jorge David Vaca Hernández	Dirección	América y Portoviejo
Buscar cliente por	<input checked="" type="radio"/> Identificación <input type="radio"/> Nombre <input type="radio"/> Nuevo	Teléfono	32111440	Correo	davidvacah@gmail.com
Identificación	1725041915				

Figura 53-04. Búsqueda identificación cliente en orden de venta. (Cisneros & Vaca, 2022)

La otra opción para asociar un cliente es agregando uno nuevo desde esa misma interfaz. Para este caso se presentan todos los campos para llenar los datos del cliente y registrarlo inmediatamente en el sistema. De ser correcto el registro, saldrán cargados sus datos tal como

en el punto anterior. A continuación se muestra el registro de un nuevo cliente desde la orden de venta.

The screenshot shows a web form titled "Nueva Orden de Venta" with a sub-header "Agregar". The form is divided into several sections:

- Fecha:** 2022-06-03
- Buscar cliente por:** Three radio buttons: "Identificación" (unselected), "Nombre" (unselected), and "Nuevo" (selected).
- Tipo de Identificación:** A dropdown menu with "Identificación" selected and a text input field containing "Identificación" with a note "máx.13 dígitos".
- Nombre:** A text input field.
- Dirección:** A text input field.
- Teléfono:** A text input field.
- Correo:** A text input field.
- REGISTRAR CLIENTE:** A prominent pink button.

Figura 54-04. Registro cliente en orden de venta. (Cisneros & Vaca, 2022)

En cuanto al módulo de garantías, se optó por la interfaz mostrada en la figura a continuación que contiene con dos cartas separadas entre sí para que se pudiera observar claramente tanto los datos del cliente como los del producto ingresado a servicio técnico.

The screenshot shows a window titled "Garantía #1" with a close button (X) in the top right corner. The window is divided into two main sections:

- Datos del Cliente:**
 - Identificación: 1725041915
 - Nombre: Jorge David Vaca Hernández
 - Dirección: América y Portoviejo
 - Teléfono: 32111440
 - Correo: davidvacah@gmail.com
- Información del Producto:**
 - Fecha: 2022-04-23
 - Número de Serie: 50026B768554B8F5
 - Producto: Disco Sólido 240 GB
 - Detalle: Es una garantía de prueba

At the bottom right of the window, there are two buttons: "CANCELAR" (grey) and "OK" (pink).

Figura 55-04. Interfaz Garantías. (Cisneros & Vaca, 2022)

El código completo implementando para este *sprint* se encuentra en los anexos del proyecto.

4.4.2 Revisión del Sprint 4

Al terminar el nuevo *sprint*, se procedió con la revisión de este junto con la presencia del usuario funcional. Dentro de esta se realizaron múltiples pruebas para verificar el funcionamiento de todos los requerimientos y se obtuvieron ciertas observaciones que son resumidas en la tabla a continuación y que serán descritas más adelante.

¿Qué salió bien en la iteración?	¿Qué no salió bien en la iteración?	¿Qué mejoras vamos a implementar en la próxima iteración?
Listado de clientes	Validar campo de identificación al registrar un nuevo cliente	Limitar los campos a 13 dígitos y solo números.
Edición de clientes	Ver detalles de una venta	Agregar en el detalle los números de serie de los productos vendidos en diferentes líneas.
Eliminar un cliente	Cantidad de productos vendidos	Actualizar la cantidad de los productos después de realizar una venta
Registro de una orden de venta	Estado producto vendido	Cambiar a vendido el estado de un producto
Administración de garantías		

Tabla 26-04. *Sprint Restrospective 4. (Cisneros & Vaca, 2022)*

Con respecto a la verificación de la administración de clientes, surgió el único error al momento de registrar uno nuevo, ya que no se encuentra controlado el tipo de dato numérico en el campo de la identificación. Además, en este mismo campo también es necesario limitar el número de caracteres a 13 dígitos tal como fue establecido en la base de datos.

En cuanto a las operaciones relacionadas con las órdenes de venta, estas funcionan de manera correcta simplemente se debe mejorar en la presentación de los números de serie en los productos vendidos ya que se muestran uno a continuación de otro sin poder distinguirse. Otro inconveniente importante ocurre después de registrar una orden de venta al no quitarse del inventario todos los productos contenidos en dicha orden y cambiar su estado a vendido.

Afortunadamente para la administración de garantías no se presentó ninguna observación al momento de realizar las pruebas de validación.

4.4.3 Sprint 5

Este nuevo *sprint* comenzó con las correcciones requeridas de todas las observaciones detalladas anteriormente. Por lo tanto, se presenta en la siguiente tabla el estado actualizado de los requerimientos del módulo.

#	Descripción	Tareas	Estatus
1	Crear una pantalla para la administración de clientes	Conectarse al microservicio de Ventas Implementar una tabla con paginación para mostrar registros Implementar formulario para agregar y editar un registro Implementar una ventana modal para confirmar la eliminación de un registro	En proceso
2	Crear una pantalla para la administración de órdenes de venta	Implementar una tabla con paginación para mostrar registros Implementar formulario para agregar un registro Implementar interfaz para visualizar los detalles del registro	En proceso
3	Crear una pantalla para la administración de garantías	Implementar una tabla con paginación para mostrar registros Implementar formulario para agregar y editar un registro Implementar una ventana modal para confirmar la eliminación de un registro	Completado

Tabla 27-04. *Sprint Backlog 4 Ventas-2. (Cisneros & Vaca, 2022)*

En primer lugar, se controló el campo identificación del cliente para que solo pudiera aceptar caracteres numéricos y solamente hasta 13 dígitos. En caso de superar o incumplir dichos parámetros se presenta la siguiente notificación que se observa en la figura a continuación para alertar al usuario de los cambios que debe realizar.

Ventas / Clientes
Clientes

La identificación debe ser máx 13 dígitos

Regresar

Nuevo Cliente

Añadir Cliente

Tipo de Identificación
 Cédula

Identificación
 1707173754333134
 Este campo es requerido
 máx.13 dígitos

Nombre
 Eddy Patricio Cisneros Tamayo

Dirección
 Jalme Roldós Aguilera

Teléfono
 2483859
 máx.13 dígitos

Correo
 eddycis2002@gmail.com

AGREGAR

Figura 56-04. Validación de identificación del cliente. (Cisneros & Vaca, 2022)

Tal como fue solicitado, se corrigió la presentación de los números de serie de los productos adquiridos para que se presenten en diferentes líneas si es que es mayor a 1 la cantidad de estos. Este cambio se visualiza a continuación.

Regresar

Orden de Venta #11

Detalles

Fecha	2022-05-23	# Comprobante	11	Cliente	Consumidor final	Ident. Cliente	999999999
ID	PRODUCTO	NUM SERIE	DESCT.	CANT.	PRECIO UNIT	PRECIO TOTAL	
1	Laptop Asus x454	212651LAP001 212651LAP002	%0	2	\$600.00	\$1,200.00	
2	TABLET HYUNDAI PLUS 10LB1	HYUNDATA001	%0	1	\$140.00	\$140.00	
						Subtotal	\$1,340.00
						Descuento	\$0.00
						IVA %12	\$0.12
						Total	\$1,500.80

Figura 57-04. Números de serie en orden de venta. (Cisneros & Vaca, 2022)

Después de realizarse una venta satisfactoria y esta es guardada en el sistema, se implementó dentro del microservicio de inventario, una función para poder disminuir la cantidad productos

vendidos de la cantidad que se tenía antes de la venta. De igual manera, se actualizó el estado de dicho producto de disponible a vendido. Con este tipo de transacciones se logra obtener un fácil conteo de los productos disponibles, y llevar un balance de los productos que entran y salen del negocio. El código mostrado en la siguiente figura realiza el proceso explicado previamente.

```
db.query(models.Productos)\
    .filter(models.Productos.id_producto == producto['productos_id_producto'])\
    .update({'cantidad_producto': models.Productos.cantidad_producto - producto['cantidad_detalleVenta']})
```

Figura 58-04. Código quitar cantidad de productos vendidos. (Cisneros & Vaca, 2022)

Durante este mismo *sprint* también se implementó el *dashboard* que se encuentra en la página principal del sistema, el mismo que presenta un reporte mensual del dinero proveniente de los módulos más importantes. Para obtener toda la información necesaria para dichos reportes se crearon nuevos **endpoints** en cada microservicio correspondiente. Estos puntos de entrada son detallados en la tabla a continuación.

Microservicio	HTTP Request	Ruta Endpoint	Acción
Compras	GET	/sumatoria_compra_mensual	Reporte Compras Mensual
Ventas	GET	/ventas/reporte	Últimas Ventas
Ventas	GET	/sumatoria_venta_mensual	Reporte Ventas Mensual
Inventarios	GET	/ítems/dineroEnInventario	Dinero en Inventario

Tabla 28-04. Endpoints para dashboard principal. (Cisneros & Vaca, 2022)

Como se observa en la siguiente imagen, se presentan cuatro datos importantes los mismos que corresponden al dinero mensual obtenido en compras, ventas e inventario junto con el número de nuevos usuarios registrados en el *e-commerce*. Además, en este *dashboard* se incluye un listado de las últimas cinco ventas realizadas.

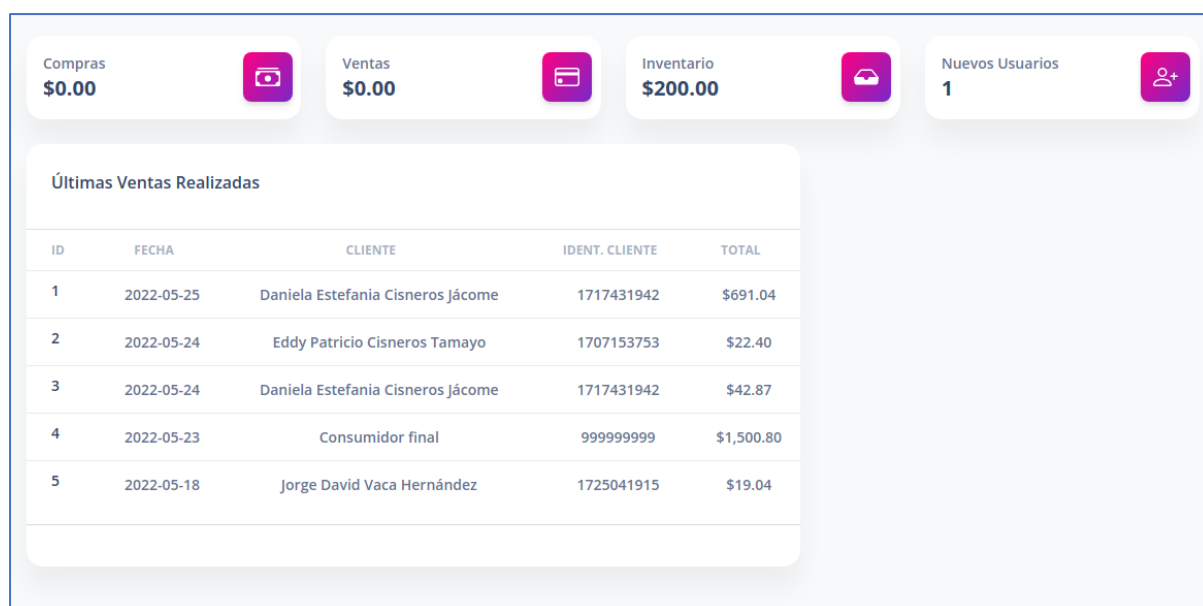


Figura 59-04. Reporte mensual del dashboard. (Cisneros & Vaca, 2022)

El código completo implementando para este *sprint* se encuentra en los anexos del proyecto.

4.4.4 Revisión del Sprint 5

En esta nueva revisión que también contó con la presencia del usuario funcional se verificaron los requerimientos del módulo que quedaron pendientes por cumplirse y se realizaron las respectivas pruebas de estos. Al no presentarse ningún inconveniente, se concluyó que el módulo se encuentra terminado y se puede proceder con el desarrollo de la tienda en línea.

4.5 E-Commerce

El *e-commerce* comprende todos los requerimientos funcionales que el cliente solicita para obtener una experiencia de compra en línea. Entre estos se encuentran fundamentalmente un carrito de compras junto con el catálogo de productos y la generación de órdenes de compra. La implementación de todos estos requerimientos fue dividida en múltiples *sprints* los cuales serán descritos más adelante.

4.5.1 Sprint 5

Este primer *sprint* del *e-commerce* estuvo enfocado en la creación de la página principal de la tienda con todos sus componentes esenciales, es decir, una barra de navegación y varias secciones con información de los servicios que presta la empresa y las marcas con las que trabaja incluyendo también un catálogo de nuevos productos. Además, se implementaron las páginas para el inicio de sesión y registro de los usuarios en caso de ser requerido.

Todos los requerimientos asociados a este *sprint* con su estado respectivo se detallan en la siguiente tabla.

#	Descripción	Tareas	Estatus
1	Crear la página principal	Conectarse al microservicio del e-commerce Crear la barra de navegación principal	Por iniciar
2	Crear una página para el registro de usuarios	Implementar un formulario para el registro de nuevos usuarios	Por iniciar
3	Crear una página para el inicio de sesión de usuarios	Implementar un formulario para el inicio de usuarios	Por iniciar
4	Crear un catálogo de productos	Mostrar los nuevos productos de la empresa en la página principal Mostrar todos los productos de la empresa	Por iniciar
5	Crear un modal para ver la información de un producto	Implementar una ventana modal para visualizar información adicional de un producto	Por iniciar

Tabla 29. *Sprint Backlog 5 Ecommerce-1. (Cisneros & Vaca, 2022)*

Tal como en el sistema de información, en el lado del *backend* se implementó un nuevo microservicio exclusivo para el uso del *e-commerce*. En este se encuentran los nuevos puntos de entrada para inicio y registro de usuarios, los mismos que serán consumidos del lado del *frontend*. Así mismo, se agregaron nuevos *endpoints* en el microservicio de inventario para

obtener la información del módulo y desplegarla en la tienda. En la tabla a continuación se presentan todos los puntos de entrada creados durante esta iteración.

Microservicio	HTTP Request	Ruta Endpoint	Acción
Ecommerce	POST	/login	Inicio sesión
Ecommerce	POST	/usuario/create	Registro usuarios
Inventario	GET	/productos/public	Listar productos
Inventario	GET	/public/productosNuevos	Listar nuevos productos
Inventario	GET	/detalleProducto/venta/findById	Detalle producto

Tabla 30-04. Endpoints para el e-commerce 1. (Cisneros & Vaca, 2022)

Para facilitar el diseño de todas las interfaces de la tienda en línea se utilizaron distintos componentes gratuitos de las librerías Tailwind UI y PrimeVue. De esta manera, se logró implementar la página principal de manera rápida y sencilla con pequeños cambios para adaptarse a las necesidades de la empresa. Como se observa en la siguiente figura, la parte superior de la página principal se compone de una barra de navegación que contiene enlaces a las páginas de catálogo de productos, inicio de sesión o registro de usuarios y dos íconos, uno para buscar un producto en específico y el otro para acceder al carrito de compras.



Figura 60-04. Parte superior de la página principal. (Cisneros & Vaca, 2022)

En esta página también se incluyó un pequeño catálogo para mostrar los nuevos productos ingresados a la empresa usando el diseño de tarjetas explicado en el apartado 3.7.2 correspondiente a la interfaz gráfica. Además, en cada una de estas tarjetas se añadió la funcionalidad de abrir una ventana modal al momento de seleccionar la imagen de un producto. De esta manera, se puede visualizar más detalles de este como su descripción y su marca, al igual que se permite aumentar la cantidad de productos a agregar al carrito. En la siguiente figura, se encuentra el diseño de esta ventana emergente.



Figura 61-04. Ventana modal con detalles de un producto. (Cisneros & Vaca, 2022)

Para el resto de las secciones de la página principal se colocó información sobre los servicios que brinda la empresa a sus clientes y también de las marcas con las que trabaja.

Por otro lado al igual que en el sistema, se trabajaron con tokens de autenticación con la única diferencia de que en esta ocasión se necesitaron dos de ellos debido a los dos tipos de sesiones que se presentan. Se considera una sesión de invitado cuando cualquier persona accede a la tienda en línea y navega en ella sin estar registrado. Caso contrario, la sesión se asocia a las credenciales del usuario iniciado.

Para reconocer el primer tipo de sesión mencionado, se creó un identificador único universal con la ayuda de la librería `Uuidv4` junto con su tiempo de expiración y se lo guardó en el *local storage* del navegador para facilitar su posterior uso. La figura a continuación muestra el código utilizado para el proceso explicado anteriormente.

```
crearSessionID(){
  const id = uuidv4()
  const exp = Date.now() + (3600 * 1000 * 24)
  if(!localStorage.getItem('ID')){
    localStorage.setItem('ID',id)
    localStorage.setItem('ID_exp',exp)
  }
},
```

Figura 62-04. Token sesión invitado. (Cisneros & Vaca, 2022)

En caso de iniciar sesión con las credenciales de un usuario registrado y después de ser verificado en el lado del *backend*, se devuelve un *JWTToken* con el nombre de usuario y su tiempo de validez que también es guardado en el mismo *local storage*. Si se trata de un usuario que desea crearse una cuenta por primera vez en la tienda, debe llenar un pequeño formulario para guardar sus futuros datos de acceso y de ser satisfactorio el registro, se lo redirecciona a la página de iniciar sesión.

El código completo implementando para este *sprint* se encuentra en los anexos del proyecto.

4.5.2 Revisión del Sprint 5

Durante la revisión de este *sprint*, que contó con la presencia de todos los involucrados del proyecto, se revisó el cumplimiento de los requerimientos establecidos para esta iteración. El resumen de las observaciones encontradas se presenta en la siguiente tabla, las cuales serán detalladas más adelante.

¿Qué salió bien en la iteración?	¿Qué no salió bien en la iteración?	¿Qué mejoras vamos a implementar en la próxima iteración?
Barra de navegación	Tiempo de expiración del JWTOKEN	Calcular el tiempo de expiración en milisegundos
Token de sesión de invitado	Limitar cantidad de ítems	Cerrar la sesión después de expirado el tiempo
Inicio y registro de usuarios		Informar al usuario que debe volver a iniciar sesión
Catálogo nuevos productos		Controlar la cantidad de ítems disponibles de un producto al agregar al carrito

Tabla 31-04. *Sprint Retrospective 5. (Cisneros & Vaca, 2022)*

Al momento de revisar el comportamiento del catálogo de los nuevos productos se hicieron ciertas pruebas y en una de ellas se presentó un inconveniente cuando se agregaba un producto al carrito. Esto surgía en la ventana modal de un producto en la que se permite al cliente agregar varios ítems de un mismo producto a carrito de compras, sin embargo, esta cantidad no estaba limitada. Es decir, se puede agregar cualquier número sin tomar en cuenta la disponibilidad de dicho ítem en el inventario.

En cuanto a las pruebas para el inicio y registro de usuarios no se encontraron observaciones ya que su funcionamiento es el esperado. Sin embargo, al verificar el tiempo de expiración cuando un usuario registrado inicia sesión este no era el adecuado. Como se mencionó el JWTOKEN devuelto después de un inicio de sesión correcto contiene en su decodificación el tiempo de validez de este en milisegundos, el mismo que debe ser posteriormente transformado a segundos para poder obtener el tiempo correcto.

4.5.3 Sprint 6

De forma general, esta nueva iteración está centrada en la corrección de las observaciones del anterior iteración y el funcionamiento del carrito de compras de la tienda en línea. Antes de comenzar, se actualizó el estado de los requerimientos previos según su cumplimiento hasta el momento y se agregó aquellos que están a destinados a completarse en el *sprint*.

#	Descripción	Tareas	Estatus
1	Crear un modal para ver la información de un producto	Implementar una ventana modal para visualizar información adicional de un producto	En proceso
2	Implementar un carrito de compras	<p>Conectarse a firebase para utilizar una base de datos en tiempo real</p> <p>Crear un carrito de compras que permita añadir múltiples productos</p> <p>Controlar la cantidad de productos al agregar al carrito según el stock disponible</p> <p>Calcular los precios correspondientes de cada producto según la cantidad agregada y el subtotal de la compra</p>	Por iniciar
3	Implementar un panel de búsqueda	<p>Permitir la búsqueda de productos por su nombre</p> <p>Filtrar los productos por su marca</p>	Por iniciar
4	Crear una página para confirmar la compra	<p>Asociar la compra a un cliente buscándolo por su identificación</p> <p>Registrar un nuevo cliente para guardar la compra</p> <p>Calcular descuentos, IVA y total de la compra</p>	Por iniciar
5	Generar un comprobante de compra	Presentar al cliente un resumen de la compra realizada	Por iniciar
6	Crear una página para ver el historial de compras de un usuario	<p>Presentar un listado de las compras realizadas por el usuario ingresado</p> <p>Mostrar el detalle de cada compra realizada</p>	Por iniciar

Tabla 32-04. *Sprint Backlog 6 Ecommerce-1. (Cisneros & Vaca, 2022)*

Para corregir el cálculo del tiempo de expiración del token de sesión, se implementó la función mostrada en la siguiente figura. Esta se encarga de convertir el tiempo decodificado en una

fecha válida, el mismo que será restado del tiempo actual para poder obtener los segundos en los que continúa vigente el token. Pasado dicho tiempo, aparece una ventana emergente indicando al usuario que es necesario volver a iniciar sesión.

```
async mounted(){
  var decodedToken = getDecoded()
  if (decodedToken != null){
    var date = new Date(decodedToken.exp *1000)
    setTimeout(()=>{
      this.openConfirmation()
    }, date - Date.now())
  }
},
```

Figura 63-04. Verificar tiempo de expiración del token. (Cisneros & Vaca, 2022)

En cuanto a la cantidad de productos disponibles, se limitó por detrás al componente de selección para que permitiera agregar hasta la cantidad de stock que existe de dicho producto en el inventario. Esto se logró agregando un campo más en la consulta de los detalles de un producto en específico a la base. De igual manera, no existe la posibilidad de añadir cualquier otro carácter que no sea un número entero positivo mayor a 1 y menor que la cantidad disponible del producto.

```
async getDetalleProducto(id){
  await this.axios.get(`http://10.147.17.173:5002/detalleProducto/venta/findById/${id}`)
  .then(response => {
    this.nombre = response.data.nombre_producto
    this.detalle = response.data.detalle_producto
    this.precio = response.data.pvp_item
    this.stock = response.data.cantidad_producto
    this.marca = response.data.marca_producto
    this.nombres = response.data.imagen_producto
    for (var i = 0; i < this.nombres.length; i++) {
      var nombre = {
        nombreImagen: this.nombres[i]
      }
      this.imagenes.push(nombre)
    }
  })
  .catch(e => {
    this.$toast.add({severity:'error', summary: 'Error', detail: e.response.data.detail, life: 3000});
  })
},
```

Figura 64-04. Campo stock de un producto. (Cisneros & Vaca, 2022)

Durante esta misma iteración se agregaron nuevos *endpoints* en distintos microservicios del sistema para obtener la información de los módulos correspondientes y utilizarla para la tienda en línea. A continuación, se presentan todos los nuevos puntos de entrada creados durante este *sprint*.

Microservicio	HTTP Request	Ruta Endpoint	Acción
Ventas	POST	/ecommerce/cliente	Registrar cliente
Ventas	GET	/public/clientes/findByIdent	Buscar clientes
Ventas	GET	/public/ventaDetalle/{id_venta}	Leer venta y detalle
Ventas	GET	/public/ventas/findByCliente	Leer compras de un usuario
Inventario	GET	productos/public/findByWord	Buscar productos

Tabla 33-04. Endpoints para el e-commerce 2. (Cisneros & Vaca, 2022)

Para la implementación del carrito de compras de la tienda en línea se hizo uso de una base de datos en tiempo real, la misma que permitiera llevar un registro de cada uno de los productos que los usuarios mantienen en su carrito durante su sesión. Por lo tanto, se ocupó *firebase* de *google* ya que es gratuita y permite lograr el requerimiento explicado previo. En la siguiente figura, se encuentra el código de la configuración que posibilita la conexión a dicha base de datos desde el *e-commerce*.

```
export default {
  apiKey: "AIzaSyCp0wHJTAVtmZKyMz00hq7vYKi9-yA5UE",
  authDomain: "compunex-19db1.firebaseio.com",
  databaseURL: "https://compunex-19db1-default-rtdb.firebaseio.com",
  projectId: "compunex-19db1",
  storageBucket: "compunex-19db1.appspot.com",
  messagingSenderId: "999870272695",
  appId: "1:999870272695:web:afabe5b3d5e2a11b8d6932"
}
```

Figura 65.04. Configuración para la conexión a firebase. (Cisneros & Vaca, 2022)

De tal manera que se pueda asociar los productos que un cliente ha añadido a su carrito de compras, se utilizó el token de sesión de invitado creado en el *sprint* anterior si este es un usuario no registrado; caso contrario se usa el nombre del usuario almacenado en el *JWTToken*. Así, se identifica claramente a qué sesión pertenecen los productos. En esta base de datos, que es únicamente para el carrito de compras, solamente se almacenan el id del producto agregado con su cantidad de ítems respectiva. A continuación, se muestra en la figura cómo se guarda dicha información en la base de datos de *firebase*.



Figura 66-04. Carrito de compras en *firebase*. (Cisneros & Vaca, 2022)

Para lograr guardar toda la información mencionada anteriormente, se utilizaron varios de los métodos propios de *firebase*. Entre estos se encuentran los métodos *push* and *update* que sirven para agregar y actualizar registros a la base, respectivamente. Antes de agregar cualquier producto, se realiza una verificación de existencia de este en el carrito de compras. En caso de no encontrarse dicho producto, se agrega a la base su identificación y la cantidad solicitada del mismo; caso contrario, se actualiza la cantidad de ítems actual a la nueva ingresada. La función implementada para agregar nuevos ítems al carrito de compras se muestra en la figura a continuación.

```
var promesa = this.verificarExistencia(carritoUser, this.id)
promesa.then(
  response => {
    if(response){
      if(cantidad <= stock){
        var articuloRef= ref(db, 'carrito/'+ ident + '/' + response.key)
        update(articuloRef,{
          cantidad : cantidad
        })
        this.terminarCompra()
      }
    }else{
      push(carritoUser,{
        cantidad:cantidad,
        id: this.id
      })
      this.terminarCompra()
    }
  }
)
)
```

Figura 67-04. Agregar ítems al carrito de compras. (Cisneros & Vaca, 2022)

Dentro del carrito de compras se permite al cliente modificar la cantidad de productos agregados y también quitar aquellos que ya no desea comprar más con la ayuda del método *remove* de *firebase*. Automáticamente, el sistema se encarga de calcular los precios según la cantidad ingresada seguido del subtotal correspondiente de toda la compra. A continuación, se muestran las tres funciones implementadas para lograr lo explicado anteriormente.

```
agregarPrecios(){
  this.precios=[]
  for (let i = 0; i < this.productos.length; i++) {
    const precio = this.productos[i].precio
    this.precios.push(precio)
  }
},
calcularSubtotal(){
  this.subtotal = null
  for (var i = 0; i < this.precios.length; i++)
    this.subtotal += parseFloat(this.precios[i])
  this.subtotal = this.subtotal.toFixed(2)
},
quitarElemento(key){
  var ident = ''
  if( getAccessToken() == null)
    ident = localStorage.getItem('ID')
  else
    ident = getUser()
  var carritoRef = ref(db, "carrito/"+ ident + '/' +key)
  remove(carritoRef)
  this.cargarCarrito()
},
```

Figura 68-04. Funciones de los productos del carrito de compras. (Cisneros & Vaca, 2022)

Una vez que el usuario desea confirmar su compra, se procede a verificar si este ha iniciado sesión para continuar con el proceso. En caso de no estarlo, se lo redirecciona a otra página para que pueda acceder a su cuenta debido a que es necesario estar registrado en el sistema para poder generar una orden de compra. Dicha orden puede estar asociada a los datos de un cliente o si se desea, solo como consumidor final. A un mismo usuario de la tienda en línea se le permite vincular sus órdenes de compra a diferentes clientes ya sea buscándolos por su tipo de identificación o registrando uno nuevo en el sistema. Además, se calculan aquí aquellos descuentos posibles que pueda tener la compra así como el IVA generado para obtener el total final a pagar.

```
calcularDescuentoTotal(){
  this.descuento = null
  for (var i = 0; i < this.descuentos.length; i++)
    this.descuento += parseFloat(this.descuentos[i])
  this.descuento = this.descuento.toFixed(2)
},
calcularIva(){
  this.iva = (this.subtotal - this.descuento)
  this.iva = ((this.iva*0.12).toFixed(2))
},
calcularTotal(){
  this.total = null
  this.total = (this.subtotal - this.descuento) + parseFloat(this.iva)
  this.total = parseFloat(this.total).toFixed(2)
},
},
```

Figura 69-04. Funciones para cálculos del carrito de compra. (Cisneros & Vaca, 2022)

En caso de no presentarse ningún impedimento para generar y guardar la orden de compra en el sistema, se muestra al usuario un recibo de confirmación de la orden realizada. En este se encuentra un resumen de todo el pedido con los datos del cliente asociado. Con esto finaliza el proceso de compra y puede regresar a la página principal para ver nuevos productos si lo desea. En la siguiente figura se muestra el código para guardar la orden de compra una vez que se han obtenido todos los datos necesarios.

```
async confirmarCompra(id){
  var params = {
    total_venta: parseFloat(this.total).toFixed(2),
    subtotal_venta: parseFloat(this.subtotal).toFixed(2),
    descuento_venta: parseFloat(this.descuento).toFixed(2),
    clientes_id_cliente: id,
    detalle_venta: this.detalles
  }
  await this.axios.post('http://10.147.17.173:5004/public/ventas',
  params,{ headers:{ Authorization: 'Bearer ' + getAccessToken() }
  }).then((response) => {
    const idVenta= response.data
    this.$toast.add({severity:'success', summary: 'Compra registrada con éxito', life: 3000})
    this.vaciarCarrito()
    setTimeout(()=>{
      this.$router.push({
        path: "/orderSummary/"+idVenta,
      })
    }, 3000)
  })
},
},
```

Figura 70-04. Función para guardar orden de compra. (Cisneros & Vaca, 2022)

De manera adicional, se dio al usuario de la tienda la posibilidad de visualizar un historial de todas las órdenes que ha hecho accediendo a sus compras desde la barra de navegación. Así, este puede llevar un seguimiento claro de sus pedidos pasados y ver el detalle de cada uno de estos. Para esto, se utilizó el componente *DataTable* de *PrimeVue* para mostrar de manera ordenada los registros encontrados de la consulta de compras del usuario de la tienda. A continuación, se visualiza la utilización de dicho componente.

```
<div class="container">
  <DataTable :value="compras" :paginator="true" :rows="10"
    paginatorTemplate="CurrentPageReport FirstPageLink PrevPageLink PageLinks NextPageLink LastPageLink
    :rowsPerPageOptions="[10,20,30]" responsiveLayout="scroll"
    currentPageReportTemplate="Mostrando {first} a {last} registros de {totalRecords}">
    <Column field="numeroComprobante_venta" header="# Comprobante"></Column>
    <Column field="fecha_venta" header="Fecha"></Column>
    <Column field="nombre_cliente" header="Nombre Cliente"></Column>
    <Column field="total_venta" header="Total"></Column>
    <Column field="id_venta" :exportable="false" >
      <template #body="slotProps">
        <Button icon="pi pi-search" class="p-button-rounded p-button-help mr-2"
          @click="viewDetails(slotProps.data.id_venta)" />
      </template>
    </Column>
  </DataTable>
</div>
```

Figura 71-04. Componente para visualizar historial de compras. (Cisneros & Vaca, 2022)

Por último, en este *sprint* se implementó una página adicional para una búsqueda de productos por su nombre. Esto se realizó con la finalidad de facilitar al usuario el encontrar un producto

en específico de manera rápida. Asimismo, se permite filtrar por las marcas del producto asociadas a dicha búsqueda. En la siguiente figura se encuentra la función que permite realizar el proceso explicado anteriormente.

```
async buscarProductos(texto){
  this.vacio=false
  await this.axios.get(`http://10.147.17.173:5002/productos/public/findByWord/${texto}`)
  ).then((response) => {
    if(response.data !=null){
      this.temp = response.data
      for (let i = 0; i < this.temp.length; i++) {
        var contenido ={
          pvp_item : this.temp[i].pvp_item,
          id_producto: this.temp[i].id_producto,
          nombre_producto: this.temp[i].nombre_producto,
          imagen_producto: this.temp[i].imagen_producto,
          descuentoPorcentaje_item: this.temp[i].descuentoPorcentaje_item,
          precioDescuento_producto : precioFinal,
          marca_producto:this.temp[i].marca_producto
        }
        this.productos.push(contenido)
      }
      this.obtenerMarcas(this.productos)
    }
  })
},
```

Figura 72-04. Búsqueda de productos por nombre. (Cisneros & Vaca, 2022)

El código completo implementando para este *sprint* se encuentra en los anexos del proyecto.

4.5.4 Revisión del Sprint 6

Así como en las anteriores iteraciones, se realizó una nueva reunión para la revisión del cumplimiento de los requerimientos del *sprint* actual. Esta reunión contó con la presencia de todo el grupo scrum junto con el usuario funcional para poder verificar el producto entregado. En la tabla a continuación se reflejan las observaciones presentadas a lo largo de la revisión, las mismas que serán explicadas detalladamente más adelante.

¿Qué salió bien en la iteración?	¿Qué no salió bien en la iteración?	¿Qué mejoras vamos a implementar en la próxima iteración?
Añadir productos al carrito	Verificación de productos del carrito al iniciar sesión	Verificar la existencia de productos en la sesión previa del usuario
Actualizar cantidad de productos en el carrito	Búsqueda de clientes para asociar a la compra	En caso de encontrar el cliente presentar sus datos para confirmar
Eliminar productos del carrito	Historial de compras de un usuario	Buscar las órdenes de compras por usuario del e-commerce
Comprobante de compra	Filtrar productos por marca	Filtrar solo las marcas asociadas a la búsqueda del producto
Detalles de una orden de compra pasada		
Búsqueda de productos		

Tabla 34-04. *Sprint Restrospective 6. (Cisneros & Vaca, 2022)*

El carrito de compras presentó un desempeño satisfactorio ya que permite al usuario realizar las acciones de agregar, editar y eliminar un producto sin ningún inconveniente. Sin embargo, cuando un usuario invitado agrega productos a su carrito de compras y luego procede a iniciar sesión a una cuenta que ya posee ítems en el carrito, todos los productos anteriores se transfieren al otro carrito sin comprobar su previa existencia. Es decir, se encuentran ciertos productos duplicados ya que no existe ninguna verificación al hacer este proceso.

Como se mencionó anteriormente en la confirmación de compra, el usuario tiene la posibilidad de asociar dicha compra a un nuevo cliente o a uno ya registrado en el sistema de la empresa. En caso de optar por la segunda opción y de encontrar algún resultado, es necesario mostrar en pantalla los datos del cliente encontrado para saber si es el indicado o se debe realizar una nueva búsqueda.

Otro inconveniente importante ocurrió al momento de obtener el historial de compras ya que dicha búsqueda de órdenes pasadas está asociada a un cliente mas no al usuario como tal. Por

lo tanto, no se presenta el resumen correcto debido a que un mismo usuario puede tener varias compras vinculadas a diferentes clientes.

La última observación encontrada fue en la sección de búsqueda de productos por su nombre. A pesar de que esta búsqueda funcione correctamente, es decir si se presentan todas las coincidencias encontradas al texto ingresado en el buscador, las marcas que se muestran para poder filtrar esa misma búsqueda no corresponden a las coincidencias de la búsqueda sino a todas las registradas en el sistema. Por lo tanto, al momento de filtrar por alguna marca que no pertenezca no se mostrará nunca un resultado.

4.5.5 Sprint 7

Este último *sprint* estuvo centrado solamente en la corrección de todos los inconvenientes detallados en el apartado 4.5.4. los mismos que fueron encontrados en la revisión del producto anterior entregado. Para esto se presenta una nueva lista con los estados de los requerimientos actualizados para esta nueva iteración. En la siguiente tabla se visualizan estos cambios.

#	Descripción	Tareas	Estatus
1	Implementar un carrito de compras	<p>Conectarse a firebase para utilizar una base de datos en tiempo real</p> <p>Crear un carrito de compras que permita añadir múltiples productos</p> <p>Controlar la cantidad de productos al agregar al carrito según el stock disponible</p> <p>Calcular los precios correspondientes de cada producto según la cantidad agregada y el subtotal de la compra</p>	En proceso
2	Implementar un panel de búsqueda	<p>Permitir la búsqueda de productos por su nombre</p> <p>Filtrar los productos por su marca</p>	En proceso
3	Crear una página para confirmar la compra	<p>Asociar la compra a un cliente buscándolo por su identificación</p> <p>Registrar un nuevo cliente para guardar la compra</p>	En proceso

	Calcular descuentos, IVA y total de la compra	
4	<p>Crear una página para ver el historial de compras de un usuario</p> <p>Presentar un listado de las compras realizadas por el usuario ingresado</p> <p>Mostrar el detalle de cada compra realizada</p>	En proceso

Tabla 35-04. *Sprint Backlog 6 Ecommerce-2. (Cisneros & Vaca, 2022)*

En la página de inicio de sesión se implementó una función de verificación del carrito cuando un usuario quiere acceder a la tienda. Esto se realiza básicamente para transferir los productos de una sesión de invitado a la de un usuario registrado y en caso de que este último ya contenga en su carrito productos de sesiones previas se verifica que no sean los mismos y se los agrega. En la siguiente figura se visualiza el código de la función desarrollada para lograr el proceso explicado anteriormente.

```

var carritoNew = ref(db, "carrito/"+ username)
get(carritoNew).then((snapshot) => {
  if(snapshot.exists()){
    snapshot.forEach(function (childSnapshot) {
      var value = childSnapshot.val()
      var producto ={
        key : childSnapshot.key,
        id: value.id,
        cantidad: value.cantidad
      }
      carritoAnterior.push(producto)
    })
    carritoAnterior.forEach(item => {
      arregloCarrito.push(item.id)
    })
    for (let i = 0; i < carritoSesion.length; i++) {
      if (!arregloCarrito.includes(carritoSesion[i].id)){
        push(carritoNew,{
          cantidad: carritoSesion[i].cantidad,
          id: carritoSesion[i].id
        })
      }
    }
  }
})

```

Figura 73-04. Verificación carrito de compras al iniciar sesión. (Cisneros & Vaca, 2022)

En la sección de búsqueda de productos se incorporó el código mostrado en la figura a continuación, el mismo que se encarga de almacenar solamente aquellas marcas de los productos resultantes de un búsqueda realizada. Posteriormente, se presentan al usuario estas

marcas en una lista para que puedan ser seleccionadas y filtrar los resultados asociados a dicha marca escogida.

```
if(marcas.length !=0){
  for (let i = 0; i < this.productos2.length; i++) {
    if(marcas.includes(this.productos2[i].marca_producto)){
      coincidencias.push(this.productos2[i])
    }
  }
  this.productos = coincidencias
}else
this.productos = this.productos2
```

Figura 74-04. Filtrar marcas en búsqueda de productos. (Cisneros & Vaca, 2022)

Tal como fue solicitado, al momento de ingresar los datos de un cliente para asociarlo a la orden de compra y de haberlo encontrado en el sistema, se presentan sus datos confirmados en pantalla. Caso contrario, se notifica al usuario para que este pueda registrarlo como nuevo en el sistema. En la siguiente figura se muestra el comportamiento de la tienda en caso de no encontrar dicho cliente en sus registros.

The screenshot displays the COMPUNEX website interface. At the top, there is a navigation bar with the logo, 'Nuestros Productos', a user profile for 'danicis', and 'Mis compras'. A notification bubble in the top right corner states 'Cliente no encontrado' and 'Registre sus datos'. The main content area is split into two columns. The left column, titled 'Orden de Compra', has radio buttons for 'Con Datos' (selected) and 'Consumidor Final'. Below this is the 'Datos Personales' section with form fields for 'Tipo de Identificación' (set to 'Cédula'), 'Identificación' (1707865437), 'Nombre', 'Dirección', 'Teléfono', and 'Correo'. A blue 'Registrar Cliente' button is at the bottom. The right column, titled 'Resumen', shows a product 'TABLET AMAZON QC 8INC 32GB BLACK' for \$89.00. A summary table below lists: Subtotal: \$89.00, Descuento: -\$0.00, IVA %12: \$10.68, and Total: \$99.68.

Figura 75-04. Registro de nuevo cliente en orden de compra. (Cisneros & Vaca, 2022)

Finalmente, para el historial de compras se modificó la consulta hacia la base de datos para poder obtener las órdenes de un usuario en específico del *e-commerce*. Por lo tanto, ahora se

presenta un listado de estas órdenes sin importar el nombre del cliente asociado. A continuación, se visualiza la función que permite obtener dicha información desde el *backend* y mandarla al *frontend* para que sea consumida y renderizada.

```
def listar_ventas_por_usuario(db: Session, id_usuario: int):  
    ventas = db.query(models.Ventas.id_venta,  
                      models.Ventas.numeroComprobante_venta,  
                      models.Ventas.total_venta,  
                      models.Ventas.fecha_venta,  
                      modelsClientes.Clientes.nombre_cliente  
                      )\  
    .join(modelsClientes.Clientes)\  
    .filter(models.Ventas.usuario_venta == id_usuario)\  
    .order_by(models.Ventas.numeroComprobante_venta.desc())\  
    .all()  
    if not ventas:  
        return None  
    return ventas
```

Figura 76-04. Listado de compras de un usuario e-commerce. (Cisneros & Vaca, 2022)

4.5.6 Revisión del Sprint 7

Esta última reunión consistió en la verificación del cumplimiento de los requerimientos que quedaron pendientes de la iteración previa. Todo el equipo scrum junto con el usuario final, realizaron las pruebas correspondientes en el producto entrega en búsqueda de cualquier inconveniente. Al no presentarse ninguna observación, se llegó a la conclusión que la tienda en línea se encuentra acabada y cumple con todos los requerimientos solicitados.

5. CAPÍTULO 5: Conclusiones y Recomendaciones

En el siguiente capítulo se detallan las conclusiones y recomendaciones obtenidas tras cumplir el alcance propuesto para el presente trabajo de titulación.

5.1 Conclusiones

Se abordó satisfactoriamente el análisis, diseño y desarrollo de un sistema de información web para la automatización de los procesos de compras, ventas e inventarios de la empresa Compunex incluyendo la tienda en línea.

El uso de la metodología ágil Scrum en el desarrollo del trabajo brindó un tiempo de respuesta más rápido al esperado ya que gracias a sus continuas reuniones con los usuarios especializados se pueden satisfacer los requerimientos de una manera más eficaz y eficiente.

La arquitectura de microservicios resultó de gran utilidad al desarrollar tanto el sistema de información como la tienda en línea, debido a que por su independencia entre microservicios se redujo el número de fallos que afecten a la totalidad del sistema.

Es de suma importancia conocer y comprender los procesos de la empresa para poder implementar una arquitectura de microservicios que abarque correctamente cada uno de ellos siempre buscando la optimización de recursos y una buena gestión de estos.

Por seguridad y al tratarse de un sistema de información de tres capas, se bloqueó toda conexión o acceso a cada servidor con excepción de dos de ellas. La conexión del servidor *backend* puede aceptar solicitudes *REST* del servidor *frontend*, y el servidor de base datos admite solo consultas provenientes del servidor *backend*.

En algunas ocasiones, al trabajar con dos equipos que se encuentran conectados mediante internet, las peticiones y solicitudes toman bastante tiempo en realizarse debido a las

intermitencias que cuenta dicho servicio, afectando directamente al enlace de conexión entre el servidor de *frontend* y *backend*. Mientras que, al estar conectadas en una misma red local no se presenta ninguna demora.

Durante el desarrollo del sistema de información se presentaron ciertos inconvenientes en el manejo de los usuarios con las tablas propias de *PostgreSQL* lo que limitaba el enfoque que solicitaba la empresa Compunex. Por lo tanto, se optó por la implementación de un propio módulo de autenticación y autorización que satisface los requerimientos solicitados.

La generación de un token firmado resulta ser una buena práctica de seguridad sobre todo para el acceso a la información debido a que este puede contener datos importantes de un usuario, los mismos que son encriptados por un algoritmo y validados por una firma o certificado generado por el servidor de autorización.

El desarrollo de un sistema gestionado mediante un token de seguridad permite la concurrencia de múltiples usuarios conectados a la misma vez. Pudiendo así, cada uno realizar transacciones independientes gracias a la información encriptada que se almacena en el token.

El sistema *e-commerce* debe ampliarse enfocándose más en la experiencia del usuario mediante prácticas de marketing para así obtener una mejor inmersión del cliente dentro del sitio web con la finalidad de incrementar el número de ventas realizadas mediante esta plataforma.

5.2 Recomendaciones

Antes de implementar una arquitectura de microservicios en un desarrollo, analizar las ventajas y problemas al trabajar bajo esa arquitectura ya que puede ser complicado de gestionar al tener un gran número de servicios web levantados.

En un desarrollo con servidores independientes para cada capa, se debe mantener a todos estos servidores; *frontend*, *backend* y de base de datos en una misma red con una velocidad local óptima para evitar latencias y desconexiones.

Tomar en cuenta el ancho de banda al momento de usar el sistema de información, y de ser posible implementar una aplicación que permita el balanceo de carga cuando se trabaja con muchos usuarios.

Al ser un sistema concurrente con una cantidad de usuarios ilimitados, se aconseja realizar una investigación para conocer el número de usuarios máximos conectados que soporta la plataforma sin dejar de prestar un servicio óptimo.

Investigar métodos de cifrado más fuertes a los utilizados y proporcionados por *FastAPI* con el fin de contar con una mayor seguridad en el acceso a datos sensibles que se encuentran en el token de acceso.

Utilizar y mejorar el módulo de autenticación y autorización implementado para el manejo de los usuarios del sistema de información en futuros trabajos de titulación.

Con el fin de mejorar, optimizar y aumentar las ventas de la tienda en línea, mostrar productos similares o relacionados con los que el usuario agregó a su carrito de compras, basándose en la minería de datos ya obtenidos por productos anteriormente buscados o comprados.

Para complementar el funcionamiento del *e-commerce*, es necesario implementar una pasarela o botón de pagos que permita completar el proceso de compra, así como también un módulo que posibilite la conexión al Servicio de Rentas Internas para la facturación.

Bibliografía

Abrego, D., Sánchez, Y., & Medina, J. (2016). Influencia de los sistemas de información en los resultados organizacionales. Mexico. doi:<http://dx.doi.org/10.1016/j.cya.2016.07.005>

American Marketing Association. (2021). *Definitions of Marketing*. Obtenido de AMA:

<https://www.ama.org/the-definition-of-marketing-what-is-marketing/>

Apeiron Software. (13 de Julio de 2019). *SCRUM- ROLES, ARTIFACTS AND CEREMONIES*.

Obtenido de <https://www.apeironsoftware.com/scrum-roles-artifacts-and-ceremonies/>

Arenal Laza, C. (2017). *Técnicas de venta*. España: Tutor Formación. Obtenido de

<https://elibro.puce.elogim.com/es/ereader/puce/44231?page=4>

Basantes, A., Gallegos, M., Guevara, C., Jácome, A., Posso, Á., Quiña, J., & Vaca, C. (2016).

Comercio Electrónico. Ibarra: Universidad Técnica del Norte.

Bertone, R., & Thomas, P. (2017). *Introducción a las bases de datos: Fundamentos y Diseño*. Buenos

Aires: Pearson Education. Obtenido de

<https://elibro.puce.elogim.com/es/ereader/puce/78343?page=1>

Cisneros, D., & Vaca, J. (2022). *Análisis, diseño y desarrollo de un sistema de información web para automatizar los procesos de compras, inventarios y ventas (e-commerce)*. Quito: PUCE.

Django. (2021). *Django overview*. Obtenido de <https://www.djangoproject.com/start/overview/>

Esper, T., & Waller, M. (2017). *Administración de inventarios*. Ciudad de México: Pearson

Education. Obtenido de <https://elibro.puce.elogim.com/es/ereader/puce/38086?page=15>

Espinoza, C. (2018). *Análisis del proceso de compras para diseñar una propuesta de indicadores de gestión que permita mejorar los procesos del área de compras*. [Tesis de Grado, Universidad

Católica de Santiago de Guayaquil]. Repositorio Institucional, Guayaquil. Obtenido de

<http://repositorio.ucsg.edu.ec/handle/3317/11016>

FastAPI. (2022). *FastAPI*. Obtenido de <https://fastapi.tiangolo.com/>

López, D., & Maya, E. (2017). *Arquitectura de Software basada en Microservicios para desarrollo de aplicaciones web*. Séptima Conferencia de Directores de Tecnología de Información, Ecuador. Obtenido de <https://dspace.redclara.net/bitstream/10786/1277/1/93%20Arquitectura%20de%20Software%20basada%20en%20Microservicios%20para%20Desarrollo%20de%20Aplicaciones%20Web.pdf>

Márquez, N. (29 de Enero de 2020). Frameworks de desarrollo web back-end. *Moleqla*(36), 11-13. Obtenido de <https://www.upo.es/cms1/export/sites/upo/moleqla/documentos/Numero36/Numero-36.pdf>

Martínez, D. (2019). Concepto y origen de base de datos. En D. Martínez, *Manual de base de datos* (págs. 14-15). Santiago de los Caballeros, República Dominicana: Universidad Abierta para Adultos. Obtenido de <https://elibro.puce.elogim.com/es/ereader/puce/175897>

Meltzer, J. (Agosto de 2018). A digital trade policy for Latin America and the Caribbean. *Inter-American Development Bank*.

Navarro, A., Fernández, J., & Morales, J. (2013). Revisión de metodologías ágiles para el desarrollo de software. *PROSPECTIVA*, 11(2), págs. 30-39. Obtenido de <https://www.redalyc.org/pdf/4962/496250736004.pdf>

Nuxt.js. (2021). *NUXTJS*. Obtenido de <https://nuxtjs.org/>

Oracle. (2021). *¿Qué es una base de datos?* Obtenido de Oracle Database: <https://www.oracle.com/ar/database/what-is-database/#link5>

Pesántez, A., Romero, J., & González, M. (Enero-Abril de 2020). Comercio electrónico B2B como estrategia competitiva en el comercio internacional: Desafíos para Ecuador. *INNOVA reseach journal*, 5(1), págs. 72-93. doi:<https://doi.org/10.33890/innova.v5.n1.2020.1166>

Pulido, E., Escobar, Ó., & Núñez, J. (2019). *Base de datos*. Ciudad de México: Grupo Editorial Patria.

Obtenido de <https://elibro.puce.elogim.com/es/lc/puce/titulos/121283>

Rolando, J., Zea, M., Contento, M., & García, F. (2017). Estado del arte: Metodologías de desarrollo en aplicaciones web. *3c Tecnología: glosas de innovación aplicadas a la pyme*, 6(3), págs. 54-71. Obtenido de <https://dialnet.unirioja.es/servlet/articulo?codigo=6143045>

Schwaber, K., & Sutherland, J. (Noviembre de 2017). La guía definitiva de Scrum: las reglas de juego. Obtenido de <https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-SouthAmerican.pdf>

Tiam Kok, L. (2020). *Hands on Nuxt.js Web Development*. Birmingham: Packt Publishing.

Valdivia, J. (2015). *Comercialización de productos y servicios en pequeños negocios o microempresas*. Málaga: IC Editorial.

Vázquez del Cerro, P. (2020). Frameworks y lenguajes de desarrollo para frontend. *MoleQla*, 36, 13-15. Obtenido de <https://www.upo.es/cms1/export/sites/upo/moleqla/documentos/Numero36/Numero-36.pdf>

Zea, M., Molina, J., & Redrován, F. (2017). Administración de base de datos con PostgreSQL. 3Ciencias. Obtenido de https://play.google.com/store/books/details?id=5-mkDgAAQBAJ&rdid=book-5-mkDgAAQBAJ&rdot=1&source=gbs_vpt_read&pcampaignid=books_booksearch_viewport

Zeballos, R. (Noviembre de 2012). Aplicando SCRUM. *Revista Investigación y Tecnología*. Obtenido de http://www.revistasbolivianas.org.bo/scielo.php?pid=S2306-05222012000100014&script=sci_arttext

Zumba, J. (2018). Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software. *INNOVA Research Journal*, 3(10), págs. 20-33. doi:<https://dialnet.unirioja.es/descarga/articulo/6777227.pdf>

Anexos

Anexo 1. Diagramas de Caso de Uso (Cisneros & Vaca, 2022)

Anexo 2. Modelo Base de Datos (Cisneros & Vaca, 2022)

Anexo 3. Wireframes E-commerce (Cisneros & Vaca, 2022)

Anexo 4. Wireframes Sistema (Cisneros & Vaca, 2022)

Anexo 5. Historias de Usuario Scrum (Cisneros & Vaca, 2022)

Anexo 6. Product Backlog Scrum (Cisneros & Vaca, 2022)

Anexo 7. Sprint Backlog Scrum (Cisneros & Vaca, 2022)

Anexo 8. Sprint Retrospective Scrum (Cisneros & Vaca, 2022)

Listado de Figuras

Figura 1-01. Estructura de un sprint (Apeiron Software, 2019).....	11
Figura 2-01. Patrón básico de la arquitectura de microservicios. (López & Maya, 2017)	13
Figura 3-02. Organigrama de la empresa Compunex. (Cisneros & Vaca, 2022)	31
Figura 4-02. Proceso de venta de la empresa Compunex. (Cisneros & Vaca, 2022)	32
Figura 5-03. Diseño del módulo de compras. (Cisneros & Vaca, 2022)	40
Figura 6-03. Diseño del módulo ventas. (Cisneros & Vaca, 2022)	41
Figura 7-03. Diseño del módulo inventarios. (Cisneros & Vaca, 2022).....	42
Figura 8-03. Diseño módulo garantías. (Cisneros & Vaca, 2022).....	43
Figura 9-03. Diseño de gestión del sistema. (Cisneros & Vaca, 2022)	44
Figura 10-03. Diagrama de caso de uso general. (Cisneros & Vaca, 2022)	45
Figura 11-03. Caso de uso ventas. (Cisneros & Vaca, 2022)	46
Figura 12-03. Caso de uso clientes. (Cisneros & Vaca, 2022)	47
Figura 13-03. Caso de uso submódulo proveedores. (Cisneros & Vaca, 2022)	47
Figura 14-03. Interfaz Inicio Sesión. (Cisneros & Vaca, 2022)	49
Figura 15-03. Interfaz página principal con Dashboard. (Cisneros & Vaca, 2022).....	50
Figura 16-03. Interfaz visualización de registros. (Cisneros & Vaca, 2022).....	51
Figura 17-03. Interfaz para añadir un registro. (Cisneros & Vaca, 2022)	52
Figura 18-03. Interfaz para editar un registro. (Cisneros & Vaca, 2022)	53
Figura 19-03. Interfaz modal confirmación de eliminación. (Cisneros & Vaca, 2022).....	53
Figura 20-03. Interfaz registro de usuarios E-commerce. (Cisneros & Vaca, 2022).....	54
Figura 21-03. Interfaz catálogo en forma de grid. (Cisneros & Vaca, 2022)	55
Figura 22-03. Interfaz catálogo en forma de lista. (Cisneros & Vaca, 2022)	56
Figura 23-03. Interfaz carrito de compras. (Cisneros & Vaca, 2022).....	57
Figura 24-03. Interfaz confirmación de compra. (Cisneros & Vaca, 2022)	58
Figura 25-03. Interfaz historial de compras. (Cisneros & Vaca, 2022).....	58

Figura 26-03. Interfaz detalle orden de compra. (Cisneros & Vaca, 2022)	59
Figura 27-04. Sistema de virtualización (Cisneros & Vaca, 2022)	63
Figura 28-04. Estructura del microservicio. (Cisneros & Vaca, 2022).....	65
Figura 29-04. Rutas del módulo. (Cisneros & Vaca, 2022).....	68
Figura 30-04. Componentes. (Cisneros & Vaca, 2022).....	69
Figura 31-04. Propiedades de un componente. (Cisneros & Vaca, 2022)	69
Figura 32-04. Activos del proyecto. (Cisneros & Vaca, 2022).....	70
Figura 33-04. Estructura del JW Token. (Cisneros & Vaca, 2022)	71
Figura 34-04. Almacenamiento local. (Cisneros & Vaca, 2022).....	71
Figura 35-04. Funciones de Autenticación. (Cisneros & Vaca, 2022).....	72
Figura 36-04. Middleware de autenticación. (Cisneros & Vaca, 2022)	73
Figura 37-04. Verificación expiración de Token JW. (Cisneros & Vaca, 2022).....	73
Figura 38-04. Ejemplo solicitud con Axios. (Cisneros & Vaca, 2022)	74
Figura 39-04. Usuarios activos e inactivos del sistema. (Cisneros & Vaca, 2022)	77
Figura 40-04. Permisos de un rol. (Cisneros & Vaca, 2022)	78
Figura 41-04. Imágenes de un producto. (Cisneros & Vaca, 2022).....	81
Figura 42-04. Función para redimensionar imágenes. (Cisneros & Vaca, 2022).....	81
Figura 43-04. Función para guardar archivos. (Cisneros & Vaca, 2022)	82
Figura 44-04. Listado de productos. (Cisneros & Vaca, 2022)	84
Figura 45-04. Componente Vueper slides. (Cisneros & Vaca, 2022).....	85
Figura 46-04. Carrusel de imágenes. (Cisneros & Vaca, 2022)	85
Figura 47-04. Código para listar compras con detalle. (Cisneros & Vaca, 2022)	88
Figura 48-04. Función agregar productos a inventario. (Cisneros & Vaca, 2022)	89
Figura 49-04. Orden de compra. (Cisneros & Vaca, 2022)	90
Figura 50-04. Verificación número de teléfono. (Cisneros & Vaca, 2022).....	92
Figura 51-04. Código actualizar cantidad producto. (Cisneros & Vaca, 2022).....	93

Figura 52-04. Orden de venta. (Cisneros & Vaca, 2022)	96
Figura 53-04. Búsqueda identificación cliente en orden de venta. (Cisneros & Vaca, 2022)	96
Figura 54-04. Registro cliente en orden de venta. (Cisneros & Vaca, 2022)	97
Figura 55-04. Interfaz Garantías. (Cisneros & Vaca, 2022)	97
Figura 56-04. Validación de identificación del cliente. (Cisneros & Vaca, 2022)	100
Figura 57-04. Números de serie en orden de venta. (Cisneros & Vaca, 2022).....	100
Figura 58-04. Código quitar cantidad de productos vendidos. (Cisneros & Vaca, 2022)	101
Figura 59-04. Reporte mensual del dashboard. (Cisneros & Vaca, 2022).....	102
Figura 60-04. Parte superior de la página principal. (Cisneros & Vaca, 2022)	104
Figura 61-04. Ventana modal con detalles de un producto. (Cisneros & Vaca, 2022).....	105
Figura 62-04. Token sesión invitado. (Cisneros & Vaca, 2022).....	106
Figura 63-04. Verificar tiempo de expiración del token. (Cisneros & Vaca, 2022)	109
Figura 64-04. Campo stock de un producto. (Cisneros & Vaca, 2022).....	109
Figura 65-04. Configuración para la conexión a firebase. (Cisneros & Vaca, 2022)	110
Figura 66-04. Carrito de compras en firebase. (Cisneros & Vaca, 2022).....	111
Figura 67-04. Agregar ítems al carrito de compras. (Cisneros & Vaca, 2022).....	112
Figura 68-04. Funciones de los productos del carrito de compras. (Cisneros & Vaca, 2022).....	112
Figura 69-04. Funciones para cálculos del carrito de compra. (Cisneros & Vaca, 2022).....	113
Figura 70-04. Función para guardar orden de compra. (Cisneros & Vaca, 2022).....	114
Figura 71-04. Componente para visualizar historial de compras. (Cisneros & Vaca, 2022).....	114
Figura 72-04. Búsqueda de productos por nombre. (Cisneros & Vaca, 2022)	115
Figura 73-04. Verificación carrito de compras al iniciar sesión. (Cisneros & Vaca, 2022)	118
Figura 74-04. Filtrar marcas en búsqueda de productos. (Cisneros & Vaca, 2022)	119
Figura 75-04. Registro de nuevo cliente en orden de compra. (Cisneros & Vaca, 2022).....	119
Figura 76-04. Listado de compras de un usuario e-commerce. (Cisneros & Vaca, 2022)	120

Listado de Tablas

Tabla 1-02. Estado financiero de Compunex. (Cisneros & Vaca, 2022)	28
Tabla 2-03. Requerimientos funcionales módulo compras. (Cisneros & Vaca, 2022)	35
Tabla 3-03. Requerimientos funcionales módulo ventas. (Cisneros & Vaca, 2022)	36
Tabla 4-03. Requerimientos funcionales módulo inventarios. (Cisneros & Vaca, 2022)	36
Tabla 5-03. Requerimientos funcionales módulo garantías. (Cisneros & Vaca, 2022)	37
Tabla 6-03. Requerimientos funcionales E-commerce. (Cisneros & Vaca, 2022)	37
Tabla 7-03. Requerimientos no funcionales. (Cisneros & Vaca, 2022)	38
Tabla 8-03. Requerimientos del sistema. (Cisneros & Vaca, 2022)	39
Tabla 9-04. Roles de Scrum. (Cisneros & Vaca, 2022)	60
Tabla 10-04. Historias de Usuario. (Cisneros & Vaca, 2022)	61
Tabla 11-04. Product Backlog. (Cisneros & Vaca, 2022)	62
Tabla 12-04. Sprint Backlog 1 Administración-1. (Cisneros & Vaca, 2022)	64
Tabla 13-04. Endpoints para el primer sprint. (Cisneros & Vaca, 2022)	68
Tabla 14-04. Sprint Restrospective 1. (Cisneros & Vaca, 2022)	75
Tabla 15-04. Sprint Backlog 1 Administración-2. (Cisneros & Vaca, 2022)	77
Tabla 16-04. Sprint Backlog 2 Inventario-1. (Cisneros & Vaca, 2022)	79
Tabla 17-04. Endpoints módulo inventario. (Cisneros & Vaca, 2022)	80
Tabla 18-04. Sprint Restrospective 2. (Cisneros & Vaca, 2022)	82
Tabla 19-04. Sprint Backlog 2 Inventario-2. (Cisneros & Vaca, 2022)	83
Tabla 20-04. Sprint Backlog 3 Compras-1. (Cisneros & Vaca, 2022)	87
Tabla 21-04. Endpoints módulo compras. (Cisneros & Vaca, 2022)	87
Tabla 22-04. Sprint Retrospective 3. (Cisneros & Vaca, 2022)	91
Tabla 23-04. Sprint Backlog 3 Compras-2. (Cisneros & Vaca, 2022)	92
Tabla 24-04. Sprint Backlog 4 Ventas-1. (Cisneros & Vaca, 2022)	94
Tabla 25-04. Endpoints módulo ventas y garantías. (Cisneros & Vaca, 2022)	95

Tabla 26-04. Sprint Restrospective 4. (Cisneros & Vaca, 2022)	98
Tabla 27-04. Sprint Backlog 4 Ventas-2. (Cisneros & Vaca, 2022)	99
Tabla 28-04. Endpoints para dashboard principal. (Cisneros & Vaca, 2022).....	101
Tabla 29. Sprint Backlog 5 Ecommerce-1. (Cisneros & Vaca, 2022)	103
Tabla 30-04. Endpoints para el e-commerce 1. (Cisneros & Vaca, 2022).....	104
Tabla 31-04. Sprint Retrospective 5. (Cisneros & Vaca, 2022).....	107
Tabla 32-04. Sprint Backlog 6 Ecommerce-1. (Cisneros & Vaca, 2022)	108
Tabla 33-04. Endpoints para el e-commerce 2. (Cisneros & Vaca, 2022).....	110
Tabla 34-04. Sprint Restrospective 6. (Cisneros & Vaca, 2022)	116
Tabla 35-04. Sprint Backlog 6 Ecommerce-2. (Cisneros & Vaca, 2022).....	118