



**Facultad de Ciencias y Tecnologías Computacionales**

# **Sistema de gestión de información web para los almacenes de suministros médicos distribuidos por EMSUME**

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

**Autor(es):** Arianna Pérez Rodríguez

**Tutor(es):** Ing. Leonel Eduardo Perdomo Roig

**Co-tutor:** Lic. Liliannes Caridad Matamoros Benítez

La Habana, noviembre de 2022

Año 63 de la Revolución

## **DECLARACIÓN DE AUTORÍA**

Declaro por este medio que yo Arianna Pérez Rodríguez con carnet de identidad: 99121206835, soy la única autora del trabajo titulado “**Sistema de gestión web para los almacenes de suministros médicos distribuidos por EMSUME**” y autorizo a la Universidad de las Ciencias Informáticas de los derechos patrimoniales de la investigación con, carácter exclusivo.

Para que así conste, firmo la presente declaración de autoría a los \_\_\_\_ días del \_\_\_\_ mes del año \_\_\_\_.

### **Autora:**

\_\_\_\_\_  
Arianna Pérez Rodríguez

### **Tutores:**

\_\_\_\_\_  
Ing. Leonel Eduardo Perdomo Roig

\_\_\_\_\_  
Lic. Liliannes Caridad Matamoros Benítez

## **DATOS DE CONTACTO**

**Leonel Eduardo Perdomo Roig:** graduado en Ingeniería en Ciencias Informáticas (UCI), 2020. Especialista en Ciencias Informáticas del Centro de Análisis y Representación de Datos. Profesor de Ingeniería de Software del Departamento de Informática de la Facultad de Ciencias y Tecnologías Computacionales. Experiencia Laboral: 2 años, correo electrónico leo@uci.cu.

**Liliannes Caridad Matamoros Benítez:** graduada de Licenciatura en Psicología en la Universidad de La Habana, 2007. Profesora del departamento de informática en la facultad de Ciencias y Tecnologías Computacionales. Experiencia Laboral: 15 años. Líneas de investigación: desarrollo de competencias emocionales en equipos de desarrollo de software. Grupo de investigación: Gestión de Proyectos, correo electrónico: lmatamorosb@uci.cu.

## AGRADECIMIENTOS

*A mis padres por su apoyo y amor incondicional, su dedicación, por ser el motor impulsor de mi vida y ayudarme a cumplir mis metas.*

*A mis hermanas Yadira y Mariela, por su amor, sus consejos y por creer en mí.*

*A mis amigos , en especial a Michel, Brian, Káterin, Elizabeth, Álvaro, Thalia, Marie y Javier por su apoyo en los buenos y malos momentos.*

*A mis tutores, por ser tan entregados en su labor de guía durante estos meses de trabajo y brindarme su apoyo.*

## DEDICATORIA

*Dedicado a mi familia, en especial mis padres, mis hermanas, mi abuela Amparo, mi abuelo Alfredo y mi tío Enrique.*

## RESUMEN

Los Sistemas de Gestión de la Información, cuyo desarrollo está marcado por el avance de las Tecnologías de la Información y las Comunicaciones (TIC), han propiciado la optimización de los procesos en las organizaciones, lo que contribuye al control efectivo de su información. En el área comercial de la empresa EMSUME, se lleva a cabo la elaboración de informes de inventario de la entidad. Este proceso es realizado de forma manual y sistemática, provocando la falta de inmediatez requerida a la hora de dar un reporte y la dificultad en el control de los productos que entran, existen y salen de los almacenes. En la presente investigación se desarrolla un Sistema de Gestión de la Información que permite controlar adecuadamente la información relacionada con este proceso. Con el fin de alcanzar los objetivos trazados se analizan los principales conceptos asociados a la obtención de inventarios y control de los almacenes, se realiza una investigación a los sistemas similares existentes, tanto en el ámbito nacional como internacional. Se utilizaron como lenguaje de desarrollo Python y JavaScript, se seleccionó la metodología Scrum y como herramientas de base de datos SQL Server 2017 y SQLite 3. Se realizan un conjunto de pruebas para garantizar el correcto funcionamiento de las funcionalidades propuestas, teniendo en cuenta también que la aplicación cumple con los requerimientos del cliente. Como resultado se obtuvo un sistema para gestionar toda la información referente al control de los almacenes, que permite además la generación de reportes de faltantes y ayuda a la toma de decisiones.

**Palabras clave:** gestión, información, inventario, suministros médicos

**ABSTRACT**

*Information Management Systems, whose development is marked by the progress of Information and Communication Technologies (ICT), have led to the optimization of processes in organizations, which contributes to the effective control of their information. In the commercial area of the company EMSUME, the preparation of inventory reports of the entity is carried out. This process is carried out manually and systematically, causing the lack of immediacy required at the time of giving a report and the difficulty in controlling the products that enter, exist and leave the warehouses. In the present research, an Information Management System is developed that allows to adequately control the information related to this process. In order to achieve the objectives outlined, the main concepts associated with inventory and warehouse control are analyzed, and an investigation is made of similar existing systems, both nationally and internationally. Python and JavaScript were used as development language, the Scrum methodology was selected and SQL Server 2017 and SQLite 3 were used as database tools. A set of tests were performed to ensure the correct operation of the proposed functionalities, also taking into account that the application meets the customer's requirements. As a result, a system was obtained to manage all the information related to the control of the warehouses, which also allows the generation of reports of shortages and helps in decision making.*

**Key words:** *management, information, inventory, medical supplies.*

**TABLA DE CONTENIDOS**

INTRODUCCIÓN.....	1
CAPÍTULO 1: Fundamentos de los sistemas de gestión de información.....	5
1.1 Conceptos Fundamentales.....	5
1.2 Sistemas de Gestión de Información.....	6
1.3 Análisis de los sistemas existentes.....	7
1.3.1 Sistemas ERP.....	7
1.3.2 Soluciones informáticas Internacionales.....	7
1.3.3 Soluciones Nacionales.....	8
1.3.4 Comparación entre las soluciones existentes.....	9
1.3.5 Valoración crítica de los sistemas existentes.....	9
1.4 Lenguajes, Metodologías y Herramientas.....	10
1.4.1 ¿Qué es una metodología de desarrollo de software? .....	10
1.4.1.1 ¿Qué es una metodología de desarrollo de software? .....	10
1.4.1.2 Fundamentación del uso de la Metodología SCRUM.....	11
1.4.2 Lenguaje de modelado.....	12
1.4.3 Herramienta de modelado.....	12
1.4.4 Lenguajes y Tecnologías.....	13
Conclusiones parciales del capítulo .....	18
CAPÍTULO 2: Diseño del Sistema de gestión de información para los almacenes de suministros médicos de EMSUME.....	19
2.1 Modelado del dominio.....	19
2.2.1 Descripción de conceptos.....	20
2.2 Product Backlog (Requisitos funcionales y no funcionales).....	20
2.2.1 Requisitos Funcionales.....	20
2.2.2 Requisitos No Funcionales.....	24
2.2.3 Modelado de los requisitos.....	25
2.3 Diseño de la interfaz de usuario.....	26
2.3.1 Mapa de Navegabilidad.....	27
2.4 Arquitectura especializada.....	28
2.4.1 Patrón Arquitectónico.....	28
2.4.2 Estilo Arquitectónico.....	29
2.4.3 Selección de la arquitectónico.....	29
2.5 Diagrama de Clases del Diseño y Patrones del Diseño.....	32
2.5.1 Descripción de las clases del diagrama.....	33

2.6 Patrones de Diseño.....	34
2.6.1 Patrones GRASP.....	34
2.6.2 Patrones GOF.....	35
2.7 Modelo Entidad Relación.....	36
2.7.1 Descripción del Modelo de la Base de Datos Suministros Médicos.....	37
Conclusiones del capítulo.....	40
CAPÍTULO 3: Implementación y validación del sistema de gestión de información web para EMSUME.....	41
3.1 Implementación del sistema.....	41
3.1.1 Planificación del Sprint Backlog.....	41
3.1.2 Resultados de la revisión del Sprint.....	43
3.1.3 Diagramad de Despliegue.....	43
3.1.4 Estándares de codificación.....	44
3.2 Pruebas de Software.....	45
3.2.1 Estrategia de Pruebas.....	45
3.2.2 Aplicación de las pruebas.....	46
Conclusiones del capítulo.....	53
CONCLUSIONES FINALES.....	54
RECOMENDACIONES.....	55
REFERENCIAS BIBLIOGRFICAS.....	56
ANEXOS.....	60

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Comparación entre las soluciones existentes. Fuente: Elaboración propia.....	9
<b>Tabla 2.</b> Descripción del Product Backlog. Fuente: Elaboración propia. ....	21
<b>Tabla 3.</b> Historia de usuario RF11. Fuente: Elaboración propia. ....	25
<b>Tabla 4.</b> Historia de usuario RF13. Fuente: Elaboración propia. ....	26
<b>Tabla 5.</b> Descripción del diagrama de clases del diseño. Fuente: Elaboración propia. ....	33
<b>Tabla 6.</b> Descripción de la entidad Gen Producto. Fuente: Elaboración propia. ....	37
<b>Tabla 7.</b> Descripción de la entidad Gen Medida. Fuente: Elaboración propia. ....	38
<b>Tabla 8.</b> Descripción de la entidad Usuario. Fuente: Elaboración propia. ....	38
<b>Tabla 9.</b> Descripción de la entidad Grupos.....	39
<b>Tabla 10.</b> Planificación del Sprint 0. Fuente: Elaboración propia. ....	42
<b>Tabla 11.</b> Planificación del Sprint 1. Fuente: Elaboración propia. ....	42
<b>Tabla 12.</b> Diseño del Caso de prueba "Autenticar Usuario". Fuente: Elaboración propia. ....	47
<b>Tabla 13.</b> Diseño del Caso de prueba "Buscar por descripción". Fuente: Elaboración propia. .....	47
<b>Tabla 14.</b> Técnica de Partición de Equivalencia del Caso de prueba "Autenticar Usuario". Fuente: Elaboración propia.....	47
<b>Tabla 15.</b> Técnica de partición de equivalencia caso de prueba "Buscar productos de entrada". Fuente: Elaboración propia. ....	48
<b>Tabla 16.</b> Niveles de satisfacción expresados en escala numérica. Fuente: Elaboración propia.....	51
<b>Tabla 17.</b> Resultados de la aplicación de la técnica ladov. Fuente: Elaboración propia. ....	51
<b>Tabla 18.</b> Planificación del Sprint 2 .....	62
<b>Tabla 19.</b> Planificación del Sprint 3 .....	63

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Modelo del dominio. Fuente: Elaboración propia. ....	19
<b>Figura 2.</b> Prototipo de interfaz de usuario (Página de Inicio). Fuente: Elaboración propia. ...	27
<b>Figura 3.</b> Prototipo de interfaz de usuario (Autenticar usuario). Fuente: Elaboración propia. ....	27
<b>Figura 4.</b> Mapa de navegabilidad. Fuente: Elaboración propia. ....	28
<b>Figura 5.</b> Modelado del patrón arquitectónico del Front-End. Fuente: Elaboración propia. ...	30
<b>Figura 6.</b> Modelado del patrón arquitectónico del backend. Fuente: Elaboración propia. ....	31
<b>Figura 7.</b> Diagrama de clases del Diseño. Buscar producto por código de entrada. Fuente: Elaboración propia. ....	33
<b>Figura 8.</b> Patrón Controlador. Fuente: Elaboración propia. ....	34
<b>Figura 9.</b> Patrón Bajo Acoplamiento. Fuente: Elaboración propia. ....	35
<b>Figura 10.</b> Patrón Alta Cohesión. Fuente: Elaboración propia. ....	35
<b>Figura 11.</b> Patrón Decorator. Fuente: Elaboración propia. ....	36
<b>Figura 12.</b> Modelo Entidad Relación Base de Datos Suministros Médicos. Fuente: Elaboración propia. ....	37
<b>Figura 13.</b> Modelo Entidad Relación Base de Datos Usuario. Fuente: Elaboración propia. ...	38
<b>Figura 14.</b> Planificación del Sprint. Fuente: Elaboración propia. ....	41
<b>Figura 15.</b> Resultados de la revisión del Sprint. Fuente: Elaboración propia. ....	43
<b>Figura 16.</b> Diagrama de Despliegue. Fuente: Elaboración propia. ....	44
<b>Figura 17.</b> Importaciones en Django. Fuente: Elaboración propia. ....	45
<b>Figura 18.</b> Prueba de integración Autenticar Usuario. Fuente: Elaboración propia. ....	49
<b>Figura 19.</b> Cuadro Lógico Técnica de ladov. Fuente: Elaboración propia. ....	50
<b>Figura 20.</b> Resultados de la Prueba de Aceptación. Fuente: Elaboración propia. ....	51
<b>Figura 21.</b> Entrevista realizada para desarrollar la investigación. ....	60
<b>Figura 22.</b> Diagrama de Clases del Diseño. RF Exportar Inventario. Fuente: Elaboración propia. ....	61
<b>Figura 23.</b> Historia de Usuario. CRUD Usuario. Fuente: Elaboración propia. ....	61
<b>Figura 24.</b> Historia de Usuario. Generar Reporte de productos faltantes. Fuente: Elaboración propia. ....	62

## **OPINIÓN DE LOS TUTORES**

**AVAL DEL CLIENTE**

10 de noviembre/2022.

Carta de aceptación del sistema.

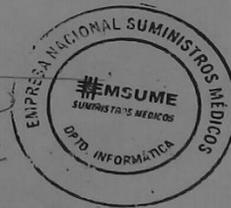
9. Quién pueda interesar.

De: Dirección Desarrollo - Informática. ELSUDE.

Por este medio hacemos constar que es estudiante: Gianna Pérez Rodríguez estudiante de 5<sup>to</sup> año de su Universidad, se encuentra realizando su ejercicio de Tesis en nuestra empresa. La misma se encuentra trabajando en la implementación de un "Sistema de Gestión de Información" con el cual damos solución a una demanda de nuestros bancos de problema.

Con fecha de hoy, la misma nos presenta el producto final, con el cual, concluimos como: "Satisfactorio", pues satisface nuestra demanda.

Isaac  
Guil Isaac Zagarne  
Esp. Informática. ELSUDE  
oprema central



## INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (TIC), se han transformado en uno de los recursos más importantes de la sociedad, tan grande ha sido su impacto que dichas tecnologías se utilizan actualmente en varias esferas de la sociedad dígase: salud, educación, economía, entre otros. Cuba ha impulsado la informatización de la sociedad como una de las políticas más importantes del país. Este proceso lo ha logrado mediante la utilización de las TIC, uno de los sectores que ha tenido gran impacto con estas tecnologías es el de la salud, debido a las grandes ventajas que ofrecen los sistemas de gestión de información para la automatización de tareas e informatización aplicada a la toma de decisiones.

Un sistema de gestión (SG) es una herramienta o aplicación informática que permite controlar todos y cada uno de los aspectos de una empresa (pedidos, producción, control de presencia, facturación, ventas, administración, etc.), un SG se hace imprescindible ya que, no solamente permite un control exhaustivo del ciclo de producción, sino que además permite la interacción con los dispositivos. (CRISTALDO y KLOSTER, 2016)

El Sistema Nacional de Salud, como tarea priorizada de la Revolución, ha decidido informatizar sus procesos, con el objetivo de desarrollarlos con mayor eficiencia y rapidez, evitando la pérdida o duplicación de la información. La informatización del SNS está apoyada en estrategias y políticas trazadas por la dirección del país y el MINSAP, mediante el uso de los sistemas de gestión.

El Ministerio de Salud Pública (MINSAP) es el organismo rector del Sistema Nacional de Salud, encargado de dirigir, ejecutar y controlar la aplicación de la política del Estado y del gobierno en cuanto a la Salud Pública, el desarrollo de las Ciencias Médicas y la Industria Médico Farmacéutica (CASTRO RUZ,2003). Esta institución dentro de sus objetivos incluye la planificación material, la cual, consiste en realizar un conteo de la cantidad de suministros médicos que se necesitan distribuir.

En este proceso interviene la Empresa de Suministros Médicos (EMSUME) perteneciente al Grupo Empresarial de Atención a la Salud Pública (GEASP) del Ministerio de Salud Pública, dicha entidad tiene como objetivo la distribución de suministros como: material gastable, material aséptico de estomatología, piezas de repuesto, material óptico y de laboratorio clínico, al universo de las instituciones de salud del país, para que puedan cumplir su misión social de atención integral a la salud del pueblo. El alcance de los productos que distribuye la entidad, abarca los 253 hospitales, 486 policlínicos y los 10 000 consultorios del médico de la familia con que cuenta el país.

La gestión de almacén en la empresa, es un proceso realizado por el área comercial, dicho proceso incluye el inventario físico para conocer la cantidad de suministros que se encuentran en existencia, además de la cantidad de entradas y salidas del almacén, de esta forma se establece el control de los suministros médicos almacenados.

En esta empresa, el proceso de control de sus almacenes es realizado manualmente por la encargada de dirección comercial de la empresa y los informes son elaborados en un documento de Excel. En cuanto al proceso de control de suministros médicos, hacia otros almacenes, se realiza mediante una llamada telefónica o correo electrónico desde la empresa distribuidora nacional hacia el resto de almacenes que son distribuidores y están vinculados a la misma.

A partir de la aplicación de la entrevista estructurada como técnica seleccionada para la recopilación de información, se identificaron varias limitaciones en la realización del proceso de gestión de almacenes de la empresa, estas limitaciones detectadas, dan origen a la siguiente situación problemática:

- ✓ Demora en la búsqueda de un determinado producto que existe o falta en el almacén.
- ✓ El inventario se realiza de forma manual, esto produce falta de inmediatez en la obtención de informes del inventario.
- ✓ Atraso en la obtención de reportes de productos faltantes.
- ✓ Necesidad de informatizar el proceso de control de los almacenes por la gran demanda e importancia que representa la empresa para el desarrollo sostenible del país.

Dada la situación problemática antes descrita, se identifica como **problema de investigación**: ¿Cómo contribuir al control de almacenes de los suministros médicos por parte de la empresa EMSUME?

Este problema está centrado en el **objeto de estudio**: sistema basado en la web para la gestión de información.

Se define como **campo de acción**: Sistema basado en la web para la gestión de información de almacenes de suministros médicos distribuidos por EMSUME.

Para dar respuesta al problema antes mencionado, se traza como **objetivo general** de la investigación:

Desarrollar un sistema informático para la gestión de los almacenes de suministros médicos distribuidos por EMSUME.

Para guiar el desarrollo de la presente investigación surgen las siguientes **preguntas científicas**:

- ¿Cómo se realiza actualmente la de gestión de almacenes en la empresa?
- ¿Qué metodología, tecnologías y herramientas conforman el ambiente de desarrollo para la implementación del sistema basado en la web para la gestión de almacenes de suministros médicos de la empresa?
- ¿Cómo desarrollar un sistema de información basado en la web para la gestión de almacenes de suministros médicos que se ajuste a las características de la empresa EMSUME?
- ¿Cómo validar la propuesta de solución a partir de los resultados obtenidos?

Para dar solución al objetivo general de la investigación, se proponen las siguientes **tareas de investigación**:

- ✓ Elaboración un marco teórico referencial sobre los sistemas de información basado en la web.
- ✓ Selección y descripción de los métodos, herramientas y tecnologías para el desarrollo del sistema.
- ✓ Obtención de los requisitos funcionales y no funcionales del sistema.
- ✓ Implementación de un sistema de información basado en la web para la gestión de almacenes de suministros médicos de EMSUME.
- ✓ Elaboración y aplicación de los casos de prueba para la propuesta de solución.
- ✓ Validación de la gestión de almacenes de suministros médicos con la solución propuesta a partir de los resultados obtenidos.

Con el desarrollo de la presente investigación se obtiene como posible resultado un: sistema basado en la web, que permita, la gestión de los almacenes de suministros médicos que se encuentran en la empresa EMSUME.

Para darle cumplimiento a las tareas de investigación se emplean varios **métodos científicos** en la búsqueda y procesamiento de los datos tales como:

Métodos teóricos:

Analítico-Sintético: Para definir las principales características y ventajas de las aplicaciones web, los lenguajes de programación a utilizar y las tecnologías; así como los sistemas de conexión segura a servidores web y servidores de Bases de Datos para almacenar la información de la aplicación. (FALCÓN,2021)

### **Métodos Empíricos**

- Análisis documental: Este método se aplicó para extraer la información necesaria de la bibliografía existente y consultar información en sitios de interés nacional e internacional, con el objetivo de apoyar las tareas de investigación (FALCÓN,2021)

### **Técnica de indagación empírica**

- Entrevista: Esta técnica se aplicó a los trabajadores de la empresa EMSUME, para conocer los recursos con los que disponen para el desarrollo de un sistema basado en la web que permita satisfacer sus necesidades (ARIAS,2020).

El contenido de la investigación se encuentra estructurado en tres capítulos:

### **Capítulo 1: Fundamentos de los sistemas gestión de información**

Este capítulo se centra en los aspectos teóricos y conceptos fundamentales relacionados con el objeto de estudio y el campo de acción de la investigación. Se analizan las soluciones existentes tanto en el ámbito internacional como en el nacional y se describen las herramientas a utilizar para el diseño e implementación de la solución.

### **Capítulo 2: Análisis y Diseño del sistema.**

En este capítulo se realiza una propuesta del sistema basado en la web, se describe cómo debe funcionar y se destacan sus características distintivas; además, se especifican sus requisitos funcionales y no funcionales y se muestran los modelos necesarios para la realización del diseño de la propuesta de solución.

**Capítulo 3: Implementación y Validación del Sistema.**

Se analizan los procesos referentes al desarrollo de la solución y las pruebas realizadas para validar el correcto funcionamiento del sistema basado en la web.

Finalmente, se presentan las Conclusiones, Recomendaciones, Referencias Bibliográficas y Anexos derivados de la investigación.

## Capítulo 1: Fundamentos de los sistemas de gestión de información.

Este capítulo está destinado a brindar una panorámica de todo lo referente a la gestión de almacén, concepto y funciones del mismo. Se realizan comparaciones entre los sistemas gestión de información que existen a nivel nacional e internacional. También se describen las herramientas, tecnologías y la metodología que se utilizará para el proceso de desarrollo de software.

### 1.1 Conceptos Fundamentales.

A continuación, se exponen los principales conceptos relacionados con el marco teórico de la investigación, con el objetivo de profundizar en los distintos puntos de vista y asumir una posición al respecto.

#### **Gestión:**

Es la actividad dirigida a obtener y asignar los recursos necesarios para el cumplimiento de los objetivos de la organización (MURRAY,2002).

#### **Información:**

Según la Real Academia de la Lengua Española, se define la información como la comunicación o adquisición de conocimientos que permiten ampliar o precisar los que se poseen sobre una materia determinada.

#### **Gestión de Información:**

Según Phil Bartle, “la gestión de la información es el proceso de analizar y utilizar la información que se ha recabado y registrado para permitir a los administradores (de todos los niveles) tomar decisiones documentadas”. La gestión de la información es el conjunto de las actividades que se realizan con el propósito de adquirir, procesar, almacenar y finalmente recuperar, de manera adecuada, la información que se produce o se recibe en una organización y que permite el desarrollo de su actividad (SUÁREZ ALFONSO, y otros, 2015). A partir de esta definición la autora de la presente investigación define que la gestión de información es el tratamiento, procesamiento y extracción de la información, con el objetivo de mejorar la toma de decisiones y garantizar un mejor desempeño en las organizaciones.

#### **Definiciones de gestión de almacén según autores:**

La gestión de almacén permite controlar unitariamente los productos y ubicarlos correctamente para reducir al máximo las operaciones de mantenimiento, los errores y el tiempo de dedicación (FLAMARIQUE, 2018).

Según (Correa,2010), la gestión de almacenes, es un proceso clave que busca regular los flujos entre la oferta y la demanda, optimizar los costos de distribución y satisfacer los requerimientos de ciertos procesos productivos (ESPINAL,2010).

A partir de los conceptos analizados anteriormente, la autora de la presente investigación concluye que: “la gestión de almacén representa un conjunto de actividades que están

orientadas al control, clasificación y ubicación de los productos que se encuentran almacenados, con el objetivo de mejorar la organización del almacén.”

### **Almacén:**

Un almacén es un espacio debidamente dimensionado que la empresa destina a la ubicación y manipulación eficiente de sus materiales y mercancías (BRENES, 2015).

### **Actividades de un almacén**

- Recepción: Se reciben los productos enviados de los proveedores. Se comprueba la mercancía con la información del albarán y el pedido.
- Expedición: Se prepara y se entrega la mercancía a los clientes según vayan demandando a los productos.
- Almacenamiento: Se ubican los productos en un lugar donde se pueda localizar y acceder.
- Conservación y manutención: Se conserva la mercancía en la mejor situación posible durante el tiempo que permanezca en el almacén.
- Control y organización de existencia: Se realiza un control exhaustivo de las existencias en el almacén para no dejar desabastecidos a los clientes y conseguir de este modo una gestión eficiente del espacio (BRENES, 2015).

### **Almacenaje:**

Actividad de depósito y manipulación de los productos, componentes, materiales y materias primas para poder garantizar el buen funcionamiento de la actividad empresarial (BRENES, 2015).

### **Inventario:**

El inventario se define como una relación detallada, ordenada y valorada de los elementos que componen el patrimonio de una organización. En el almacén, el inventario es la relación detallada de los productos almacenados, en unidades económicas (FLAMARIQUE, 2019).

### **Gestión de Inventario:**

Es el proceso estratégico dentro del aprovisionamiento dirigido a administrar eficientemente el inventario, mediante medidas coherentes de control y planificación de la demanda, sin afectar el servicio al cliente (DE GUEVARA, 2020).

### **¿Qué es el inventario informático?**

Los inventarios informáticos son simples listados en los cuales el sistema ha calculado cada una de las referencias, teniendo en cuenta las existencias iniciales, las entradas y las salidas (FLAMARIQUE, 2019).

## **1.2 Sistemas de Gestión de Información.**

A continuación, se profundiza sobre el objeto de estudio que se contempla en esta investigación:

Un sistema de información es un conjunto de componentes que interaccionan entre sí para alcanzar un fin determinado, el cual es satisfacer las necesidades de información de dicha organización. Estos componentes pueden ser personas, datos, actividades o recursos

materiales en general, los cuales procesan la información y la distribuyen de manera adecuada, buscando satisfacer las necesidades de la organización. El objetivo primordial de un sistema de información es apoyar la toma de decisiones y controlar todo lo que en ella ocurre.

A su vez, la gestión es el proceso mediatizado por un conjunto de actividades que permiten la obtención de información, lo más pertinente, relevante y económica posible, para ser usada en el desarrollo y el éxito de una organización. Con lo antes expuesto se concluye que, un SGI permite la gestión de los recursos de información tanto internos como externos. Su finalidad es generar servicios y productos que respondan a las necesidades y sobrepasen las expectativas de los usuarios, posibilitando que el sistema trabaje eficientemente y económicamente a la vez.

El SGI aprovecha al máximo sus recursos de información en función de la mejora continua y de la toma de decisiones organizacional a todos los niveles jerárquicos desde la cúspide estratégica hasta la base operativa (FERNÁNDEZ, 2008).

Elusa Morales Flores plantea que la finalidad de la Gestión de la información es ofrecer mecanismos que permitieran a la organización adquirir, producir y transmitir, al menos coste posible, datos e informaciones con una calidad, exactitud y actualidad suficientes para servir a los objetivos de la organización (FLORES, 2014).

### **1.3 Análisis de los sistemas existentes.**

En el proceso de investigación y búsqueda de sistemas homólogos que puedan brindar solución o facilitar la gestión de almacén de la empresa EMSUME, se analizarán como sistemas internacionales en la investigación: SAP ERP y ERP Microsoft Great Plains y en cuanto a las soluciones nacionales se analizarán los sistemas: VERSAT-SARASOLA y RODAS XXI.

#### **1.3.1 Sistemas ERP.**

Según Borda Salamanca, (2019), los Erp son instrumentos informáticos que conectan las distintas áreas de una empresa con el objetivo de agrupar información y ofrecer las pautas para crear reportes precisos, con detalles y datos concretos, que faculten a cada uno de los colaboradores a cumplir su tarea de forma efectiva.

#### **1.3.2 Soluciones informáticas Internacionales.**

##### **SAP ERP:**

Con más de 40 años de experiencia y 50.000 clientes, el software ERP es líder en el mercado es una base comprobada y fiable diseñada para dar soporte tanto a las organizaciones más grandes del mundo, como a las pequeñas y medianas empresas de 25 industrias diferentes. Permite optimizar los procesos en todos los departamentos, incluidos compras, producción, servicios, ventas, finanzas y recursos humanos. Este sistema posee costos de licenciamiento

altos y posibilidad nula de adaptar nuevas funcionalidades por desarrollo propio (software propietario) (CAIZA MUYOLEMA, 2015).

#### **ERP Microsoft Great Plains:**

El software planificación de recursos empresariales (ERP) de Microsoft Dynamics le ofrece a su empresa (pequeña, mediana o grande) las herramientas para administrar toda su organización, desde la cadena de abastecimiento, contratación y recursos humanos, hasta finanzas y proyectos en colaboración. Este software es propietario y el precio de su licencia es muy elevado (CAIZA MUYOLEMA, 2015).

#### **1.3.3 Soluciones Nacionales.**

##### **VERSAT-SARASOLA**

Versat es el primer sistema de contabilidad cubano certificado. Conformado por 12 módulos que permiten llevar el control y registro contable individual de todos los hechos económicos que se originan en las estructuras internas de las entidades, siendo configurable cada una de ellas en el momento de su instalación (ROMERO, 2009).

##### **RODAS XXI versión 3.0:**

Sistema multiempresa y multiusuario creado por CITMATEL2 para automatizar la gestión empresarial. Contiene diferentes módulos que pueden usarse integrados o independientes. El módulo de Inventario de Rodas XXI le permite tener un control detallado de los inventarios de su entidad, realizando en el mismo momento que se registra un movimiento, su contabilización. Se pueden realizar todo tipo de operaciones de entradas y salidas de los almacenes con facilidad en el momento que se desee, generando el documento asociado al movimiento (CIMATEL, 2022).

##### **Características del Sistema:**

A continuación, se presentan las características que deben cumplir las soluciones informáticas que se analizaron. Se analizaron estas características por ser las necesarias para satisfacer los requerimientos de la problemática del cliente.

##### **Software Libre:**

El software libre se basa en la cooperación y se basa en una serie de libertades para el usuario, permitiéndole tener acceso al código fuente del programa y modificar ciertas funcionalidades para adaptarlo a las necesidades de cada usuario o empresa (AGUILAR,2018).

##### **Costo de Licencia:**

El gasto que representa para la empresa adquirir la licencia para el uso del sistema.

##### **Sistema Web:**

Aplicación que funciona mediante el acceso a un servidor web a través de internet mediante un navegador.

##### **Módulo de inventario:**

El módulo de inventario permite exportar informes de inventario con la mercancía que se encuentra almacenada.

### 1.3.4 Comparación entre las soluciones existentes.

A continuación, se establece una comparación entre los sistemas analizados.

**Tabla 1.** Comparación entre las soluciones existentes. Fuente: Elaboración propia.

Soluciones Informáticas	Software Libre	Costo de Licencia (Bajo)	Aplicación Web	Módulo de Inventario
SAP ERP	No	No	No	Sí
Erp Microsoft Great Plains	No	No	No	Sí
RODAS XXI Versión 3.0	No	Sí	No	Sí
VERSAT-SARASOLA	No	Sí	No	Sí

### 1.3.5 Valoración crítica de los sistemas existentes.

A partir del análisis y la comparación de los sistemas estudiados tanto internacionales como nacionales, se determinó que los sistemas internacionales (**SAP ERP y ERP Microsoft Great Plains**), son sistemas privativos y el pago de las patentes o licencias es un aspecto a tener en cuenta, pues el país no se encuentra en condiciones económicas de realizar estos pagos. Otro punto que se debe tener en cuenta es que estos sistemas se ajustan a las políticas de los sistemas capitalistas.

En el caso de la solución nacional (**Rodas XXI**) está desarrollada sobre software propietario cuando el país está en proceso de emigrar hacia el software libre. Además de ser una aplicación de escritorio, por lo que se hace necesario la instalación en todas las computadoras de la entidad. En cuanto al software **VERSAT-SARASOLA**, se puede determinar que esta solución tampoco satisface las necesidades de la empresa, pues, es una aplicación de escritorio y su uso solo satisface las necesidades de los almacenes internos de la empresa, además que cuenta con una licencia privada y la entidad necesita un sistema basado en la web y de uso gratuito.

Por todo lo antes expuesto se concluye, que se hace necesario crear un sistema gestión que permita el control de los almacenes tanto internos como externos a la empresa, sea capaz de realizar el inventario de los almacenes y permita buscar los suministros que se encuentran almacenados, además el sistema debe ser consecuente con la filosofía de conseguir la soberanía tecnológica, basada en herramientas y tecnologías libres.

## 1.4 Lenguajes, Metodologías y Herramientas.

A continuación, se describen los lenguajes de desarrollo que se utilizaron en esta investigación, la metodología de software que guía el proceso de desarrollo de software y las herramientas a utilizar.

### 1.4.1 ¿Qué es una metodología de desarrollo de software?

Una metodología de desarrollo de software es un marco de trabajo utilizado para estructurar, planificar y controlar el proceso de desarrollo de software. Este marco de trabajo consiste en una filosofía de desarrollo de software, herramientas, modelos y métodos que asisten al proceso de desarrollo de software (MOLINA, 2013).

#### Metodologías Ágiles

Las metodologías ágiles son flexibles, sus proyectos son subdivididos en proyectos más pequeños, incluyen comunicación constante con el cliente, son altamente colaborativos y se adaptan mejor a los cambios (AL-SAQQA,2020).

#### 1.4.1.1 ¿Qué es una metodología de desarrollo de software?

La metodología SCRUM es un marco de trabajo diseñado para lograr la colaboración eficaz de equipos en proyectos, que emplea un conjunto de reglas y artefactos y define roles que generan la estructura necesaria para su correcto funcionamiento. SCRUM utiliza un enfoque incremental (KEN, 2018).

#### Fundamentos de SCRUM

**Sprint:** Evento principal que corresponde a una ventana de tiempo donde se crea una versión utilizable del producto (incremento).

**Elementos de un Sprint** (KEN, 2018).

- **Planeación del Sprint:** Se define su plan de trabajo: qué se va a entregar y cómo se logrará. Es decir, el diseño del sistema y la estimación de cantidad de trabajo.
- **Daily Scrum:** Es un evento del equipo de desarrollo de quince minutos, que se realiza cada día con el fin de explicar lo que se ha alcanzado desde la última reunión; lo que se hará antes de la siguiente; y los obstáculos que se han presentado (KEN, 2018).
- **Revisión del Sprint:** ocurre al final del Sprint y su duración es de cuatro horas para un proyecto de un mes (o una proporción de ese tiempo si la duración es menor). En esta etapa: el dueño del proyecto revisa lo que se hizo, identifica lo que no se hizo y discute acerca del Product Backlog; el equipo de desarrollo cuenta los problemas que encontró y la manera en que fueron resueltos, y muestra el producto y su funcionamiento (KEN, 2018).
- **Retrospectiva del Sprint (Feedback):** Es una reunión de tres horas del equipo Scrum en la que se analiza cómo fue la comunicación, el proceso y las herramientas; qué estuvo bien, qué no, y se crea un plan de mejoras para el siguiente Sprint.

## Roles de SCRUM

- **Product Owner:** Es la única persona autorizada para decidir las funcionalidades y características funcionales que tendrá el producto. Es quien representa al cliente y usuarios del software (KEN, 2018).
- **Scrum Máster:** Es quien interactúa con el equipo, el cliente y los gestores. Garantiza el funcionamiento de los procesos y la metodología. Debe ser miembro del equipo y trabajar a la par. Coordina los encuentros diarios y se encarga de eliminar obstáculos (MOLINA, 2013).
- **Scrum Team:** Es el equipo de desarrolladores multidisciplinario, integrado por programadores, diseñadores, arquitectos y testers, que en forma auto-organizada son los encargados de desarrollar el producto (KEN, 2018).

## Artefactos de SCRUM (KEN, 2018)

- **Product Backlog (Pila de Producto):** Es una lista ordenada por valor, riesgo, prioridad y necesidad de los requerimientos que el dueño del producto define, actualiza y ordena.
- **Sprint Backlog:** Es un subconjunto de ítems del Product Backlog y el plan para realizar en el Incremento del producto. Debido a que el Product Backlog está organizado por prioridad, el Sprint Backlog es construido con los requerimientos más prioritarios del Product Backlog y con aquellos que quedaron por resolver en el Sprint anterior.
- **Incremento:** Es la suma de todos los ítems terminados en el Sprint Backlog. Si hay ítems incompletos deben ser devueltos al Product Backlog con una prioridad alta para que sean incluidos en el siguiente Sprint.

### I.4.1.2 Fundamentación del uso de la Metodología SCRUM.

Se seleccionó para guiar el proceso de desarrollo de software la metodología SCRUM, como parte del cumplimiento de las exigencias del cliente, el cual planteó la necesidad de estar involucrado durante todo el proceso de desarrollo para validar la propuesta de solución a su problemática. El uso de esta metodología ágil tiene como ventajas: (RODRÍGUEZ & DORADO, 2015)

- ✓ **Satisfacción del cliente:** El gran diferenciador de las metodologías ágiles, es que hacen al cliente parte del equipo de trabajo y lo comprometen con el resultado final. Esto es un gran cambio con respecto a metodologías tradicionales, en los que el cliente es una persona, o grupo de personas, con el que se realiza la tarea de levantar requerimientos o necesidades de tipo funcional y que posteriormente aprueba extensos documentos que no volverá a ver hasta que el producto se encuentre terminado.
- ✓ **Simplicidad:** los eventos manejados por Scrum están claramente identificados, indicando para cada uno: quienes participan, su objetivo, el tiempo que debe tomar y cuál es el resultado esperado. Lo cual en esencia facilita a los integrantes del equipo la adopción de la metodología.

- ✓ Inspección: uno de los componentes que resalta Scrum, es la inspección y por ello, tres de sus eventos están orientados a estos objetivos: la reunión diaria, la revisión del sprint y la retrospectiva de este último.
- ✓ **Adaptación:** la mejor parte de la metodología es la disposición que tienen al cambio las características del producto. Este es uno de los componentes que más la diferencia con el resto, ya que el cambio puede ser efectuado en cualquier momento, incluso dentro del desarrollo de la ejecución de las diferentes iteraciones o Sprints siempre y cuando no afecte la entrega pactada.

### 1.4.2 Lenguaje de modelado.

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables (GARCÍA ,2020).

#### Dentro de sus características resaltan:

- ✓ Tecnología orientada a objetos.
- ✓ Viabilidad en la corrección de errores.
- ✓ Participación del cliente en todas las etapas del proyecto.
- ✓ Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.

### 1.4.3 Herramienta de modelado.

A continuación, se presenta la herramienta de modelado que se utilizó para la elaboración de diagramas basados en UML.

**Visual Paradigm para UML 8.0:** La herramienta de modelado Visual Paradigm for UML 8.0 es una herramienta CASE profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue; y no emplea una metodología en específico. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad (PARADIGM, 2020).

Esta herramienta ofrece las siguientes ventajas:

#### Ventajas:

- ✓ Es multiplataforma.
- ✓ Generador de informes para generación de documentos.
- ✓ Ingeniería Inversa de Base de Datos, desde sistemas gestores de bases de datos existentes a diagramas entidad relación.
- ✓ Importación y exportación de ficheros.
- ✓ Soporte de UML.

#### 1.4.4 Lenguajes y Tecnologías.

A continuación, se describen los lenguajes y tecnologías seleccionadas para el desarrollo de la investigación.

##### **Servidor Web**

Un servidor web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS (la versión cifrada y autenticada).

##### **Servidor Web Apache 2.2.4**

Apache es un servidor web gratuito, potente, flexible, funciona en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y los diferentes entornos, hacen que a menudo sean necesarias diferentes características o funcionalidades, o que en una misma característica o funcionalidad sea implementada de diferente manera para obtener una mayor eficiencia (BOTH,2020).

##### **Lenguaje de desarrollo.**

Un lenguaje de desarrollo es una herramienta que permite la implementación de un software ya sea de tipo móvil, web o una aplicación desktop.

##### **JavaScript 1.7**

JavaScript es el lenguaje que nos permite interactuar con el navegador de manera dinámica y eficaz, proporcionando a las páginas web rapidez y vida. JavaScript comparte muchos elementos con otros lenguajes de alto nivel. Hay que tener en cuenta que este lenguaje es muy semejante a otros como C, Java o PHP, tanto en su formato como en su sintaxis, aunque tiene sus propias características definitorias (LUNA, 2019).

##### **Ventajas del Lenguaje:**

- ✓ Es multiplataforma.
- ✓ Es versátil para el desarrollo web dinámico y de móviles.
- ✓ Se relaciona de modo fluido con HTML y CSS.

##### **Python 3.9.5:**

Python es un lenguaje de programación multiparadigma, pues soporta orientación a objetos, programación imperativa, programación orientada a aspectos y en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico, es fuertemente tipado y multiplataforma (WALKER,2018).

##### **Ventajas del uso de Python:**

- ✓ Propósito general: se pueden crear todo tipo de programas.
- ✓ Multiplataforma: hay versiones disponibles de Python en muchos sistemas informáticos distintos.
- ✓ Facilidad de uso: lenguaje con una rápida curva de aprendizaje lo que permite empezar a hacer programas sencillos en Python en muy poco tiempo.
- ✓ Orientado a Objetos: la programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes

reutilizables. Además, Python también permite la programación imperativa, programación funcional y programación orientada a aspectos.

- ✓ Funciones y librerías: existen una gran variedad de bibliotecas disponibles para extender la funcionalidad básica de Python a cualquier campo.

A continuación, se describen los marcos de trabajo a utilizar teniendo en cuenta lenguaje de programación seleccionado.

#### **Framework:**

Un framework es una estructura de archivos y utilidades que aceleran la programación de una aplicación informática, que provee una metodología de trabajo que sistematiza y facilita la generación de formularios, funciones y módulos de uso común, permitiendo al desarrollador dedicar su atención hacia los aspectos específicos de cada aplicación (LÓPEZ 2008).

#### **Django 4.0.6:**

Django es un marco web Python de alto nivel que fomenta un desarrollo rápido y un diseño limpio y pragmático. Creado por desarrolladores experimentados, se encarga de gran parte de las molestias del desarrollo web, por lo que puede concentrarse en escribir su aplicación sin necesidad de reinventar la rueda. Es gratis y de código abierto. Abstrae a los programadores de los problemas comunes del desarrollo web y acelera las tareas más frecuentes en la programación (VINCENT, 2021).

- ✓ A través de plantillas ayuda a separar el contenido de la presentación evitando tener que manipular la lógica de negocio cuando se necesite realizar cambios de apariencia en la página.
- ✓ Django incluye muchas aplicaciones comunes a todos los sitios web, como la autenticación de usuarios o la administración del contenido del sitio.
- ✓ Su arquitectura está inspirada en el patrón Modelo-Vista-Controlador (MVC), encargándose en gran parte de los controladores, y proveyendo herramientas para facilitar el desarrollo de las vistas y los modelos.

Teniendo en cuenta estas características, se escoge este marco de trabajo en su versión 4.0.6 para la implementación del sistema, pues se centra en el desarrollo rápido, la reutilización y la seguridad.

#### **Entorno de desarrollo Integrado:**

Un entorno de desarrollo integrado o IDE (Integrated Development Environment) es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código. Estas herramientas pueden estar creadas para su utilización con un único lenguaje.

#### **PyCharm:**

PyCharm 2021.3, es un entorno de desarrollo integrado multiplataforma utilizado para desarrollar en el lenguaje de programación Python. (JETBRAINS,2020)

Provee funcionalidades que permiten una experiencia única y aumentan en gran medida la productividad:

- ✓ PyCharm ofrece un soporte de primer nivel para JavaScript, CoffeeScript, TypeScript, HTML y CSS, así como para sus sucesores modernos. El depurador JavaScript se incluye con PyCharm y está integrado con la configuración de ejecución del servidor Django.
- ✓ PyCharm funciona en Windows, macOS o Linux. Puede instalar y ejecutar PyCharm en tantas máquinas como tenga, y usar el mismo entorno y funcionalidad en todas sus máquinas.
- ✓ PyCharm ofrece una gran compatibilidad específica para cada marco de trabajo de desarrollo web como Django, Flask, Google App Engine, Pyramid y web2py, incluyendo depurador de plantillas Django.

Por todas las características anteriormente expresadas, se decidió la utilización del IDE de desarrollo PyCharm -2021.3.

### **Django Rest Framework:**

Django REST framework es un conjunto de herramientas potente y flexible para crear API web (VAINIKKA,2018).

### **API:**

Las siglas API vienen del inglés (Application Programming Interface, Interfaz de Programación de Aplicaciones), es un conjunto de funciones y procedimientos que cumplen una o muchas funciones con el fin de ser utilizadas por otro software. La API permite implementar las funciones y procedimientos que engloba en el proyecto sin la necesidad de programarlas de nuevo. En términos de programación, es una capa de abstracción. (RUBIO,2017)

### **REST:**

REST (Representational State Transfer, Transferencia de Estado Representacional) es cualquier interfaz entre sistemas que usen HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON. REST se define como un tipo de arquitectura más natural y estándar para crear API's para servicios orientados a Internet. Es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP (RUBIO,2017).

### **Características de API REST:**

- ✓ Protocolo cliente/servidor sin estado: cada petición HTTP contiene toda la información necesaria para ejecutarla, lo que permite que ni cliente ni servidor necesiten recordar ningún estado previo para satisfacerla.
- ✓ Las operaciones más importantes relacionadas con los datos en cualquier sistema REST son: POST, PUT, DELETE, GET.
- ✓ Los objetos en REST siempre se manipulan a partir de la URL, el identificador único de cada recurso de ese sistema REST. Facilita acceder a la información para su modificación, borrado o para compartir su ubicación exacta con terceros (RUBIO,2017).

### **VueJS v.3.0:**

Vue.js es un marco JavaScript progresivo de código abierto para crear interfaces de usuario (UI) y aplicaciones de una sola página. Este marco utiliza "alto desacoplamiento", lo que permite a los desarrolladores crear progresivamente interfaces de usuario (UI). (NAVARRO,2021)

### **Ventajas del uso de Vue:**

- ✓ Fácil de aprender y utilizar
- ✓ Reactividad: La aplicación reacciona a los cambios sin que sea necesaria la recarga completa de la página, esto permite mejorar la experiencia de usuario.
- ✓ Modularidad: VueJS trabaja con librerías separadas, lo que permite añadir funciones e interactividad al desarrollo.

### **Vue Devtools:**

Vue Devtools es una valiosa extensión de navegador para Chrome y Firefox que acelerará su desarrollo y búsqueda de errores (NELSON,2018).

### **Axios:**

Axios es una librería JavaScript que puede ejecutarse en el navegador y permite hacer sencillas las operaciones de un cliente HTTP, permite realizar solicitudes a un servidor y recibiremos respuestas fáciles de procesar. Es considerada una de las librerías más populares y cuenta con algunas características que la API Fetch no proporciona, como, por ejemplo: interceptores, convertir la respuesta del servidor a JSON automáticamente y también proporcionar protección contra ataques cross-site (HERRERA,2020).

### **Vuetify 2.6.12**

Vuetify es una biblioteca de interfaz de usuario de Vue. Incluye una colección integral de componentes preconstruidos que están listos para usar en producción desde el primer momento. (NELSON, 2018)

### **JSON:**

JSON por sus siglas significa (JavaScript Object Notation), es un formato ligero de intercambio de datos que se envían mediante una API (SOURD,2022).

### **SQLite 3:**

Es un motor de base de datos SQL transaccional de código abierto, ligero, autónomo, de configuración simple y sin servidor, que se caracteriza por almacenar información persistente de forma sencilla. (BHOSALE,2015)

### **SQL Server 2017:**

SQL Server es el sistema de base de datos profesional desarrollado por Microsoft. Contiene una variedad de herramientas y características que se puede usar para desarrollar y administrar base de datos. Es un Sistema de administración de datos gratuito, eficaz y confiable que ofrece un almacén de datos completo y confiable para sitios web ligeros y aplicaciones de escritorio. SQL Server tiene soporte para Windows y Linux (SANTAMARÍA,2016).

**Ventajas:**

- ✓ Soporte de transacciones
- ✓ Escalabilidad, estabilidad y seguridad.
- ✓ Soporte de procedimientos almacenados.

Se seleccionó el uso de estos lenguajes y tecnologías por su integración, se ajustan a las necesidades de la empresa y por las ventajas que ofrece su utilización para el proceso de desarrollo, además de ser de uso gratuito evitando costos adicionales a la empresa.

## Conclusiones parciales del capítulo

Después de realizar el estudio de la bibliografía relacionada con los sistemas basado en la web para la gestión de información de los almacenes, se pudo identificar que las soluciones existentes no satisfacen las condiciones deseadas. Por esta razón se propone la implementación de un sistema que sea capaz de ajustarse a los requerimientos de la empresa y que se nutra de las principales ventajas que muestran algunas de las soluciones analizadas como por ejemplo la obtención de informes de inventarios de forma diaria. Además, debe ser una aplicación web para que cumpla las necesidades de la empresa.

Del análisis sobre las tendencias actuales en el desarrollo del software, se seleccionó la metodología de desarrollo y un grupo de tecnologías y herramientas adecuadas para realizar el sistema gestión de información basado en la web que se requiere: como metodología de desarrollo, la metodología ágil SCRUM, como sistemas de gestión de base de datos SQL Server 2017 y SQLite 3 para el almacenamiento de la información, como herramienta para el modelado se seleccionó Visual Paradigm for UML 8.0, Python y JavaScript como lenguaje de programación, se empleó como framework de desarrollo: Django en su versión 4.0.6 y como framework CSS se utilizó Vuetify en su versión 2.6.12 , haciendo uso de PyCharm 2021.3 como IDE de desarrollo.

## Capítulo 2: Diseño del Sistema de gestión de información para los almacenes de suministros médicos de EMSUME.

En este capítulo se define el modelo de dominio y se da una propuesta de solución al problema planteado. Se identifican, especifican y describen los requisitos no funcionales y funcionales. Se brinda un acercamiento a la implementación a través de la etapa de diseño, construyéndose para ello el mapa de navegabilidad, los prototipos de interfaz de usuario y el diagrama de clases del diseño. Además, se selecciona el estilo arquitectónico, se modela la arquitectura del sistema a través del patrón modelo-vista-plantilla y el patrón modelo-vista-modelo de vista y se muestra el modelo de datos con la descripción de las tablas de la base de datos.

### 2.1 Modelado del dominio.

Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. Se realiza cuando no se logra determinar el proceso del negocio con fronteras bien establecidas y donde los flujos de información son difusos (múltiples orígenes, sólo eventos, sucesos), cuando existe solapamiento de responsabilidades, así como múltiples responsabilidades. Se representa con un conjunto de diagramas de clases UML en los que no se define ninguna operación (LARMAN, 2009). En la Figura 1 se muestra el Modelo de Domino del Sistema de Gestión de Información para los almacenes de suministros médicos distribuidos por EMSUME.

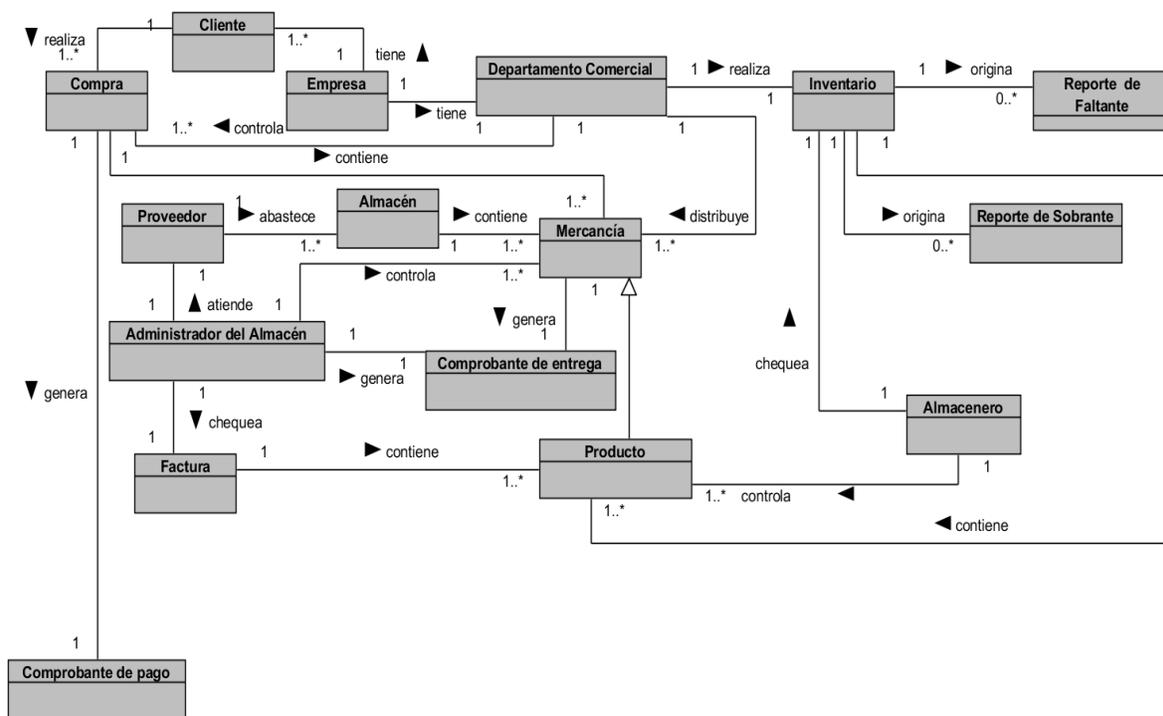


Figura 1. Modelo del dominio. Fuente: Elaboración propia.

### 2.2.1 Descripción de conceptos.

A continuación, se presenta la descripción de los conceptos modelados en el modelo conceptual.

- ✓ Proveedor: Es quien abastece el almacén de la empresa.
- ✓ Almacén: Espacio donde se guarda la mercancía.
- ✓ Mercancía: Producto que se comercializa en la empresa y es almacenado en los almacenes.
- ✓ Producto: suministro médico que comercializa y distribuye la empresa.
- ✓ Inventario: Documento que contiene la cantidad de suministros que entran, existen y se venden en la empresa.
- ✓ Reporte de faltante: Documento que contiene los suministros que faltan por entrar al almacén.
- ✓ Reporte de sobrante: Documento que contiene los suministros que sobran.
- ✓ Departamento Comercial: Departamento de la empresa EMSUME, que se encarga de realizar el inventario de los suministros médicos.
- ✓ Empresa: Encargada de la dirección comercial y de atender a los clientes.
- ✓ Compra: Representa la solicitud de compra mayorista de los suministros de la empresa.
- ✓ Cliente: Es quien realiza la solicitud de compra (711). El cliente representa otras instituciones de salud.
- ✓ Comprobante de pago: Informe que se genera a partir del pago de la venta que se le realiza al cliente.
- ✓ Almacenero: Encargado de depositar en el almacén los suministros.
- ✓ Administrador del Almacén: Encargado de atender al proveedor y de generar el comprobante de entrega una vez que llega la mercancía a los almacenes.
- ✓ Comprobante de entrega: Informe que se genera una vez que los suministros llegan a los almacenes de la empresa.
- ✓ Factura: Informe que contiene el costo de los suministros que se adquieren para los almacenes.

### 2.2 Product Backlog (Requisitos funcionales y no funcionales).

La ingeniería de requisitos es el proceso de desarrollar una especificación de software. Las especificaciones pretenden comunicar las necesidades del sistema del cliente a los desarrolladores del sistema (SOMMERVILLE,2011).

#### 2.2.1 Requisitos Funcionales.

Un requisito funcional es una capacidad o condición que el sistema debe cumplir (PRESSMAN,2010).

#### Product Backlog:

**RF1.** Registrar usuario (nombre, apellidos, usuario, correo, contraseña)

- RF2. Eliminar usuario por nombre de usuario(usuario)
- RF3. Editar usuario (nombre, apellidos, correo, usuario, contraseña)
- RF4. Buscar usuario por nombre de usuario(usuario)
- RF5. Autenticar usuario (usuario, contraseña)
- RF6. Listar productos de entrada ()
- RF7. Buscar productos de entrada por nombre de entidad(nombre\_entidad)
- RF8. Buscar productos por número de entrada(número\_entrada)
- RF9. Buscar productos de entrada por código(código\_producto)
- RF10. Buscar productos de entrada por descripción(descripción)
- RF11. Buscar productos de entrada por nombre del almacén(nombre\_almacén)
- RF12. Buscar productos de entrada por estado(estado)
- RF13. Exportar informe de inventario ()
- RF14. Listar productos en existencia ()
- RF15. Buscar producto en existencia por código de almacén(código\_almacén)
- RF16. Buscar producto en existencia por nombre de almacén(nombre\_almacén)
- RF17. Buscar producto en existencia por código del producto (código\_producto)
- RF18. Buscar producto en existencia por descripción (descripción\_producto)
- RF19. Listar productos en venta ()
- RF20. Buscar productos en venta por código del almacén(código\_almacén)
- RF21. Buscar productos en venta por nombre de almacén(nombre\_almacén)
- RF22. Buscar productos en venta por número de factura (número\_factura)
- RF23. Buscar productos en venta por código de entidad(código\_entidad)
- RF24. Buscar productos en venta por nombre de la entidad(nombre\_entidad)
- RF25. Buscar productos en venta por código de producto (código\_producto)
- RF26. Buscar productos en venta por descripción de producto(descripción\_producto)
- RF27. Buscar productos en venta por estado(estado)
- RF28. Listar productos faltantes ()
- RF29. Buscar productos faltantes por nombre del almacén(nombre\_almacén)
- RF30. Generar reporte de productos faltantes en el almacén ()

**Tabla 2.** Descripción del Product Backlog. Fuente: Elaboración propia.

ID	Nombre del Requisito	Descripción	Prioridad
RF1	Registrar usuario.	El sistema permite registrar a un usuario.	Media
RF2	Editar usuario	El sistema permite editar los datos de un usuario.	Media
RF3	Buscar usuario por nombre de usuario	El sistema permite buscar a un usuario mediante su nombre.	Media
RF4	Eliminar usuario por nombre de usuario.	El sistema permite eliminar los datos de un usuario mediante su nombre	Media

RF5	Autenticar Usuario.	El sistema permite que el usuario acceda al mismo mediante su nombre de usuario y contraseña.	Media
RF6	Listar productos de entrada.	El sistema permite obtener un listado de los productos que entran a los almacenes.	Alta
RF7	Buscar productos de entrada por nombre de entidad.	El sistema permite que el usuario pueda buscar todos los productos de entrada según el nombre de la entidad.	Alta
RF8	Buscar productos por número de entrada.	El sistema permite que el usuario pueda buscar todos los productos de entrada según el número de entrada.	Alta
RF9	Buscar productos de entrada por código.	El sistema permite que el usuario pueda buscar todos los productos de entrada según el código de entrada del mismo.	Alta
RF10	Buscar productos de entrada por descripción.	El sistema permite que el usuario pueda buscar todos los productos de entrada según la descripción.	Alta
RF11	Buscar productos de entrada por nombre del almacén.	El sistema permite que el usuario pueda obtener todos los productos de entrada según el nombre del almacén	Alta
RF12	Buscar productos de entradas por estado	El sistema permite que el usuario pueda buscar todos los productos de entrada según el estado.	Alta
RF13	Exportar informe de inventario	El sistema permite que el usuario pueda exportar en un documento Excel el inventario de los productos ya sean de entrada, salidas o las ventas	Alta
RF14	Listar productos en existencia	El sistema permite que el usuario pueda obtener todos los productos que existen en los almacenes.	Alta
RF15	Buscar productos en existencia por código de almacén	El sistema permite que el usuario pueda buscar todos los productos que se encuentran en existencia según el código del almacén.	Alta
RF16	Buscar productos en existencia por nombre de almacén.	El sistema permite que el usuario pueda buscar todos los productos que se encuentran en existencia según la ubicación del almacén.	Alta
RF17	Buscar productos en existencia por código del producto.	El sistema permite que el usuario pueda buscar todos los productos que se encuentran en existencia según el código del producto.	Alta
RF18	Buscar productos en existencia por descripción del producto.	El sistema permite que el usuario pueda buscar todos los productos que se encuentran en existencia según la descripción del producto.	Alta

RF19	Listar productos en venta.	El sistema permite que el usuario pueda obtener todos los productos que están en venta en los almacenes.	Alta
RF20	Buscar productos en venta por código del almacén.	El sistema permite que el usuario pueda buscar todos los productos que se encuentran en venta según el código del almacén.	Alta
RF21	Buscar productos en venta por nombre de almacén	El sistema permite que el usuario pueda buscar todos los productos que se encuentran en venta según el nombre del almacén.	Alta
RF22	Buscar productos en venta por número de factura	El sistema permite que el usuario pueda buscar todos los productos que se encuentran en venta según el número de factura.	Alta
RF23	Buscar productos en venta por código de entidad	El sistema permite que el usuario pueda buscar todos los productos que se encuentran en venta según el código de entidad.	Alta
RF24	Buscar productos en venta por nombre de entidad.	El sistema permite que el usuario pueda buscar todos los productos que se encuentran en venta según el nombre de entidad.	Alta
RF25	Buscar productos en venta por código de productos	El sistema permite que el usuario pueda buscar todos los productos que se encuentran en venta según el código del producto.	Alta
RF26	Buscar productos en venta por descripción del producto.	El sistema permite que el usuario pueda buscar todos los productos que se encuentran en venta según la descripción del producto.	Alta
RF27	Buscar productos en venta por estado	El sistema permite que el usuario pueda buscar todos los productos que se encuentran en venta según el estado.	Alta
RF28	Listar productos faltantes	El sistema permite que el usuario pueda obtener todos los productos que faltan en los almacenes.	Alta
RF29	Buscar productos faltantes por nombre del almacén	El sistema permite que el usuario pueda buscar todos los productos que faltan en cada almacén.	Alta
RF30	Generar reporte de productos faltantes en el almacén	El sistema permite que el usuario tenga conocimiento de que productos se encuentran agotados en el almacén.	Alta

## 2.2.2 Requisitos No Funcionales.

Los requisitos no funcionales (RNF), son requisitos que imponen restricciones en el diseño o la implementación como restricciones en el diseño o estándares de calidad. Son propiedades o cualidades que el producto debe tener (PRESSMAN,2010).

Según la norma ISO/IEC 25010 el modelo de calidad representa la piedra angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto. En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar las propiedades de un producto software determinado. La calidad del producto software se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios aportando de esta manera un valor.

### **RNF1. Restricciones de Diseño:**

**RNF1.1.** La herramienta gestora de base de datos a utilizar es SQL Server 2017.

### **RNF2. Interfaz:**

**RNF2.1.** Los colores a utilizar serán blanco, negro y verde.

**RNF2.2.** El sistema debe incluir el nombre y logo de la empresa.

### **RNF3. Seguridad:**

La seguridad es la capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos.

**Confidencialidad:** Es la capacidad de protección contra el acceso de datos e información no autorizados, ya sea accidental o deliberadamente

**RNF3.1.** El control de acceso al sistema se establecerá por roles que se les asignaran a los usuarios que interactúen con el sistema.

### **RNF4. Eficiencia:**

**RNF4.1.** El tiempo de respuesta promedio de las peticiones que se realizan al servidor no deben exceder los 30 segundos.

### **RNF5. Disponibilidad:**

**RNF5.1.** A los usuarios autorizados se les deberá garantizar el acceso a la información solicitada en todo momento

### **RNF6. Usabilidad:**

Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones.

Esta característica se subdivide a su vez en las siguientes sub características:

**RNF6.1.** Capacidad para reconocer su adecuación: El sistema debe permitir que el usuario pueda obtener el inventario de los productos que se encuentran en los almacenes.

**RNF6.2.** Capacidad de aprendizaje: El sistema debe poseer una interfaz de usuario fácil e intuitiva.

### 2.2.3 Modelado de los requisitos.

Según la metodología SCRUM, las Historias de usuario son unas de las variantes que permiten encapsular los requisitos funcionales del sistema.

#### Historias de Usuario

Las historias de usuario se usan, en el contexto de la ingeniería de requisitos ágil, como una herramienta de comunicación que combina las fortalezas de ambos medios: escrito y verbal. Describen, en una o dos frases, una funcionalidad de software desde el punto de vista del usuario, con el lenguaje que éste emplearía (MEZINSKY,2018).

#### Elementos de la Historia de Usuario:

- **Número:** Indica el id del requisito funcional.
- **Usuario:** Indica que usuario utilizara este requisito.
- **Nombre de Historia:** Nombre del requisito funcional.
- **Prioridad en Negocio:** Representa que requisitos deben implementarse primero. La prioridad se establece como: alta, media o baja, según la importancia de este requisito para el negocio.
- **Riesgo en Desarrollo:** Evidencia el nivel de riesgo en caso de no realizarse la HU. El riesgo se establece como medio, alto o bajo.
- **Puntos Estimados:** Este atributo no es más que una estimación hecha por el equipo de desarrollo del tiempo de duración de la HU, 1 punto equivale a 1 día de programación.
- **Programador Responsable:** Persona encargada de implementar el requisito.
- **Iteración Asignada:** Determina en que sprint se comienza a implementar el requisito según su prioridad.
- **Descripción:** Describe de forma breve que espera el usuario de esta funcionalidad.
- **Validación:** Confirma y valida que el requisito cumple las expectativas del usuario.

A continuación, se muestran algunas de las historias de usuarios más significativas:

**Tabla 3.** Historia de usuario RF11. Fuente: Elaboración propia.

Historia de Usuario	
<b>Número:</b> 11	<b>Usuario:</b> Cliente
<b>Nombre de Historia:</b> Buscar producto de entrada por nombre de almacén	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo en Desarrollo:</b> Alto
<b>Puntos estimados:</b> 0.5	<b>Iteración Asignada:</b> 1
<b>Programador Responsable:</b> Arianna Pérez Rodríguez	
<b>Descripción:</b> El cliente desea buscar los suministros de entrada por el nombre del almacén para tener el control de esos productos.	

<p><b>Validación:</b> El cliente puede buscar el producto de entrada por el nombre del almacén.</p>
---

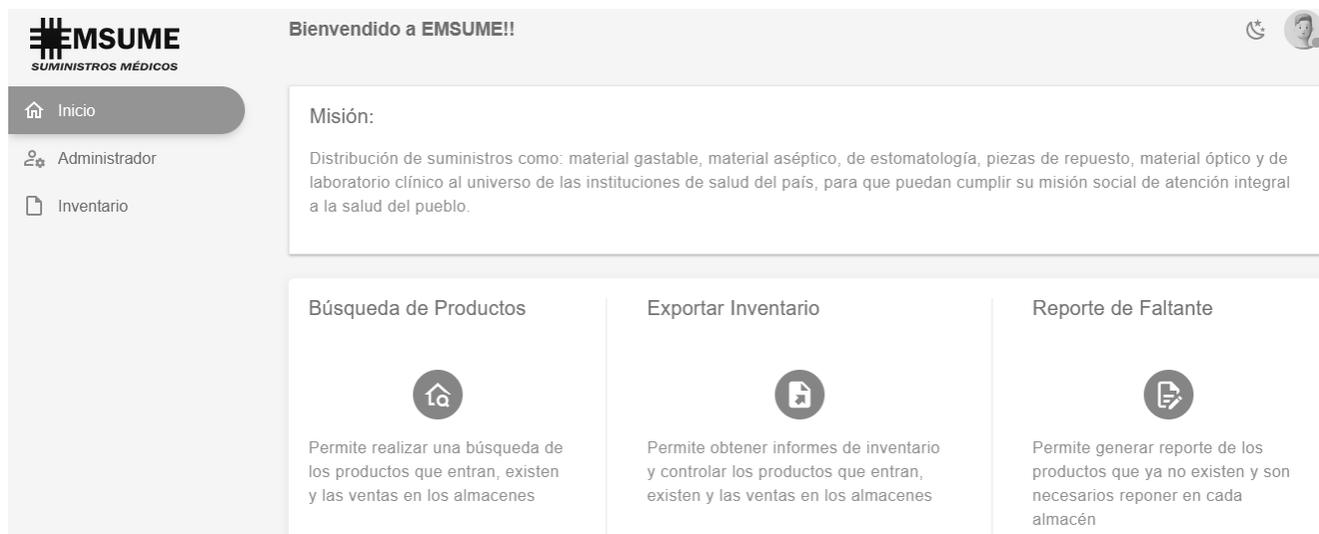
**Tabla 4.** Historia de usuario RF13. Fuente: Elaboración propia.

Historia de Usuario	
<b>Número:</b> 13	<b>Usuario:</b> Cliente
<b>Nombre de Historia:</b> Exportar Inventario.	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo en Desarrollo:</b> Alto
<b>Puntos estimados:</b> 0.5	<b>Iteración Asignada:</b> 2
<b>Programador Responsable:</b> Arianna Pérez Rodríguez	
<p><b>Descripción:</b> El cliente desea exportar el inventario de las entradas, existencias y ventas para tener el control de los suministros médicos que se encuentran en el almacén.</p>	
<p><b>Validación:</b> El cliente puede exportar el inventario de las entradas, existencias y ventas de los suministros médicos.</p>	

### 2.3 Diseño de la interfaz de usuario.

El diseño de la interfaz de usuario crea un medio eficaz de comunicación entre los seres humanos y la computadora. Siguiendo un conjunto de principios de diseño de la interfaz, el diseño identifica los objetos y acciones de ésta y luego crea una plantilla de pantalla que constituye la base del prototipo de la interfaz de usuario (PRESSMAN, 2010).

A continuación, se muestra el prototipo de interfaz de usuario que debe tener el sistema a implementar.



**Figura 2.** Prototipo de interfaz de usuario (Página de Inicio). Fuente: Elaboración propia.

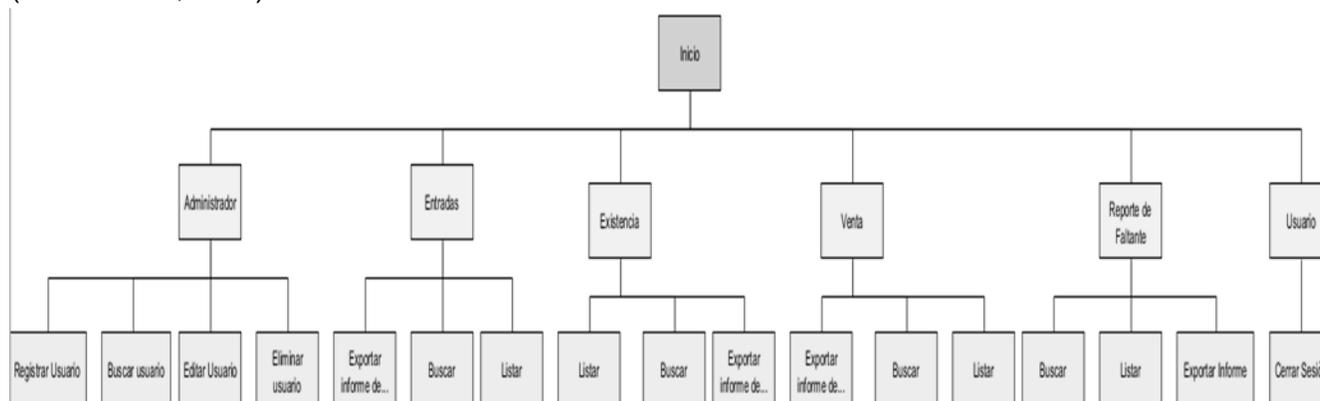


**Figura 3.** Prototipo de interfaz de usuario (Autenticar usuario). Fuente: Elaboración propia.

### 2.3.1 Mapa de Navegabilidad.

En el presente epígrafe, se modela el mapa de navegabilidad del sistema y se describen sus elementos. Proporcionan una representación esquemática de la estructura del hipertexto, indicando los principales conceptos incluidos en el espacio de la información y las interrelaciones que existen entre ellos. La organización jerárquica de los vínculos ofrece una

buena orientación a los usuarios pues la jerarquía es una forma de organización elemental (SANSANO,2017).



**Figura 4.** Mapa de navegabilidad. Fuente: Elaboración propia.

En el mapa de navegabilidad se contempla la estructura que tendrá el sistema a nivel de diseño. El sistema contará con una página de inicio que proporcionará información sobre las opciones del sistema (inventario de entrada, inventario de existencia, inventario de venta, reporte de faltante, administrador y cerrar sesión). La opción de administrador permite al administrador del sistema registrar, eliminar, editar datos del usuario y buscar los usuarios que se encuentran registrados. Las opciones inventario de entrada, existencia, ventas y reporte de faltante permite exportar un documento Excel con el inventario de los productos. La opción cerrar sesión permite al usuario salir del sistema.

## 2.4 Arquitectura especializada.

A continuación, se describe el concepto de arquitectura de software, se representan los patrones arquitectónicos, el estilo arquitectónico y los patrones de diseño a utilizar en la implementación del sistema.

### Arquitectura de Software:

La arquitectura de software se define como el diseño estratégico de una actividad relacionada con requisitos globales. Su solución se implementa en paradigmas de programación, estilos arquitectónicos, estándares de ingeniería de software basados en componentes, patrones arquitectónicos, seguridad, escala, integración y regularidades regidas por leyes (JAISWAL,2019).

Es la estructura u organización de los componentes del programa (módulos), la manera en que estos componentes interactúan, y la estructura de datos que utilizan los componentes (PRESSMAN,2008).

### 2.4.1 Patrón Arquitectónico.

Un patrón arquitectónico brinda la descripción de un problema particular y recurrente de diseño, que aparece en contextos de diseños específicos, y presenta un esquema genérico demostrado con éxito para su solución. El esquema de solución se especifica mediante la

descripción de los componentes que lo constituyen, sus responsabilidades desarrollos, así como también la forma como estos colaboran entre sí. El patrón arquitectónico es quien define la estructura básica de la aplicación (EBENEZER,2021).

#### **2.4.2 Estilo Arquitectónico.**

Un estilo arquitectónico es una lista de tipos de componentes que describen los patrones o las interacciones a través de estos. Un estilo influye en toda la arquitectura del software y puede combinarse en la propuesta de solución. Los estilos ayudan a un tratamiento estructural que concierne más bien a la teoría, la investigación académica y la arquitectura en el nivel de abstracción más elevado, expresando la arquitectura en un sentido más formal y teórico (CAPILLA,2022).

Para la solución propuesta se determinó el empleo del estilo llamada y retorno, pues este permite obtener una estructura de programa fácil de modificar persiguiendo la escalabilidad del sistema. Según Pressman el empleo de este estilo posibilita además la comunicación, la coordinación y la cooperación entre los componentes y las restricciones que definen como se integran para conformar el sistema, así como los modelos semánticos que facilitan al diseñador el entendimiento de todas las partes del sistema, evitando que las variaciones realizadas a funcionalidades o componentes específicos afecten el funcionamiento general (PRESSMAN, 2008).

#### **2.4.3 Selección de la arquitectónico.**

A continuación, se presentan los patrones arquitectónicos seleccionados para el lado del cliente y el lado del servidor según el framework de desarrollo utilizado.

##### **Arquitectura para Front-end:**

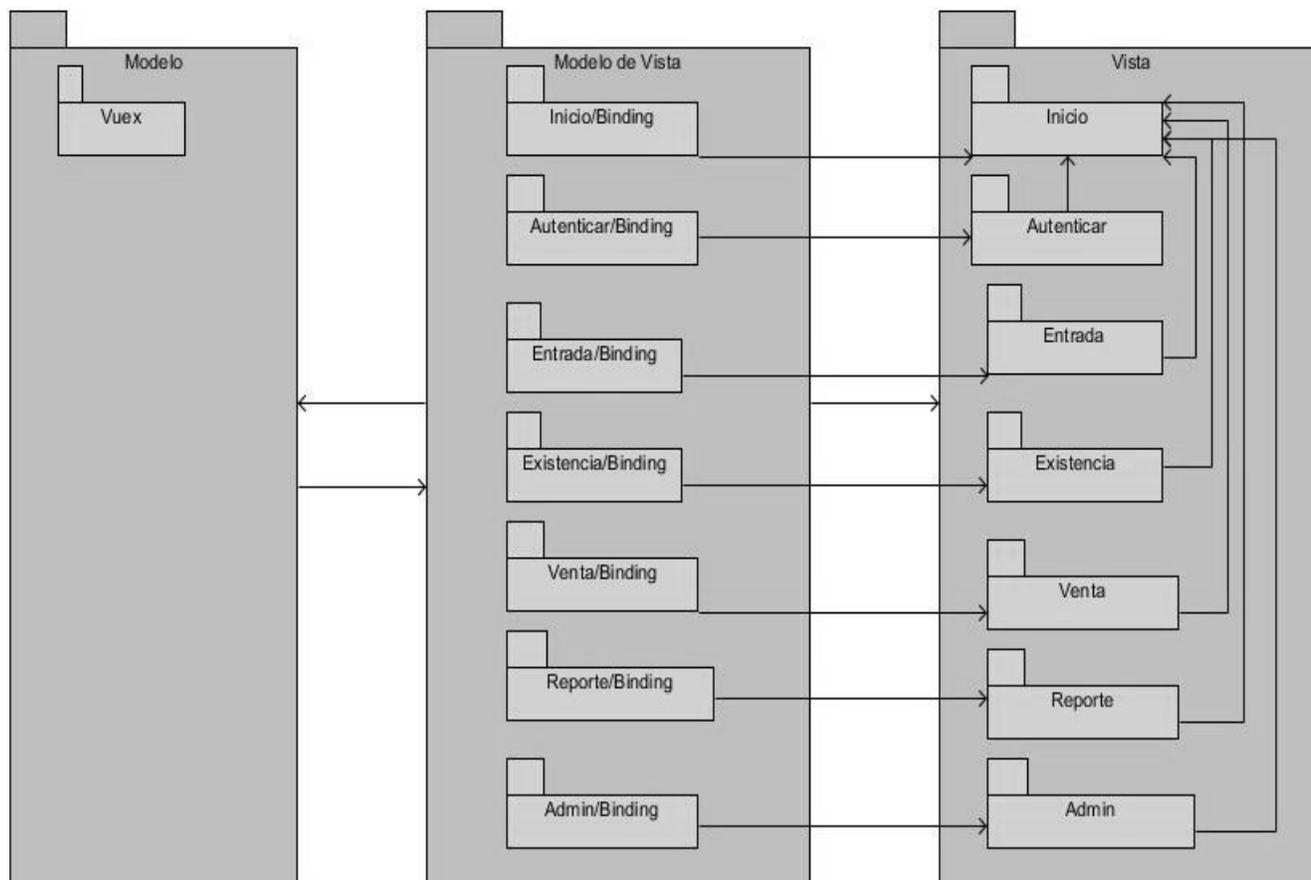
Para el lado del cliente se utilizó el framework VueJS, este trabaja bajo el patrón arquitectónico Modelo-Vista-Modelo de Vista (MVVM) (OLLILA,2022).

**Modelo:** El modelo, dentro de MVVM es el encargado de representar el modelo del negocio, proveyendo de esta manera la base necesaria para la manipulación de los datos de la aplicación.

**Vista:** La vista es la parte encargada de la parte visual de nuestra aplicación, no teniéndose que ocupar en ningún momento en el manejo de datos. En MVVM la vista tiene un rol activo, esto significa que en algún momento la vista recibirá o manejará algún evento y tendrá que comunicarse con el modelo, para poder cumplir el requerimiento.

**Modelo de Vista:** Es el encargado de ser la capa intermedia entre el modelo y la vista, procesando todas las peticiones que tenga la vista hacia el modelo, además de tener que ocuparse de manejar las reglas del negocio, la comunicación con aplicaciones externas.

A continuación, se modelará este patrón con la solución propuesta.



**Figura 5.** Modelado del patrón arquitectónico del Front-End. Fuente: Elaboración propia.

**Paquetes del Modelo:**

En el modelo se representa los datos almacenados del usuario en su inicio de sesión.

**Paquetes de la Vista:**

En la vista se representan los componentes jsx que se le muestran al usuario y mediante el cual, el usuario interactúa con el sistema.

**Paquetes del Modelo de Vista:**

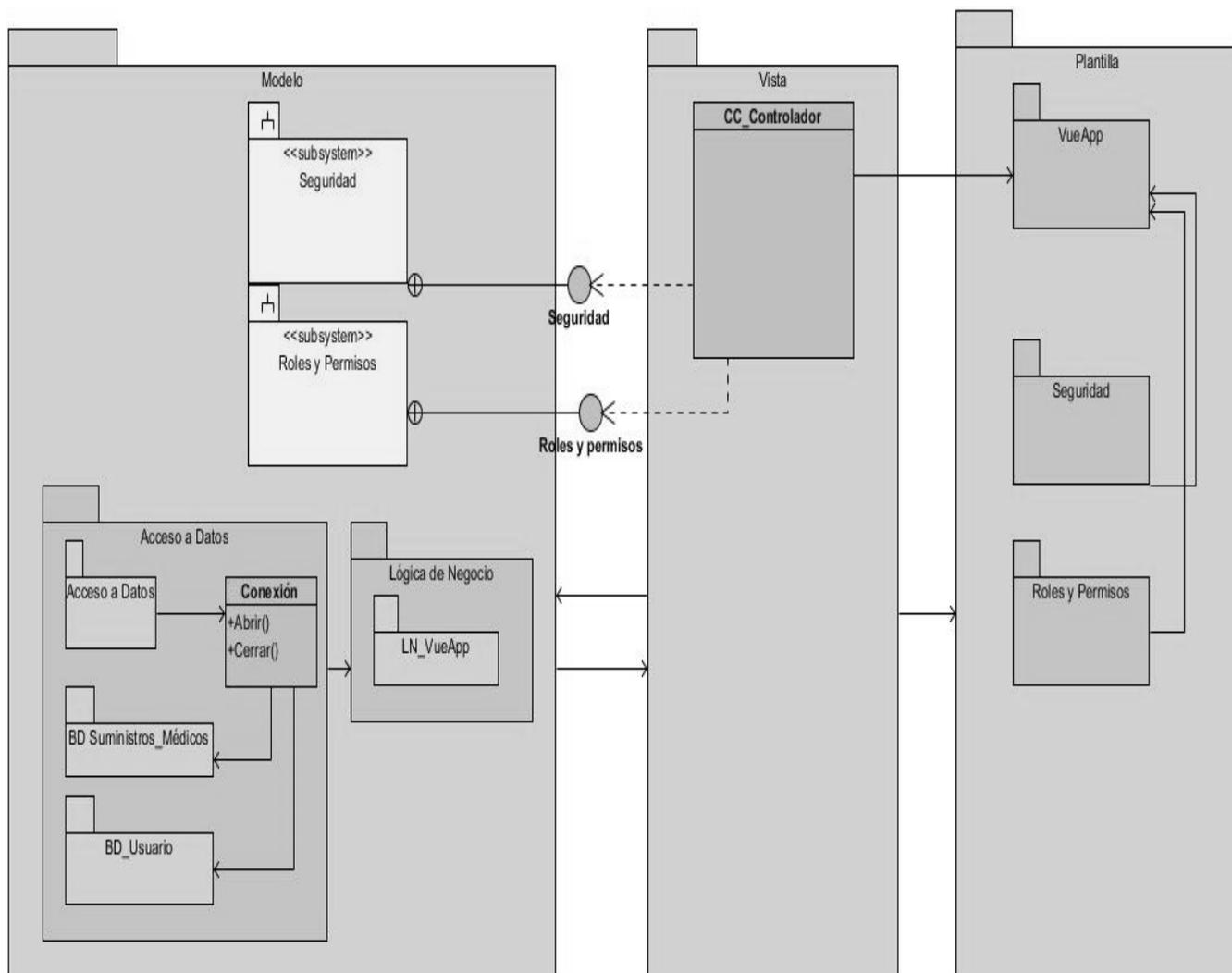
En el modelo de vista se representa la lógica del sistema que se aplica a cada componente.

**Arquitectura para Backend:**

El patrón arquitectónico utilizado para el modelado del sistema del lado del servidor, está basada en el Modelo Vista Plantilla (MVT del inglés: Model View Template), según define Django como framework de desarrollo. Esto posibilita la organización del sistema en paquetes individuales, sin dependencias entre el modelo y las plantillas (GORE, 2021).

**Modelo-Vista-Plantilla:**

A continuación, se presenta una breve descripción de esta arquitectura. Además, la misma se evidencia mediante el diagrama de paquetes del sistema, el cual se especifica más adelante (GORE, 2021):



**Figura 6.** Modelado del patrón arquitectónico del backend. Fuente: Elaboración propia.

- **M (Modelo)** es la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos; cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos.
- **T (Plantilla)** es la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación; cómo algunas cosas son mostradas sobre una página web u otro tipo de documento.
- **V (Vista)** es la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada. Es el puente entre el modelo y las plantillas.

**Paquetes y subsistemas del Modelo:**

**Paquete Subsistema:** Los paquetes subsistema representan clases y servicios que se consumirán directamente del framework de desarrollo. Se utilizarán los subsistemas roles y

permisos para controlar la accesibilidad al sistema y seguridad para garantizar la confiabilidad e integridad de los datos almacenados en la base de datos

**Paquete BD Suministros médicos:**

Es el encargado de guardar los datos y la información de los productos almacenados.

**Paquete BD Usuario:** Es el encargado de guardar los datos y la información de los usuarios que pueden acceder al sistema.

**Paquete Acceso de los Datos:** Es el encargado de gestionar las llamadas a datos y retorno de los mismos.

**Paquete Lógica del Negocio:** Es la que se encarga de definir la forma en que se extraen y se analizan los datos.

**Paquetes de la Vista:** En la vista se representa el paquete que controla las peticiones que son enviadas al modelo y la respuesta que se recibe del modelo la vista la envía a la plantilla (lado del cliente).

**Paquetes de la Plantilla:** En la plantilla se representa el paquete index.js que es el archivo donde se cargan los componentes de la aplicación cliente desarrollada con VueJS.

## **2.5 Diagrama de Clases del Diseño y Patrones del Diseño.**

A continuación, se presenta el diagrama de clases del diseño para el requisito funcional Buscar Productos de entrada por código de entrada, el diagrama se modelo basado en el patrón arquitectónico MVT.

Los diagramas de clases del diseño (DCD) son los más utilizados en el modelado de sistemas orientados a objetos. Estos muestran un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Se utilizan para modelar la vista de diseño estática de un sistema y se obtiene como resultado del refinamiento del modelo conceptual (JACOBSON, I. 2000).

Los diagramas de clases muestran cómo se lleva a cabo la colaboración entre las clases para dar cumplimiento a un requisito determinado. Para la elaboración de este diagrama se hace uso de estereotipos web que ayudan a representar de manera más fácil la función y el carácter de las clases dentro de la realización del requisito.

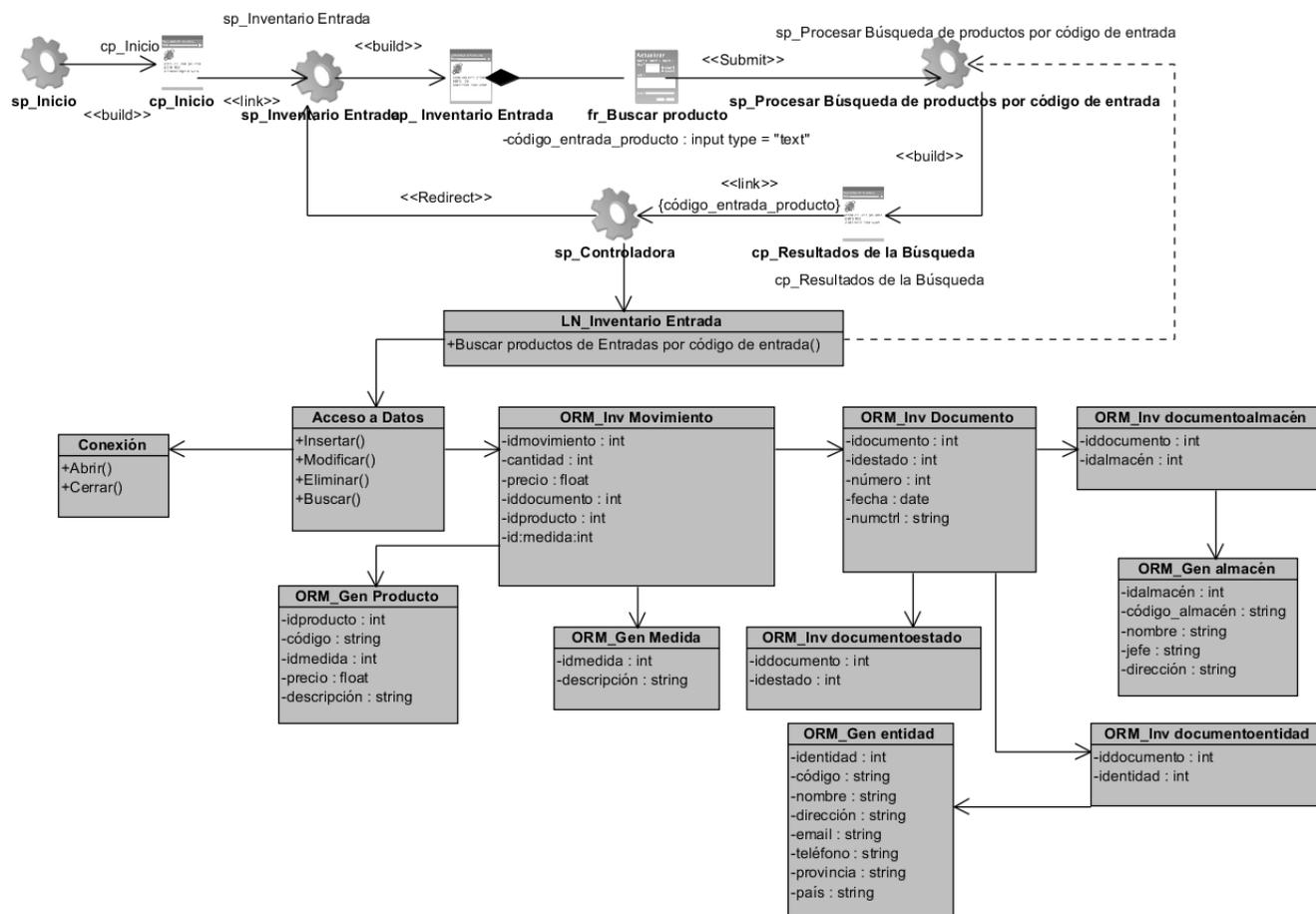


Figura 7. Diagrama de clases del Diseño. Buscar producto por código de entrada. Fuente: Elaboración propia.

### 2.5.1 Descripción de las clases del diagrama.

Tabla 5. Descripción del diagrama de clases del diseño. Fuente: Elaboración propia.

Nombre de la Clase	Descripción
Client Page	La página cliente es la encargada de mostrar los datos al usuario.
Server Page Constructora	La página server page Constructora se encarga de construirle al usuario, la página cliente con la que este desea interactuar.
ORM_InvMovimiento	Representa la entidad Movimiento con todos sus datos y las relaciones con otras clases ORM.
ORM_GenProducto	Representa la entidad GenProducto con todos sus datos y las relaciones con otras clases ORM.
ORM_InvDocumento	Representa la entidad que almacena los documentos con el inventario de los almacenes y las relaciones con otras clases ORM.

ORM_Invdocumentoalm	Representa la entidad inventario de los documentos de cada almacén con todos sus datos y las relaciones con otras clases ORM.
ORM_Invdocumentoestado	Representa la entidad inventario de los documentos de estado con todos sus datos y las relaciones con otras clases ORM.
ORM_Invdocumentoentidad	Representa la entidad inventario de los documentos de cada entidad con todos sus datos y las relaciones con otras clases ORM.
ORM_GenMedida	Representa la entidad GenMedida con todos sus datos y las relaciones con otras clases ORM.
ORM_GenAlmacén	Representa la entidad GenAlmacén con todos sus datos y las relaciones con otras clases ORM.
ORM_GenEntidad	Representa la entidad GenEntidad con todos sus datos y las relaciones con otras clases ORM.
LN_InventarioEntrada	Esta clase contiene, la opción de buscar y listar los productos que entran en los almacenes.
Acceso a Datos	Representa las acciones que se pueden realizar a la base de datos
Conexión	Esta clase contiene dos métodos, Abrir y Cerrar la conexión de la base de datos con el servidor.
Server Page Controladora	Clase encargada de manejar las peticiones de la página cliente.

## 2.6 Patrones de Diseño.

A continuación, se presentan los patrones de diseño utilizados: GRASP y GOF. Los patrones de diseño constituyen una descripción de un problema y la solución, la misma recibe un nombre y se puede aplicar a varios contextos. Muchos patrones proporcionan guías en el modo en que deberían asignarse las responsabilidades a los objetos, dada una categoría específica del problema (VARÓN, 2018).

### 2.6.1 Patrones GRASP.

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

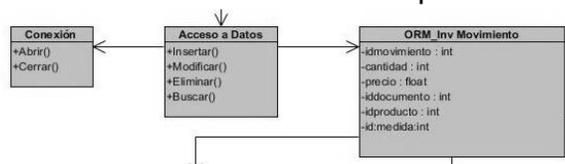
**Controlador:** Este patrón queda evidenciado en el sistema haciendo uso de las clases controladoras para atender las peticiones realizadas por el usuario. Cada una de ellas se encarga de procesar la petición de su entidad correspondiente de forma independiente.



**Figura 8.** Patrón Controlador. Fuente: Elaboración propia.

**Bajo acoplamiento:** Determina el nivel de dependencia de una clase con respecto a otras. Su uso potencia la reutilización, el mantenimiento y la mitigación de efectos a producirse en una clase al hacer cambios en otra.

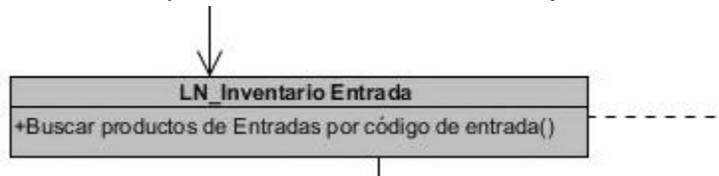
En la presente solución se pone de ejemplo la existencia de una baja dependencia en las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o la plantilla.



**Figura 9.** Patrón Bajo Acoplamiento. Fuente: Elaboración propia.

**Alta cohesión:** Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Significa que las clases del sistema tienen asignadas solo las responsabilidades que les corresponde y mantienen una estrecha relación con el resto de las clases.

Este patrón se evidencia en la propuesta de solución con las clases “Inventario Entrada” y “LN\_Inventario Entrada” pues las mismas muestran responsabilidades relacionadas coherentemente, que se complementan entre sí. Esta situación garantiza que exista un bajo acoplamiento, favoreciendo un equilibrio al interactuar los objetos.



**Figura 10.** Patrón Alta Cohesión. Fuente: Elaboración propia.

### 2.6.2 Patrones GOF.

Son soluciones técnicas. Indican resoluciones técnicas basadas en Programación Orientada a Objetos (POO).

**Estructural:** Son los que plantean las relaciones entre clases, las combinan y forman estructuras mayores. Tratan de conseguir que los cambios en los requisitos de la aplicación no ocasionen cambios en las relaciones entre los objetos. Lo fundamental son las relaciones de uso entre los objetos, y éstas están determinadas por las interfaces que soportan los objetos. Estudian cómo se relacionan los objetos en tiempo de ejecución. Sirven para diseñar las interconexiones entre los objetos. El patrón estructural que se utilizará es:

**Decorador (Decorator):** Este método pertenece a la clase abstracta View, padre de todas las vistas, que contienen un decorador para permitir agregar funcionalidades dinámicamente, se empleó para garantizar que los usuarios estén autenticados y puedan acceder a la información del sistema.

```

@login_required
class ReporteFaltante(ExportModelViewSet):
    queryset = InvExistenciaalm.objects.select_related('re_invexistenciaalm_idexistencia',
                                                       're_invexistenciaalm_idalmacen'
                                                       ).values('idexistencia__cantidad',
                                                            'idalmacen__nombre',
                                                            'idexistencia__idproducto__codigo',
                                                            'idexistencia__idproducto__descripcion'
                                                            ).filter(Q(idexistencia__cantidad__exact=0))
    serializer_class = ReporteSerializer

```

**Figura 11.** Patrón Decorator. Fuente: Elaboración propia.

**Comportamiento:** Plantea la interacción y cooperación entre las clases. Los patrones de comportamiento estudian las relaciones entre llamadas entre los diferentes objetos, normalmente ligados con la dimensión temporal.

**State (Estado):** Permite que un objeto modifique su comportamiento cada vez que cambie su estado interno. Este patrón se aplica en el componente Login cuando un usuario accede al sistema y cuando cierra sesión.

## 2.7 Modelo Entidad Relación.

El Modelo Entidad Relación (MER) proporciona una herramienta para representar información del mundo real a nivel conceptual. Permite describir las entidades involucradas en una base de datos, así como las relaciones y restricciones de estas (MENDOZA, 2018). Se empleó el uso de dos bases de datos porque en la base de datos de suministros médicos no se tiene permiso de escritura solo lectura de la información almacenada. El modelo de la base de datos que se presenta a continuación es el que se utilizó para la gestión de información de los productos que se encuentran en los almacenes de la empresa.

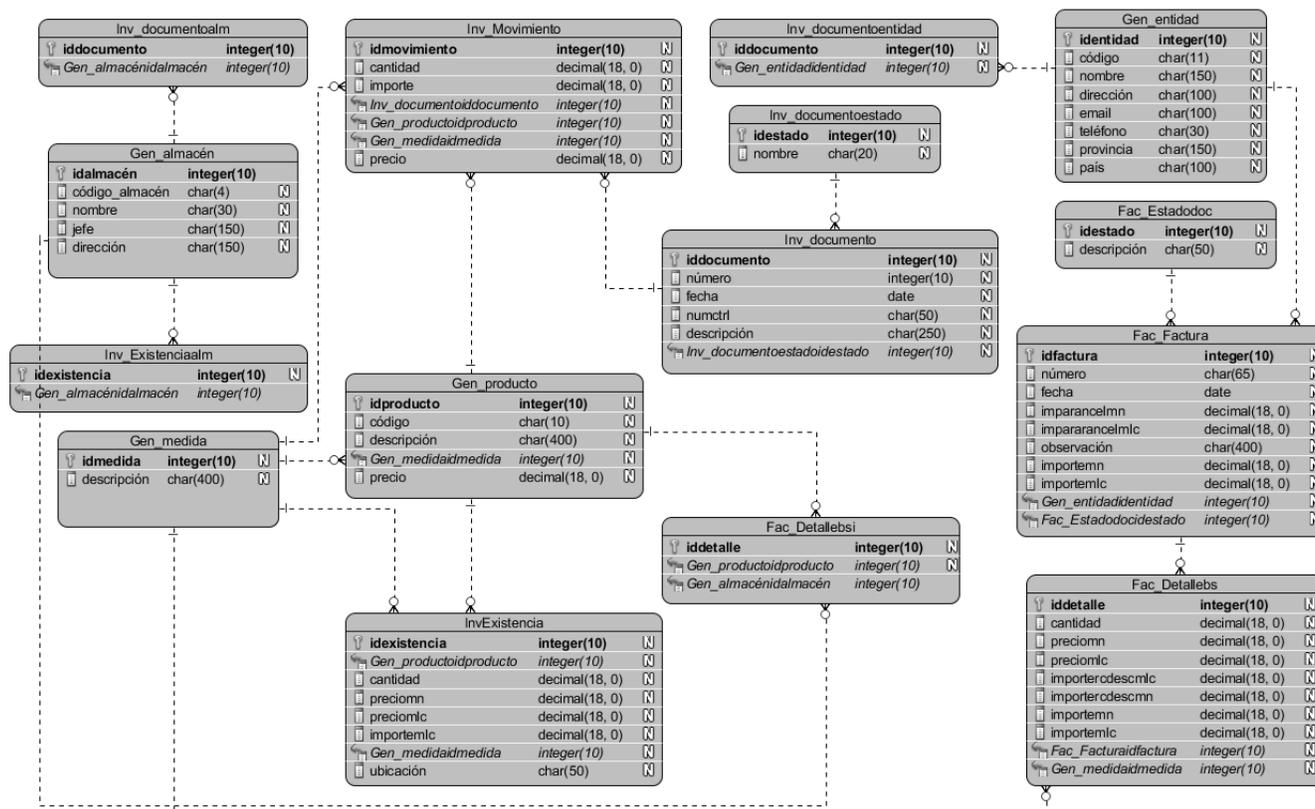


Figura 12. Modelo Entidad Relación Base de Datos Suministros Médicos. Fuente: Elaboración propia.

### 2.7.1 Descripción del Modelo de la Base de Datos Suministros Médicos.

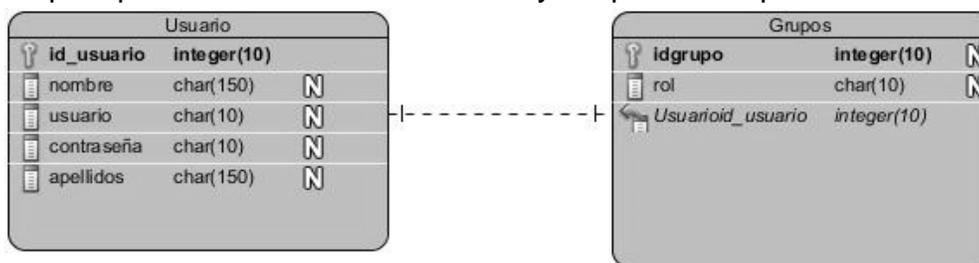
Tabla 6. Descripción de la entidad Gen Producto. Fuente: Elaboración propia.

Entidad: Gen Producto	Tipo de dato
idproducto: Representa la llave primaria de la entidad Gen Producto	integer
código: Representa el código del producto.	char
descripción: Esta columna contiene una breve descripción del producto.	char
precio: Esta columna contiene la información sobre el precio de cada producto.	decimal (18)
idmedida: Representa la llave primaria de la entidad Gen Medida, que pasa como llave foránea para la entidad Gen Producto por ser una relación de uno a muchos. (One to Many)	integer (10)

**Tabla 7.** Descripción de la entidad Gen Medida. Fuente: Elaboración propia.

Entidad: Gen Medida	Tipo de dato
Id medida: Representa la llave primaria de la entidad Gen Medida	integer (10)
descripción: Esta columna contiene una breve descripción sobre la cantidad de productos.	char (400)

A continuación, se muestra el modelo entidad relación de la base de datos usuario, la cual se emplea para tener control del acceso y los permisos que tendrá cada usuario según su rol.



**Figura 13.** Modelo Entidad Relación Base de Datos Usuario. Fuente: Elaboración propia.

**Tabla 8.** Descripción de la entidad Usuario. Fuente: Elaboración propia.

Entidad: Usuario	Tipo de dato
Id usuario: Representa la llave primaria de la entidad Usuario	integer (10)
nombre: Contiene el nombre del usuario.	char (150)
usuario: Contiene el nombre de usuario	char (10)
contraseña: Contiene la información sobre la contraseña del usuario.	char (10)
apellidos: Contiene el apellido del usuario	char (150)

**Tabla 9.** Descripción de la entidad Grupos.

<b>Entidad: Grupos</b>	<b>Tipo de dato</b>
Id grupo: Representa la llave primaria de la entidad Grupos	integer (10)
rol: Contiene el nombre del rol.	char (10)
Id usuario: Contiene la llave primaria de la entidad Usuario, que pasa como llave foránea para la entidad Grupos por ser una relación de uno a muchos (One to Many).	integer (10)

### **Conclusiones del capítulo**

La utilización de un modelo de negocio bien definido y la generación de los artefactos del mismo, permitieron conocer, el funcionamiento de todos los procesos que se llevan a cabo en la empresa EMSUME. La realización del análisis y diseño del sistema a implementar permitió obtener una representación visual de las principales relaciones entre los conceptos asociados a la problemática a través del modelo de dominio. Se identificaron 30 requisitos funcionales y 6 requisitos no funcionales que le aportan cualidades significativas al producto en cuanto a rendimiento, disponibilidad, seguridad, usabilidad, interfaz, entre otras. Para estructurar el sistema se utilizó el patrón arquitectónico MVVM y el patrón MVT este último se evidencia en la estructura del diagrama de clases de diseño. Se utilizaron los patrones de diseño GRASP: Controlador, Alta Cohesión y Bajo Acoplamiento, para la asignación de responsabilidades y los patrones GOF para la implementación del sistema. Los diseños de los modelos entidad relación representados evidencian las entidades de la base de datos y las relaciones entre las mismas.

### Capítulo 3: Implementación y validación del Sistema de gestión de información web para EMSUME.

En el presente capítulo se describen las actividades relacionadas con los procesos de implementación como: la planificación del sprint y la elaboración del diagrama de despliegue. Se realizan las pruebas al sistema web para validar su correcto funcionamiento, seguridad, así como también la satisfacción del cliente con cada incremento del producto de software que este reciba. Las pruebas realizadas son: integración con la aplicación cliente/servidor Postman, pruebas funcionales aplicando el método de caja negra a la interfaz de usuario y pruebas de aceptación aplicando la técnica de ladov.

#### 3.1 Implementación del sistema.

La fase de implementación constituye una parte transcendental en el proceso de desarrollo de un software, es en este momento donde se define y organiza el código de la propuesta de solución. Durante esta etapa se materializan, en forma de código, todos los artefactos de implementación, descripciones y arquitectura propuestos en la fase de análisis y diseño; que permite conformar el producto final requerido por el cliente.

##### 3.1.1 Planificación del Sprint Backlog.

Es la lista de tareas que elabora el equipo durante la planificación de un sprint. Se asignan las tareas a cada persona y el tiempo que debe realizar dicha tarea. De esta forma el proyecto queda descompuesto en unidades más pequeñas pudiendo determinarse en que tareas se ha avanzado. Pueden existir dependencias entre tareas, es importante diferenciarlas. Todas las tareas deben tener un coste semejante entre 15 y 30 días (TRIGÁS,2012).



**Figura 14.** Planificación del Sprint. Fuente: Elaboración propia.

**Tabla 10.** Planificación del Sprint 0. Fuente: Elaboración propia.

ID	Tarea	Responsable	Estado	Fecha inicial	Fecha final
T1	Montaje del ambiente de desarrollo con la integración de los marcos de trabajo seleccionados para el desarrollo.	Arianna Pérez Rodríguez	Resuelta	1/08/2022	15/08/2022
T2	Realizar el mapeo de la base de datos.	Arianna Pérez Rodríguez	Resuelta	1/08/2022	15/08/2022
RF6	Implementar funcionalidad: Listar productos de entrada	Arianna Pérez Rodríguez	Resuelta	1/08/2022	15/08/2022
RF7	Implementar funcionalidad: Buscar producto de entrada por nombre de entidad	Arianna Pérez Rodríguez	Resuelta	1/08/2022	15/08/2022
RF8	Implementar funcionalidad: Buscar producto por número de entrada	Arianna Pérez Rodríguez	Resuelta	1/08/2022	15/08/2022
RF9	Implementar funcionalidad: Buscar producto de entrada por código.	Arianna Pérez Rodríguez	Resuelta	1/08/2022	15/08/2022
RF10	Implementar funcionalidad: Buscar producto de entrada por descripción.	Arianna Pérez Rodríguez	Resuelta	1/08/2022	15/08/2022

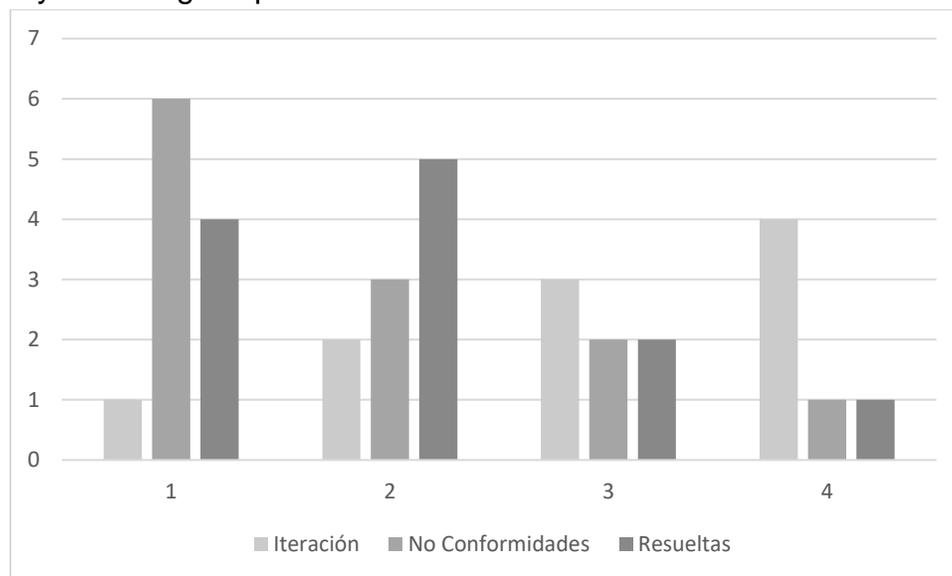
**Tabla 11.** Planificación del Sprint 1. Fuente: Elaboración propia.

ID	Tarea	Responsable	Estado	Fecha inicial	Fecha final
RF11	Implementar funcionalidad: Buscar producto de entrada por nombre del almacén.	Arianna Pérez Rodríguez	Resuelta	1/08/2022	15/08/2022
RF12	Implementar funcionalidad: Buscar producto de entradas por estado.	Arianna Pérez Rodríguez	Resuelta	1/08/2022	15/08/2022
RF13	Implementar funcionalidad: Exportar Inventario	Arianna Pérez Rodríguez	Resuelta	16/08/2022	31/08/2022
RF14	Implementar funcionalidad: Listar productos en existencia	Arianna Pérez Rodríguez	Resuelta	16/08/2022	31/08/2022

RF15	Implementar funcionalidad: Buscar productos en existencia por código de almacén	Arianna Pérez Rodríguez	Resuelta	16/08/2022	31/08/2022
RF16	Implementar funcionalidad: Buscar producto en existencia por nombre de almacén	Arianna Pérez Rodríguez	Resuelta	16/08/2022	31/08/2022
RF17	Implementar funcionalidad: Buscar producto en existencia por código.	Arianna Pérez Rodríguez	Resuelta	16/08/2022	31/08/2022
RF18	Implementar funcionalidad: Buscar producto en existencia por descripción.	Arianna Pérez Rodríguez	Resuelta	16/08/2022	31/08/2022

### 3.1.2 Resultados de la revisión del Sprint.

A continuación, se presentan los resultados obtenidos con cada iteración del Sprint, se obtuvieron un total de 12 no conformidades, las cuales se distribuyen en 6 en la primera iteración de las cuales se solucionaron 4, 3 no conformidades en la segunda las cuales fueron solucionadas y se le dio solución en esta iteración a las otras 2 no conformidades del sprint anterior, en la tercera iteración se obtuvieron 2 no conformidades y también fueron resueltas por parte del equipo de desarrollo. En la última iteración se halló 1 no conformidad a la cual se le dio solución y se entregó el producto de software listo.



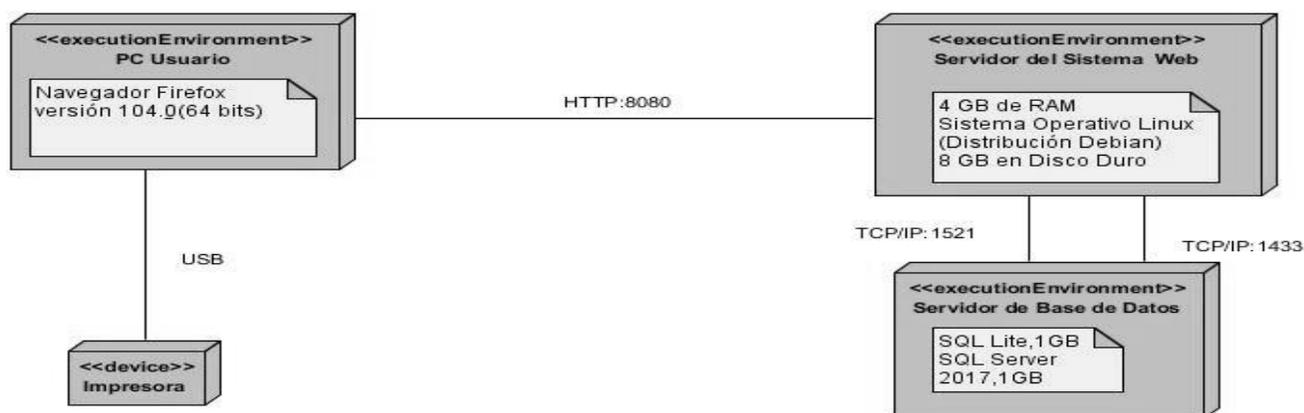
**Figura 15.** Resultados de la revisión del Sprint. Fuente: Elaboración propia.

### 3.1.3 Diagrama de Despliegue.

El diagrama de despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución, los vínculos de comunicación entre ellos y las instancias de los componentes y objetos que residen en ellos. Está compuesto por nodos, dispositivos y conectores. El propósito

del modelo de despliegue es capturar la configuración de los elementos de procesamiento y las conexiones entre estos elementos en el sistema. (SARMIENTO,2016).

- ✓ **PC Usuario:** Representa las computadoras clientes que se conectan al servidor de aplicaciones, las mismas se comunican con el servidor a través del protocolo seguro HTTP.
- ✓ **Servidor web:** Representa el servidor donde se encuentra instalada la aplicación web. Este accede al servidor de Base de Datos para el manejo de la información mediante el protocolo TCP/IP.
- ✓ **Servidor de Base de datos:** Es donde se almacena toda la información de la aplicación.
- ✓ **TCP/IP:** Protocolo para conectar el servidor de aplicaciones con las bases de datos.
- ✓ **USB:** Representa el cableado para conectar la impresora a la computadora del usuario.
- ✓ **HTTP:** Protocolo de transferencia utilizado para conectar la computadora del cliente con el servidor donde está el sistema.



**Figura 16.** Diagrama de Despliegue. Fuente: Elaboración propia.

### 3.1.4 Estándares de codificación.

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez.

Para facilitar el entendimiento del código y establecer un modelo a seguir, se establecieron los estándares de codificación mostrados a continuación para el lenguaje de programación Python en el framework de desarrollo Django (GUAGLIANO,2019).

- ✓ Las líneas podrán tener ochenta caracteres o menos.
- ✓ El método de indentación más popular en Python es con espacios.
- ✓ La declaración de importación debe escribirse en líneas separadas.
- ✓ La declaración de importación debe colocarse al principio del archivo, después de la descripción del módulo y la cadena de documentos, y antes de las variables globales.

- ✓ Los módulos deben tener un nombre corto y en minúscula.

```
from django_filters.rest_framework import DjangoFilterBackend
from rest_framework.viewsets import ModelViewSet
from myapp.api.serializer import *
from rest_framework import filters
from myapp.models import *
from myapp.api.mixin import ExportModelViewSet
from django.db.models import Q
```

*Figura 17. Importaciones en Django. Fuente: Elaboración propia.*

### 3.2 Pruebas de Software.

Las pruebas de software son el proceso de ejercitar el software con la intención de encontrar y corregir errores. Estas verifican que el software implemente una función específica de forma correcta y validan que las acciones que realiza respondan a los requisitos del cliente (PRESSMAN, 2005).

#### 3.2.1 Estrategia de Pruebas.

Estrategia de pruebas Una estrategia de pruebas de software proporciona una guía que describe los pasos que deben realizarse como parte de la prueba, cuándo se planean y se llevan a cabo dichos pasos, y cuánto esfuerzo, tiempo y recursos requieren. Por tanto, cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de casos de prueba, la ejecución de la prueba y la recolección y evaluación de los resultados (PRESSMAN, 2010).

Para la validación del sistema se propone realizar la siguiente estrategia de pruebas:

- **Pruebas funcionales:** Están basadas en ejecución, revisión y retroalimentación de las funcionalidades implementadas para el software y las especificaciones definidas por el usuario. Con estas pruebas se busca evaluar el sistema mediante modelos de pruebas. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos. Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro (PRESSMAN, 2010).
- **Prueba de integración:** Las pruebas de integración prueban la interacción entre dos o más elementos, que pueden ser clases, módulos, paquetes o subsistemas, entre otros; incluso la interacción del sistema con el entorno de producción. El objetivo de las pruebas de integración es verificar el correcto ensamblaje entre los distintos componentes una vez que han sido probados unitariamente con el fin de comprobar que interactúan correctamente a través de sus interfaces, tanto internas como externas,

cubren la funcionalidad establecida y se ajustan a los requisitos no funcionales especificados en las verificaciones correspondientes (SANCHEZ, 2015).

- **Pruebas de aceptación:** Las pruebas de aceptación son realizadas principalmente por los usuarios con el apoyo del equipo del proyecto. El propósito es confirmar que el sistema está terminado, que desarrolla puntualmente las necesidades de la organización y que es aceptado por los usuarios finales. Las pruebas de aceptación se centran en el comportamiento y capacidades de todo el sistema o producto en el entorno real. En este nivel es necesario considerar los siguientes elementos: (MARIN, 2020)
  - ✓ Se verifica la adecuación al uso del sistema por parte de usuarios de negocio.
  - ✓ Las pruebas se realizan utilizando el entorno del cliente.
  - ✓ El entorno de cliente puede reproducir nuevos fallos.

### 3.2.2 Aplicación de las pruebas.

A continuación, se muestra la aplicación de las pruebas que se le efectuaron a la solución:

#### **Pruebas Funcionales:**

En este tipo de pruebas se ejecutan los distintos servicios prestados con datos correctos e incorrectos. En caso de que los datos sean incorrectos se verifica que los mensajes de error sean los deseados y en el caso opuesto que los resultados sean los esperados (PRESS-MAN, 2010).

#### **Método de Prueba de Caja Negra:**

El método de caja negra se aplica a la interfaz del software, examina algún aspecto funcional de un sistema que tiene poca relación con la estructura lógica interna del software (PRESSMAN, 2010).

Se concentran en los requisitos funcionales del software, tratando de encontrar los siguientes errores (PRESSMAN, 2010):

- ✓ Funciones incorrectas o faltantes.
- ✓ Errores de interfaz.
- ✓ Errores de estructuras de datos o en acceso a bases de datos externas.
- ✓ Errores de comportamiento o desempeño.
- ✓ Errores de inicialización y término.

#### **Los casos de pruebas pretenden demostrar que:**

- ✓ Las funciones del software son operativas.
- ✓ La entrada se acepta de forma correcta.
- ✓ Se produce una salida correcta.
- ✓ La integridad de la información externa se mantiene.

Para el diseño de los casos de pruebas del método de Caja Negra se utilizó la Técnica de Partición Equivalente, la cual divide el dominio de entrada de un programa en clases de datos a partir de las cuales pueden derivarse casos de prueba. La partición equivalente se esfuerza por definir un caso de prueba que descubra ciertas clases de errores, reduciendo así el número total de casos de prueba que deben desarrollarse (PRESSMAN, 2010).

A continuación, se presenta el diseño de caso de prueba que se utilizará para comprobar el funcionamiento del sistema:

**Descripción de las Variables:**

**Tabla 12.** Diseño del Caso de prueba "Autenticar Usuario". Fuente: Elaboración propia.

No	Nombre de Campo	Clasificación	Valor Nulo	Descripción
1	Usuario	Campo de Texto	No	Contiene el nombre de usuario que desea acceder al sistema.
2	Contraseña	Campo de Contraseña	No	Contiene la contraseña del usuario.

**Tabla 13.** Diseño del Caso de prueba "Buscar por descripción". Fuente: Elaboración propia.

No	Nombre de Campo	Clasificación	Valor Nulo	Descripción
1	Buscar productos	Campo de Texto	No	Contiene el criterio de búsqueda.

**Descripción de Casos de Prueba**

Se evaluó y probó la validez de cada una de las entradas al sistema, utilizando datos válidos (V), inválidos (I).

**Tabla 14.** Técnica de Partición de Equivalencia del Caso de prueba "Autenticar Usuario". Fuente: Elaboración propia.

Escenario	Descripción	Usuario	Contraseña	Respuesta del sistema	Flujo Central
1.1 Insertar datos correctos	El usuario introduce sus datos correctamente	V	V	El sistema permite que el usuario acceda al sistema y le muestra en pantalla la página de inicio.	1-Se introducen los datos en los campos de texto. 2-Se selecciona el botón iniciar sesión.
		admin	admin		
1.2 Insertar datos inválidos	El usuario introduce datos erróneos.	V	I	El sistema muestra un mensaje al usuario un mensaje con los campos incorrectos.	1-Se introducen los datos en los campos de texto. 2-Se selecciona el botón iniciar sesión. 3-El sistema muestra un mensaje con el error.
		admin	admin		

1.3 Dejar campos vacíos	El usuario deja campos vacíos	I	I	El sistema muestra los campos vacíos de color rojo.	1-Se introducen los datos en los campos de texto. 2-Se selecciona el botón iniciar sesión. 3-El sistema muestra los campos vacíos de color rojo.
-------------------------	-------------------------------	---	---	---	--

**Tabla 15.** Técnica de partición de equivalencia caso de prueba "Buscar productos de entrada". Fuente: Elaboración propia.

Escenario	Descripción	Descripción del Producto	Respuesta del sistema	Flujo Central
1.1 Insertar datos correctos	El usuario introduce la descripción del producto correctamente.	V Nasobuco	El sistema muestra al usuario un listado con los productos que cumplen con ese criterio de búsqueda.	1-Selecciona la opción buscar productos. 2-Se introduce la descripción del producto.
1.2 Insertar datos inválidos	El usuario introduce la descripción del producto de forma incorrecta.	I Nasobco	El sistema notifica al usuario que no encontró ningún producto con esa descripción.	1-Selecciona la opción buscar productos. 2-Se introduce la descripción del producto. 3-Se muestra un mensaje que notifica que no se encontraron resultados

### Resultados de las pruebas utilizando el método de caja negra

En la iteración 4 se encontró una no conformidad asociada a la validación del formulario de inicio de sesión. Una vez realizadas las pruebas utilizando el método de caja negra mediante la técnica de partición equivalente usando el caso de prueba "Autenticar Usuario", se comprobaron las funcionalidades del sistema y la correcta validación de sus campos.

En el último sprint en el cual se entregó el producto completo al cliente para su validación y aceptación se solucionaron todas las no conformidades encontradas anteriormente en el plan de entregas.



preguntas cerradas que se intercalan dentro de un cuestionario y cuya relación el sujeto desconoce. Estas tres preguntas se relacionan a través de lo que se denomina el cuadro lógico de ladov (DE CASTRO,2020).

El número resultante de la interrelación de las tres preguntas, indica el nivel de satisfacción de los sujetos, que se expresa en la escala numérica que oscila entre 1 y -1.

	¿El sistema implementado cumple con todos los requerimientos funcionales?								
	Sí			No sé			No		
	¿Consideras que el sistema contribuye a mantener la seguridad de la información?								
¿La interfaz de usuario es de su agrado y le parece fácil e intuitiva?	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
No me gusta mucho	2	2	3	2	3	3	6	3	6
No me interesa	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No sé que decir	2	3	6	3	3	3	6	3	4

**Figura 19.** Cuadro Lógico Técnica de ladov. Fuente: Elaboración propia.

La escala de satisfacción es la siguiente:

1. Clara satisfacción.
2. Más satisfecho que insatisfecho.
3. No definida.
4. Más insatisfecho que satisfecho.
5. Clara insatisfacción.
6. Contradictoria.

**Tabla 16.** Niveles de satisfacción expresados en escala numérica. Fuente: Elaboración propia.

Niveles de satisfacción	Valor
Máximo de satisfacción	+1
Más satisfecho que insatisfecho	+0,5
No definido y contradictorio	0
Más insatisfecho que satisfecho	-0,5
Máxima insatisfacción	-1

**Tabla 17.** Resultados de la aplicación de la técnica ladov. Fuente: Elaboración propia.

Nivel de Satisfacción	Cantidad	Por ciento
Máximo de satisfacción	4	80
Más satisfecho que insatisfecho	1	20
No definido y contradictorio	0	0
Más insatisfecho que satisfecho	0	0
Máxima insatisfacción	0	0

**Índice de Satisfacción Grupal:**

$$ISG = \frac{A (+1) + B (+0,5) + C (0) + D (-0,5) + E (-1)}{N}$$

$$ISG = \frac{4 (+1) + 1 (+0,5) + 0 (0) + 0 (-0,5) + 0 (-1)}{5}$$

$$ISG = \frac{4.5}{5}$$

$$ISG = 0.90$$

A continuación, se presentan los resultados obtenidos tras aplicar la técnica de ladov para medir la satisfacción del usuario.



**Figura 20.** Resultados de la Prueba de Aceptación. Fuente: Elaboración propia.

Al procesar las respuestas a las encuestas en el cuadro lógico de ladov, se obtiene un grado de satisfacción grupal de 0,90, y a partir de graficar los resultados de la prueba de aceptación se obtuvo que el 80% de los usuarios muestra una clara satisfacción con el uso del sistema.

## **Conclusiones del capítulo**

En el presente capítulo se realizó la planificación del sprint Backlog, para determinar que funcionalidades se implementarían por sprint y el tiempo que se le asignaría a cada una de estas. Las descripciones del empleo de los estándares de codificación permitieron que fuera más fácil el entendimiento del código del programador y facilitar el mantenimiento futuro del sistema. Se cumplió con el objetivo planteado en el diseño teórico y se obtuvo como resultado un producto que cumple con los requerimientos del cliente. Las pruebas realizadas permitieron elevar la calidad de la solución obtenida y corroboraron el correcto funcionamiento del sistema.

## **CONCLUSIONES FINALES**

- ✓ El estudio de los referentes teóricos y el análisis de las diferentes herramientas y tendencias para la gestión de inventarios permitió determinar la no existencia de un sistema informático que responda a las necesidades requeridas por el cliente.
- ✓ El diseño de la propuesta de solución permitió generar los artefactos más significativos de acuerdo con la metodología de desarrollo de software Scrum, tomándose como referencia el product backlog.
- ✓ La implementación del sistema a través de las herramientas y lenguajes seleccionados permitió obtener un sistema web capaz de manejar datos referentes a los usuarios y los productos que se manejan en los almacenes de la empresa EMSUME.
- ✓ Las técnicas de validación aplicadas a la propuesta de solución permitieron la detección y corrección de las no conformidades detectadas y evidenciaron que el sistema constituye una solución funcional.
- ✓ La validación del problema de investigación mediante la carta de aceptación emitida por el cliente demostró que el: Sistema de Gestión de Información Web para los almacenes de suministros médicos distribuidos por EMSUME contribuye a informatizar el proceso de control de almacenes de la empresa.

## **RECOMENDACIONES**

Se recomienda añadir un módulo de clasificación de los productos almacenados para mejorar y apoyar el proceso de gestión de almacenes y extender el objetivo del sistema de gestión para obtener el control de todos los almacenes del país.

## REFERENCIAS BIBLIOGRFICAS

- [1] CRISTALDO PATRICIA y KLOSTER MIRIAM 2016. Sistemas de Gestión.
- [2] CASTRO RUZ, F. La idea esencial es acercar los servicios básicos a los ciudadanos. Discurso pronunciado el día 7 de abril en el Teatro Astral. 2003.
- [3]FALCÓN, Adriana López; SERPA, Gerardo Ramos. Acerca de los métodos teóricos y empíricos de investigación: significación para la investigación educativa. Revista Conrado, 2021, vol. 17, no S3, p. 22-31.
- [4]ARIAS GONZÁLES, José Luis. Técnicas e instrumentos de investigación científica. 2020.
- [5] BARTLE, Phil. Información para la gestión y gestión de la información. Consultado el, 2009, vol. 6.
- [6]SUÁREZ ALFONSO, A, CRUZ RODRÍGUEZ, I Y PÉREZ MACÍAS, Y. La gestión de la información: herramienta esencial para el desarrollo de habilidades en la comunidad estudiantil universitaria. Revista Universidad y Sociedad. 2015.
- [7]FLAMARIQUE, Sergi. Gestión de existencias en el almacén. Marge books, 2018.
- [8]ESPINAL, Alexander Alberto Correa; MONTOYA, Rodrigo Andrés Gómez; ARENAS, José Alejandro Cano. Gestión de almacenes y tecnologías de la información y comunicación (TIC). Estudios gerenciales, 2010, vol. 26, no 117, p. 145-171.
- [9] BRENES, Pedro. Técnicas de almacén. Editex, 2015.
- [10]DE GUEVARA, Miguel Ángel Ladrón. Gestión de inventarios. UF0476. Tutor formación, 2020.
- [11]FLORES, E. M. La gestión y los gestores de la información. Bibliodocencia. (2014).
- [12] FLAMARIQUE, Sergi. Manual de gestión de almacenes. Marge books, 2019.
- [13] FERNÁNDEZ, Vivian. La gestión de la información y las habilidades informacionales: binomio esencial en la formación universitaria. Red de Revistas Científicas de América Latina, el Caribe, España y Portugal Sistema de Información Científica. 2008.
- [14]SALAMANCA, Paola Irene Mayorga; GUTIÉRREZ, José Sánchez; URIBE, Elsa Georgina González. Efectos de las estrategias de la gestión del conocimiento en la competitividad de las pymes. Repositorio de la Red Internacional de Investigadores en Competitividad, 2015, vol. 9, no 1, p. 1728-1749.
- [15] CAIZA MUYOLEMA, Carlos David. Modelo de implementación de un sistema de planificación de recursos empresariales para pequeñas y medianas empresas, caso Unidad de Ar-tes Gráficas del Instituto Geográfico Militar, Quito-Ecuador. 2015. Tesis de Maestría. Quito: UCE.
- [16] ROMERO, Lorely Moya, et al. Funcionalidades y principales opciones del módulo de planificación del software integrado Versat Sarasola. Observatorio de la Economía Latinoamericana, 2009, no 117.

- [17] AGUILAR, Linda; GARCÍA, Galuth; MORALES, Lina. Software libre en las instituciones públicas. Atlante Cuadernos de Educación y Desarrollo, 2018, no abril.
- [18] MOLINA, Silvia Gabriela Rivadeneira. Metodologías ágiles enfocadas al modelado de requerimientos. Informes Científicos Técnicos-UNPA, 2013, vol. 5, no 1, p. 1-29.
- [19] AL-SAQQA, Samar; SAWALHA, Samer; ABDELNABI, Hiba. Agile Software Development: Methodologies and Trends. International Journal of Interactive Mobile Technologies, 2020, vol. 14, no 11.
- [20] KEN SCHWABER, J. S. The scrum guide. 2018.
- [21] LUNA, Fernando. JavaScript-Aprende a programar en el lenguaje de la web. RedUsers, 2019.
- [22] WALKER, Jonathan S. Python: La Guía Definitiva para Principiantes para Dominar Python. Babelcube Inc., 2018.
- [23] VINCENT, William S. Django for Beginners: Build websites with Python and Django. WelcomeToCode, 2021.
- [24] LÓPEZ, Esteban Saavedra. Framework para desarrollo de aplicaciones Web. 2008.
- [25] JETBRAINS, S. R. O. pyCharm: The Python IDE for Professional Developers. 2020.
- [26] VAINIKKA, Joel. Full-stack web development using Django REST framework and react. 2018.
- [27] RUBIO, Daniel. REST services with Django. En Beginning Django. Apress, Berkeley, CA, 2017. p. 549-566.)
- [28] NAVARRO MORALES, Juan José, et al. Estudio de la funcionalidad de Vue 3 en aplicaciones web. 2021.
- [29] NELSON, Brett. Why Vue. js? En Getting to Know Vue. js. Apress, Berkeley, CA, 2018. p. 1-8.
- [30] SOURD, Frank Pompa. XML, JSON y el intercambio de información. Anuario Ciencia en la UNAH, 2022, vol. 18, no 3.
- [31] RODRÍGUEZ, César; DORADO, Rubén. ¿Por qué implementar Scrum? Revista Ontare, 2015, vol. 3, no 1, p. 125-144.
- [32] BOTH, David. Apache web server. En Using and Administering Linux: Volume 3. Apress, Berkeley, CA, 2020. p. 215-234.
- [33] GARCÍA-HOLGADO, A., et al. UML. Unified Modeling Language. 2020.
- [34] PARADIGM, Visual. UML Class Diagram Tutorial. Visual Paradigm, 2020.
- [35] HERRERA, Dylan Javier Ruiz. Desarrollo de páginas web utilizando la tecnología React Ant Desing con librería de promesas Axios. Sinaloa: UPS, 2020).
- [36] BHOSALE, S. T.; PATIL, Tejaswini; PATIL, Pooja. Sqlite: Light database system. Int. J. Comput. Sci. Mob. Comput, 2015, vol. 44, no 4, p. 882-885.
- [37] SANTAMARÍA, José; HERNÁNDEZ, Javier. Microsoft SQL Server. SQL SER vs MY SQL, 2016, p. 1-6.

- [38] LARMAN, C. UML y patrones. Una introducción al análisis y diseño orientado a objetos. 2. S.l.: s.n. 2009.
- [39] SOMERVILLE, Ian. Ingeniería de Software. 9na. México: Addison Wesley: Pearson Education, Inc., 2011.
- [40] PRESSMAN, R. Ingeniería del Software Un Enfoque Práctico. 7ma ed. University of Connecticut. 2010.
- [41] ISO/IEC 25010. Systems and software engineering: software product quality and System quality in use models, 2011.
- [42] MENZINSKY, Alexander, et al. Historias de usuario. Ingeniería de requisitos ágil, 2018.
- [43] SANSANO MIRALLES, Héctor. Pruebas sobre sitios web. 2017.
- [44] JAISWAL, Manishaben. Software architecture and software design. International Research Journal of Engineering and Technology (IRJET) e-ISSN, 2019, p. 2395-0056.
- [45] PRESSMAN, R.S. Ingeniería del Software. Un enfoque práctico. Sexta Edición. Edition ed. 2008
- [46] CAPILLA SEVILLA, Rafael, et al. Temas Diseño y Arquitectura del Software. 2022.
- [47] OLLILA, Risto; MÄKITALO, Niko; MIKKONEN, Tommi. Modern Web Frameworks: A Comparison of Rendering Performance. Journal of Web Engineering, 2022.
- [48] EBENEZER, Olukunle; IRETI, Oyebode; OYERINDE, Mojisola. A REVIEW ON SOFTWARE ARCHITECTURAL PATTERNS. GSJ, 2021, vol. 9, no 8.
- [49] GORE, Himanshu, et al. Django: Web Development Simple & Fast. Annals of the Romanian Society for Cell Biology, 2021, vol. 25, no 6, p. 4576-4585.
- [50] JACOBSON. El Proceso Unificado de Desarrollo de Software. Madrid: Pearson Educación S.A. ISBN 8478290362. 2000
- [51] VARÓN, Ángel. Ingeniería de software I. 2018.
- [52] LARMAN, C. UML y Patrones. California: Prentice Hall. 2005.
- [53] MENDOZA, Abundio; LÓPEZ, Rosa. Bases de datos. 2018.
- [54] RAD, Nader K.; TURLEY, Frank. Los Fundamentos de Agile Scrum. Van Haren, 2019.
- [55] SARMIENTO, Johana. UML: Diagrama de despliegue. Visión general del diagrama de despliegue, 2016.
- [56] GUAGLIANO, Celeste. Programación en Python I: Entorno de Programación–Sintaxis–Estructuras de Control. RedUsers, 2019.
- [57] PRESSMAN, Roger S. Software engineering: a practitioner's approach. Palgrave Macmillan, 2005.
- [58] SANCHEZ PEÑO, José Manuel. Pruebas de software. fundamentos y técnicas. 2015.
- [59] MARIN DIAZ, Aymara; TRUJILLO CASAÑOLA, Yaimí; BUEDO HIDALGO, Denys. Estrategia de pruebas para organizaciones desarrolladoras de software. Revista Cubana de Ciencias Informáticas, 2020, vol. 14, no 3, p. 83-104.
- [60] DIMOSKI, Davor, et al. Testing RESTful APIs–Use Case: RESTful API for Solving Multidimensional Time–Independent Schrödinger Equation. 2022.

[61]DE CASTRO FABRE, Astrid Fernández; ORTEGA, Naydelin Sánchez; FARRAT, Yusniel Reyes. El proceso de validación mediante la Técnica de ladov en cursos por encuentros. Revista Ingeniería Agrícola, 2020, vol. 10, no 1, p. 66-70.

## ANEXOS

Entrevista a informáticos de la empresa EMSUME

1. ¿Nombre del Entrevistado?

Greidel Isaac Sagarra

2. ¿Cargo que ocupa en la empresa el entrevistado?

Especialista principal Informático

3. ¿Cómo está organizada la empresa?

4. ¿Cómo se realiza actualmente los informes de inventario en la empresa?

Mediante consultas SQL que se exportan a Excel.

5. ¿Con qué tecnologías de base de datos trabaja la empresa?

La empresa trabaja con una base de datos que usa la tecnología SQL Server 2017

6. ¿Qué tablas de la base de datos son las que se necesitan utilizar?

**Las tablas a utilizar serán:**

Gen\_entidad

Gen\_almacén

Gen\_producto

Gen\_medida

Fac\_DetalleBSI

Fac\_Factura

Fac\_Estadodoc

Fac\_DetalleBS

Inv\_Movimiento

Inv\_Documento

Inv\_documentoestado

Inv\_documentoentidad

Inv\_documentoalm

Inv\_Existencia

Inv\_Existenciaalm

**Figura 21.** Entrevista realizada para desarrollar la investigación.

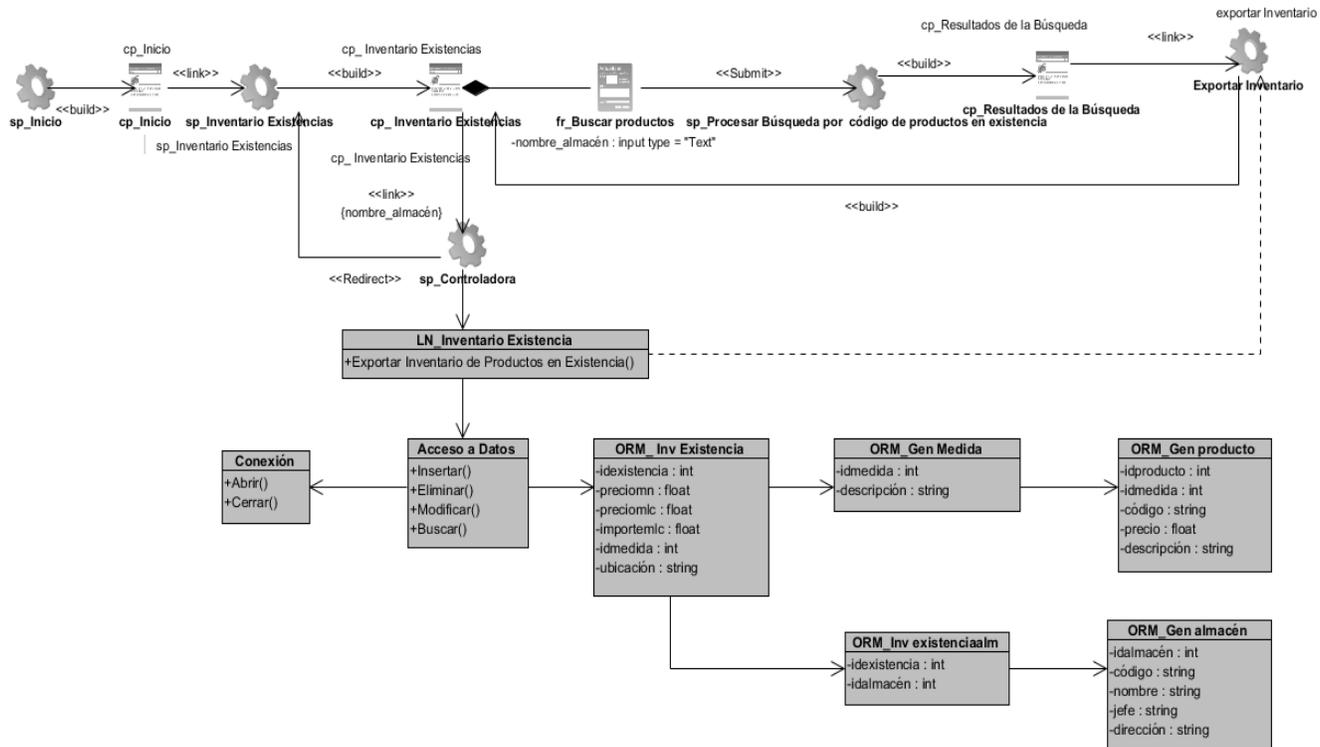


Figura 22. Diagrama de Clases del Diseño. RF Exportar Inventario. Fuente: Elaboración propia.

Historia de Usuario	
Número:1,2,3,4	Usuario: Administrador
Nombre de Historia: Gestionar Usuario	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Medio
Puntos estimados: 2	Iteración Asignada:4
Programador Responsable: Arianna Pérez Rodríguez	
Descripción: El administrador del sistema tiene permiso para registrar , eliminar, buscar y editar los usuarios que necesiten acceder a la información del sistema.	
Validación: El administrador puede realizar todas las funcionalidades acordadas.	

Figura 23. Historia de Usuario. Gestionar Usuario. Fuente: Elaboración propia.

Historia de Usuario	
Número:30	Usuario: Cliente
Nombre de Historia: Generar reporte de productos faltantes.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 0.5	Iteración Asignada:4
Programador Responsable: Arianna Pérez Rodríguez	
Descripción: El cliente desea obtener un informe con los productos que necesito reponer en el almacén.	
Validación: El cliente puede obtener el reporte de la cantidad de productos que están agotados.	

**Figura 24.** Historia de Usuario. Generar Reporte de productos faltantes. Fuente: Elaboración propia.

**Tabla 18.** Planificación del Sprint 2

ID	Tarea	Responsable	Estado	Fecha inicial	Fecha final
RF19	Implementar funcionalidad: Listar productos en venta.	Arianna Pérez Rodríguez	Resuelta	1/09/2022	15/09/2022
RF20	Implementar funcionalidad: Buscar productos en venta por código del almacén.	Arianna Pérez Rodríguez	Resuelta	1/09/2022	15/09/2022
RF21	Implementar funcionalidad: Buscar productos en venta por nombre de almacén.	Arianna Pérez Rodríguez	Resuelta	1/09/2022	15/09/2022
RF22	Implementar funcionalidad: Buscar productos en venta por número de factura.	Arianna Pérez Rodríguez	Resuelta	1/09/2022	15/09/2022
RF23	Implementar funcionalidad: Buscar productos en venta por código de entidad.	Arianna Pérez Rodríguez	Resuelta	1/09/2022	15/09/2022
RF24	Implementar funcionalidad: Buscar productos en venta por nombre de entidad	Arianna Pérez Rodríguez	Resuelta	1/09/2022	15/09/2022

RF25	Implementar funcionalidad: Buscar productos en venta por código de productos.	Arianna Pérez Rodríguez	Resuelta	1/09/2022	15/09/2022
RF26	Implementar funcionalidad: Buscar productos en venta por descripción del producto.	Arianna Pérez Rodríguez	Resuelta	1/09/2022	15/09/2022

**Tabla 19. Planificación del Sprint 3**

ID	Tarea	Responsable	Estado	Fecha inicial	Fecha final
RF27	Implementar funcionalidad: Buscar productos en venta por estado.	Arianna Pérez Rodríguez	Resuelta	16/09/2022	1/10/2022
RF28	Implementar funcionalidad: Listar productos faltantes	Arianna Pérez Rodríguez	Resuelta	16/09/2022	1/10/2022
RF29	Implementar funcionalidad: Buscar producto faltante por nombre del almacén	Arianna Pérez Rodríguez	Resuelta	16/09/2022	1/10/2022
RF30	Implementar funcionalidad: Generar reporte de productos faltantes	Arianna Pérez Rodríguez	Resuelta	16/09/2022	1/10/2022
RF1	Implementar funcionalidad: Registrar Usuario	Arianna Pérez Rodríguez	Resuelta	16/09/2022	1/10/2022
RF2	Implementar funcionalidad: Editar Usuario	Arianna Pérez Rodríguez	Resuelta	16/09/2022	1/10/2022
RF3	Implementar funcionalidad: Eliminar Usuario	Arianna Pérez Rodríguez	Resuelta	16/09/2022	1/10/2022
RF4	Implementar funcionalidad: Buscar Usuario por nombre.	Arianna Pérez Rodríguez	Resuelta	16/09/2022	1/10/2022
RF5	Implementar funcionalidad: Autenticar Usuario	Arianna Pérez Rodríguez	Resuelta	16/09/2022	1/10/2022