PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS

DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS Y COMPUTACIÓN

ANÁLISIS, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA EL CONTROL DE INVENTARIOS Y GESTIÓN DE ORDENES DE TRABAJO, CASO DE ESTUDIO: EMPRESA DE SOLUCIONES GRÁFICAS V-IMAGEN DE LA CIUDAD DE SAN LORENZO

AUTOR:

Angelo Gabriel Benavidez Chuquilla

DIRECTOR

Henry Nelson Roa Marín, PhD.

QUITO, 2022

RESUMEN

El presente trabajo de disertación está orientado al desarrollo e implementación de un sistema web denominado “análisis, desarrollo e implementación de un sistema web para el control de inventarios y gestión de ordenes de trabajo, caso de estudio: empresa de soluciones gráficas v-imagen de la ciudad de san lorenzo", el cual fue desarrollado usando una variación en la pila de desarrollo conocida como MERN (Mongo, Express, React.JS, Node.JS). Para el presente trabajo de disertación se reemplaza R de React.js por N de Next.JS el cual es un framework para aplicaciones de ambiente en productivo de React, conformando la pila de desarrollo MENN.

La metodología usada para el desarrollo fue Scrum, iniciando con la elaboración de historias de usuarios, las cuales fueron obtenidas por medio de comunicación activa con el cliente.

Una vez obtenido los requerimientos se procede a construir un modelo de base de datos, así como un borrador de las interfaces a desarrollar, permitiendo tener una estructura clara de lo que se requiere construir, las historias fueron divididas en tres sprints según su complejidad, en cada sprint se desarrollaron historias obteniendo partes del producto hasta obtener el producto completo, siendo éste sometido a pruebas unitarias y de integración con el objetivo de corregir fallos y tener un producto totalmente funcional.

Finalmente, como resultado del presente trabajo, se entrega un sistema web que permita el control de inventarios y gestión de ordenes de trabajo, desplegado en la nube y que permite tener acceso desde cualquier dispositivo con un navegador y conexión a Internet.

DEDICATORIA

El presente trabajo de disertación está dedicado a:

Mi madre Jacqueline quién ha sido un pilar fundamental en mi vida, quien con su amor, esfuerzo y motivación constante me ha permitido alcanzar este logro, a mi abuela Beatriz por sus consejos y educación que me ha brindado.

A mis familiares que siempre han estado pendientes de mi crecimiento personal y profesional.

AGRADECIMIENTO

Quiero en principio agradecer a Dios por permitirme culminar una etapa muy linda de mi vida, por sus bendiciones en todo momento de mi carrera estudiantil.

A mi familia por brindarme su apoyo, consejos y motivación.

A la Pontificia Universidad Católica del Ecuador, a la Facultad de Ingeniería y todos los docentes que la conforman, quienes han compartido conocimientos y experiencias. A mi director, Dr. Henry Roa, quien con sus consejos y observaciones me ha permitido culminar el presente trabajo de disertación.

A la empresa V-Imagen por permitirme llevar a cabo mi trabajo de disertación y proveerme de la información necesaria para la implementación del sistema web.

A mis compañeros y amigos quienes han compartido conmigo durante estos años, siendo parte importante de mi vida.

Contenido

RESUMEN I
DEDICATORIA II
AGRADECIMIENTO III
1 CAPÍTULO I: INTRODUCCIÓN 1
1.1 Justificación 1
1.2 Planteamiento del Problema 2
1.3 Objetivos 3
1.3.1 Objetivo General 3
1.3.2 Objetivos Específicos 3
1.4 Datos del Negocio: Caso de Estudio “V-Imagen” 3
1.4.1 Misión 3
1.4.2 Visión 4
2 CAPÍTULO II: FUNDAMENTOS TEORICOS 4
2.1 Inventarios 4
2.1.1 Definición 4
2.1.2 Importancia de la organización y administración de un inventario 5
2.1.3 Tipos de Inventarios 6
2.1.4 Soluciones Open Source para el Control de Inventarios 7
2.1.5 Ordenes de Trabajo 8
2.2 Metodología de Desarrollo 8
2.2.1 Metodologías de Desarrollo Ágil 8
2.2.2 Scrum 10
2.3 Aplicaciones Web 14
2.3.1 Definición 14
2.3.2 Lenguajes de programación orientados a la web 14
2.4 Arquitectura de Software 16
2.4.1 Estilos arquitectónicos 16
2.5 Servicios Web 27
2.5.1 Servicios Web SOAP 27
2.5.2 Servicios Web RESTFUL 28
2.5.3 GRAPHQL 28
2.6 Frameworks de Desarrollo 29
2.7 Base De Datos 30

2.7.1 Tipos de Bases de Datos 31
2.8 Herramientas 33
2.8.1 Backend 33
2.8.2 Frontend 35
2.8.3 Base de Datos 36
2.8.4 Pruebas 37
2.8.5 Despliegue 37
2.8.6 Control de cambios y manejo de versiones 39
2.9 Justificación de Herramientas Seleccionadas 40
3 CAPÍTULO III: ANÁLISIS Y LEVANTAMIENTO DE REQUERIMIENTOS 42
3.1 Historias de Usuario 42
3.2 Product Backlog 58
3.2.1 Características del Product Backlog 58
3.3 Formalización de Requerimientos 71
3.3.1 Requerimientos Funcionales y No Funcionales 71
3.4 Descripción del Sistema 73
3.4.1 Descripción General 73
3.4.2 Funcionalidades Generales del Sistema 74
3.4.3 Funcionalidades enfocadas al Administrador 74
3.4.4 Funcionalidades enfocadas al Operador 75
4 CAPÍTULO IV: DISEÑO 76
4.1 Casos de Uso 76
4.2 Arquitectura del sistema 85
4.3 Base de Datos 85
5 CAPÍTULO V: DESARROLLO Y PRUEBAS 87
5.1 Sprint 1 87
5.1.1 Planificación del Sprint 87
5.1.2 Sprint backlog 89
5.1.3 Duración del Sprint 89
5.1.4 Desarrollo 89
5.2 Sprint 2 120
5.2.1 Planificación del Sprint 120
5.2.2 Sprint backlog 121
5.2.3 Duración del Sprint 121

5.2.4 Desarrollo 122
5.3 Sprint 3 134
5.3.1 Planificación del Sprint 134
5.3.2 Sprint backlog 135
5.3.3 Duración del Sprint 136
5.3.4 Desarrollo 136
5.4 Pruebas 152
5.5 Despliegue 154
6 CAPITULO VI: CONCLUSIONES Y RECOMENDACIONES 155
6.1 CONCLUSIONES 155
6.2 RECOMENDACIONES 156
Bibliografía 158
ANEXOS 163

Índice de figuras

Figura 1 18
Figura 2 19
Figura 3 20
Figura 4 21
Figura 5 22
Figura 6 24
Figura 7 25
Figura 8 26
Figura 9 41
Figura 10 77
Figura 11 78
Figura 12 78
Figura 13 79
Figura 14 79
Figura 15 80
Figura 16 80
Figura 17 81
Figura 18 81
Figura 19 82
Figura 20 82
Figura 21 83
Figura 22 83
Figura 23 84
Figura 24 84
Figura 25 85

Figura 26 86
Figura 27 91
Figura 28 91
Figura 29 92
Figura 30 92
Figura 31 93
Figura 32 93
Figura 33 94
Figura 34 94
Figura 35 95
Figura 36 97
Figura 37 98
Figura 38 98
Figura 39 99
Figura 40 99
Figura 41 100
Figura 42 100
Figura 43 101
Figura 44 102
Figura 45 103
Figura 46 103
Figura 47 104
Figura 48 104
Figura 49 105
Figura 50 106
Figura 51 106
Figura 52 107
Figura 53 108
Figura 54 108
Figura 55 109
Figura 56 110
Figura 57 111
Figura 58 111
Figura 59 112
Figura 60 112
Figura 61 114
Figura 62 114
Figura 63 115
Figura 64 115
Figura 65 116
Figura 66 117
Figura 67 117
Figura 68 118
Figura 69 119
Figura 70 119
Figura 71 120

Figura 72 123
Figura 73 123
Figura 74 124
Figura 75 124
Figura 76 125
Figura 77 125
Figura 78 126
Figura 79 127
Figura 80 127
Figura 81 128
Figura 82 128
Figura 83 130
Figura 84 130
Figura 85 131
Figura 86 131
Figura 87 132
Figura 88 132
Figura 89 133
Figura 90 133
Figura 91 134
Figura 92 134
Figura 93 137
Figura 94 137
Figura 95 138
Figura 96 138
Figura 97 139
Figura 98 139
Figura 99 141
Figura 100 141
Figura 101 142
Figura 102 142
Figura 103 143
Figura 104 144
Figura 105 145
Figura 106 146
Figura 107 147
Figura 108 147
Figura 109 148
Figura 110 149
Figura 111 149
Figura 112 150
Figura 113 150
Figura 114 151
Figura 115 151
Figura 116 152
Figura 117 153

Índice de tablas

Tabla 1 6
Tabla 2 9
Tabla 3 51
Tabla 4 60
Tabla 5 72
Tabla 6 72
Tabla 7 73
Tabla 8 87
Tabla 9 90
Tabla 10 95
Tabla 11 101
Tabla 12 105
Tabla 13 109
Tabla 14 113
Tabla 15 116
Tabla 16 120
Tabla 17 122
Tabla 18 126
Tabla 19 129
Tabla 20 135
Tabla 21 136
Tabla 22 140
Tabla 23 143
Tabla 24 145
Tabla 25 148
Tabla 26 152

1 CAPÍTULO I: INTRODUCCIÓN

El objetivo de esta sección es sentar la justificación para el desarrollo del presente trabajo de disertación, así como sus objetivos generales y específicos.

1.1 Justificación

"La información es uno de los activos potencialmente más valiosos que puede tener una empresa", (Martínez Orencio, 2017) pues de ella dependen muchas de las decisiones que toma la gerencia o colaboradores en general. Si la información no se lleva de una manera adecuada, puede ocasionar problemas, incluso graves pérdidas económicas. Hoy en día ya no es una excusa que el ser una PYME sea impedimento para contar con software que ayude a llevar los procesos de la organización de manera adecuada. Por ende, todas las empresas deben apuntar hacia la automatización de los procesos que permita organizar la información y consultarla en todo momento.

Actualmente existen varios programas informáticos que ayudan a la gestión de procesos a las empresas, pero no todas las herramientas se ajustan al giro del negocio y tienen los estándares de calidad que cada empresa necesita. Por ende, es importante analizar y diseñar el sistema de acuerdo con los estándares de calidad de la empresa y su normativa. Es principal y prioritario para la empresa tener un sistema computarizado para el control del inventario de la empresa y la gestión de las ordenes de trabajo.

La Empresa V-Imagen contará con un Sistema de Inventario y Gestión de Ordenes de Trabajo que cumpla con los requerimientos reales de la organización, lo que conllevará a una mejora en general y crecimiento de ésta. El desarrollo de un sistema web ayudará a tener un mejor control del inventario y órdenes de trabajo, permitiendo mayor control en cuánto a los estados de las tareas ejecutadas.

1.2 Planteamiento del Problema

La gestión de los inventarios es uno de los procesos más importantes dentro de un negocio, de una PYME o Multinacionales, una mala gestión puede ocasionar grandes pérdidas, desembocando en un problema financiero para la empresa.

Dentro de las ventajas de los inventarios Ortega Marqués Ana et al (2017) indica que varios de los beneficios que produce un control efectivo de un inventario son los siguientes:

- Minimizar costos financieros causados por mantener un inventario desmesurado.
- Minimiza el riesgo de pérdida por fraude, robo o daños.
- Evita las pérdidas de ventas por falta de productos.

Actualmente la empresa no cuenta con un sistema para llevar el control de ingresos y egresos de materiales de bodega hasta el departamento de producción, el proceso se lleva a cabo de forma manual.

Esto a su vez acarrea los siguientes problemas secundarios:

- Datos desactualizados de existencia de productos.
- Las existencias físicas no coinciden con los datos del cuaderno.
- No existen reportes de la frecuencia de uso de determinados materiales.
- No se puede identificar datos básicos de proveedores.

Por otra parte, los requerimientos de trabajo son receptados en el local físico y por medios digitales como la página de Facebook o WhatsApp, a partir de ahí se genera manualmente una orden de trabajo en cuadrados pequeños de papel, misma que es entregada al personal operativo para su ejecución, ocasionando problemas de pérdidas de

órdenes de trabajo, confusión por orden de prioridad, lo que desencadena en pérdidas económicas para la empresa.

1.3 Objetivos

1.3.1 Objetivo General

Analizar, Desarrollar e Implementar un Sistema De Inventarios y Gestión de Ordenes de Trabajo, para la Empresa de Soluciones Gráficas V-Imagen De La Ciudad De San Lorenzo.

1.3.2 Objetivos Específicos

- Diseñar y Construir la base de datos que almacenará toda la información generada por el sistema.
- Utilizar la metodología Scrum para el desarrollo del sistema web.
- Desarrollar los módulos correspondientes al control de inventarios.
- Desarrollar los módulos correspondientes a la gestión de ordenes de trabajos.
- Realizar pruebas de funcionamiento y usabilidad al sistema a fin de comprobar que satisface los requerimientos identificados.

1.4 Datos del Negocio: Caso de Estudio “V-Imagen”

1.4.1 Misión

Nuestra misión es ser una empresa integral de artes gráficas, que satisfaga las necesidades de los clientes de forma ágil y eficiente, contribuyendo al bienestar de las áreas donde operamos, con personal comprometido que busca la excelencia.

1.4.2 Visión

Ser la empresa líder en el mercado de industrias gráficas de la provincia de Esmeraldas-Ecuador, reconocida por la calidad de sus trabajos. Expandir nuestros servicios a nivel nacional.

2 CAPÍTULO II: FUNDAMENTOS TEORICOS

El objetivo de esta sección es desarrollar una base de fundamentos teóricos que proporcionará al lector una idea clara acerca del desarrollo del sistema web y las tecnologías aplicadas en su desarrollo.

2.1 Inventarios

2.1.1 Definición

Un inventario se puede definir como: "todos aquellos artículos o stocks usados en la producción (materia prima y productos en proceso), actividades de apoyo (suministro de mantenimiento y reparación) y servicio al cliente (productos terminados y repuestos)." (Durán, 2012).

Existen muchas opiniones alrededor del origen de los inventarios, para Durán (2012) "los inventarios tienen su origen en los egipcios y demás pueblos de la antigüedad" (p.56), las personas de dichos pueblos recurrían a almacenar grandes cantidades de alimentos para consumirlos en los tiempos de escases. Es así como se tienen las primeras nociones de inventario, en sus inicios asociado a una forma de confrontar la sequía, permitiendo así la subsistencia de los negocios y familias. El objetivo principal de un inventario es suplir de insumos necesarios a la empresa para su normal funcionamiento. Juega un papel crucial en el funcionamiento de las operaciones que permite afrontar la demanda del mercado.

2.1.2 Importancia de la organización y administración de un inventario

"La base de toda empresa comercial es la compra y venta de bienes o servicios; de aquí la importancia del manejo de inventario por parte de la misma." (Ortega Marqués Ana et al, 2017, p. 1). La administración del inventario de la empresa es imprescindible para evitar problemas financieros de la empresa. Una de las principales razones por la que se administran los inventarios es porque la empresa necesita generar utilidad, y esto se logra con muchos factores, entre ellos una correcta administración del inventario.

En definitiva, el control de inventarios es necesario y debe ser bien administrado:

El inventario debe ser administrado eficientemente, ya que según Ehrhardt & Brigham (2007) persigue dos objetivos fundamentales: "1) garantizar con el inventario disponible, la operatividad de la empresa y 2) conservar niveles óptimos que permita minimizar los costos totales (de pedido y de mantenimiento)". Contar con una cantidad mínima de materiales o productos para el inventario hace crecer los precios de pedido, mientras que los inventarios grandes incrementan los costos de bodega o mantenimiento. Es el argumento de Ehrhardt & Brigham citado en (Durán, 2012, p. 56)

El nivel de inventarios debe alcanzar un equilibrio, no puede existir activos en exceso puesto que el costo de almacenamiento puede elevarse a fin de convertirse en un problema para la empresa.

Por las razones anteriormente expuestas es necesario llevar un control de inventarios y para ello existen varias técnicas, desde las empresas más pequeñas se ha demostrado que llevar un correcto control de inventarios permite ahorrar grandes cantidades de dinero y ser más productivos.

2.1.3 Tipos de Inventarios

Actualmente existen muchos tipos inventarios, en su mayoría dependen de la naturaleza de la empresa y el criterio que se considere.

La tabla 1 presenta un resumen de los tipos más comunes de inventarios que existen.

Tabla 1

Tipos de inventarios

Tipos de Inventarios	
Materias primas y componentes	Incluye todas las materias primas que forman parte del proceso de fabricación y elaboración del producto terminado, pero que aún no han sido procesados.
Piezas de repuesto de los equipos y de suministros industriales	Este tipo de inventario incluye: **Materias primas secundarias:** Son todos aquellos materiales que son usados durante la elaboración de los productos. **Artículos de consumo:** Son productos que se usan en el entorno de la producción, como combustibles para la maquinaria, artículos de limpieza, etc.
Productos terminados	Aquellos productos que luego de pasar por el proceso de fabricación se envían para su almacenamiento y posterior distribución u comercialización.
Otros: **Inventario de productos en proceso**	Este tipo de inventarios está integrado por aquellos productos que tienen que pasar por algún otro mecanismo para que se conviertan en productos terminados, pero que forman parte de un inventario.

Nota. Fuente: elaboración propia con base en Meana Coalla (2017)

2.1.4 Soluciones Open Source para el Control de Inventarios

En el mercado nos encontramos con varias herramientas Open Source para el control de inventarios de una empresa. Las cuales se detallan a continuación.

1. **ABC Inventory:** Este software tiene una versión gratuita: "Su versión gratuita, para un usuario, permite gestionar múltiples almacenes, números de serie, garantías y muchos otros datos relacionados con el stock." (Taylor & Cabrera, 2019)
2. **Odoo Inventory:** Odoo fue creado con la idea inicial de ser un ERP, tiene incluidos módulos de gestión de almacén. Maneja un inventario de doble entrada: "El inventario de doble entrada de Odoo no computa con entradas, salidas o transformaciones de existencias. En su lugar, todas las operaciones se consideran movimientos de stock entre ubicaciones." (odoo, 2020)
3. **Stockipile de Canvus:** "Canvus es un software de gestión de inventarios dirigido específicamente a pequeñas empresas. Su sistema de gestión de inventarios, llamado Stockpile, ofrece al usuario la posibilidad de aceptar devoluciones, registrar ventas y artículos dañados y supervisar la rotación de productos." (Taylor & Cabrera, 2019)

Las tres soluciones presentadas previamente resultan un gran aliado para algunas empresas, ya que les permite gestionar sus inventarios, sin embargo, la mayoría de las empresas tiene que ajustar la forma de llevar su inventario a lo que la herramienta les permita, sin poder extender funcionalidades propias del giro del negocio. Por otra parte, es posible editar el código fuente debido a la naturaleza de la herramienta y parametrizar ciertos aspectos de acuerdo con las necesidades del negocio, pero para ello se requiere que las empresas contraten expertos en el área de programación, o a su vez que tengan un

departamento de desarrollo. Estas soluciones Open Source no siempre son la mejor opción para una empresa en crecimiento.

2.1.5 Ordenes de Trabajo

Una orden de trabajo se define como "un documento escrito que la empresa le entrega a la persona que corresponda y que contiene una descripción pormenorizada del trabajo que debe llevar a cabo." (Ucha, 2013)

Las ordenes de trabajo permiten llevar un control organizado de las tareas que se le encargan al departamento de producción. Sin embargo, en muchas ocasiones esto se puede convertir en un problema, porque pueden extraviarse, deteriorarse o no definen claramente lo que se requiere. Por esta razón surgen los módulos de control de ordenes de trabajo en los sistemas de información que permiten automatizar este proceso, permitiendo que sea ágil, y tenga una alta disponibilidad, consultando eficientemente el orden de prioridad de las ordenes, los recursos a usarse y las disposiciones generales de la orden.

2.2 Metodología de Desarrollo

2.2.1 Metodologías de Desarrollo Ágil

Al referirnos a las metodologías de desarrollo agiles implica también mencionar a las metodologías de desarrollo tradicionales, puesto que las agiles nacen como alternativa a las tradicionales.

Las metodologías de desarrollo tradicionales son en su mayoría orientadas a la planificación como eje fundamental, fueron concebidas para trabajar sobre proyectos grandes. Sus fases iniciales comúnmente son Análisis y Diseño para posteriormente realizar la validación de requerimientos.

Las metodologías agiles resultan más flexibles que las tradicionales, analizan un proyecto y lo dividen en partes pequeñas, está orientado hacia las personas y no hacia los procesos.

La tabla 2 muestra una comparación entre las metodologías agiles y tradicionales, en donde se aprecian varias de sus características propias de cada una.

Tabla 2

Metodologías tradicionales vs agiles

Metodologías tradicionales	Metodologías agiles
Predictivos	Adaptativos
Orientados a procesos	Orientados a personas
Proceso rígido	Proceso flexible
Se concibe como un proyecto	Un proyecto es subdividido en varios proyectos más pequeños.
Poca comunicación con el cliente	Comunicación constante con el cliente
Entrega de software al finalizar el desarrollo	Entregas constantes de software
Documentación extensa	Poca documentación

Nota. Fuente: Cadavid et al.(2013)

Existen diversas opiniones sobre el surgimiento de las metodologías agiles y los problemas que llegan a resolver: “Las metodologías ágiles resuelven los problemas surgidos, posteriormente, a la masificación del uso del computador personal, dado que las expectativas y necesidades por parte de los usuarios se hicieron más urgentes y frecuentes.” (Herrera & Valencia, 2007, p.382).

Según Herrera & Valencia (2007) el año 2001 fue histórico para el fortalecimiento de las metodologías agiles, se reunieron en Estados Unidos, 17 empresarios de software, debatiendo temas como valores y principios que deben gobernar en el desarrollo de software de calidad, en este encuentro se acepta el termino ágil, para hacer alusión a aquellos nuevos enfoques metodológicos para el desarrollo de software. Como resultado de esta junta se creó lo que se denomina el "Manifiesto Ágil", aquellos plantearon una serie de valores y principios que rigen las metodologías de desarrollo ágil. (p. 382-383)

Entre las metodologías de desarrollo ágil más comunes y usadas actualmente se encuentran: SCRUM, Extreme Programming, Crystal Methodologies, Adaptive Software Development (ASD).

Para el presente proyecto se hará uso de la metodología Scrum, puesto que es una metodología que permite manejar de manera organizada y controlada las tareas que están involucradas en el proyecto, esto sin duda trae consigo ciertas ventajas como ahorro del tiempo. Sin duda Scrum permite que las partes interesadas participen activamente y se logre una integración deseada.

2.2.2 Scrum

Scrum es una metodología ágil para la gestión de proyectos relacionados con el desarrollo de Software, se ha popularizado por su versatilidad, lo que la ha convertido en un conjunto de buenas prácticas para el trabajo colaborativo.

"Scrum es una metodología para el desarrollo de software iterativa e incremental, debe su nombre a la jugada de rugby llamada de la misma manera, se dice que es iterativa ya que se ejecuta en bloques temporales cortos y fijos" (Cohn, 2010 como se citó en Salazar et al., 2018, p. 32). Estos bloques temporales reciben el nombre de Sprint, y son incrementales obteniendo parte del producto final en cada iteración.

2.2.2.1 Roles de Scrum

Los equipos de Scrum están conformados por 3 roles.

1. ***El Scrum Máster:*** responsable de asegurar los procesos.
2. ***El Dueño del Producto (Product Owner):*** responsable de maximizar el valor del producto.
3. ***El Equipo (Development Team):*** responsable de realizar el trabajo.

2.2.2.2 Artefactos y Herramientas

"Scrum, propone tres herramientas o "artefactos" para mantener organizados los proyectos: Estos artefactos, ayudan a planificar y revisar cada uno de los Sprint, aportando medios ineludibles para efectuar cada una de las ceremonias". (Bahit, 2012, p. 39)

1. **Backlog (Lista) de Producto**: Es una lista dinámica y visible para todos los involucrados en el proyecto. El responsable de este artefacto es el dueño del producto. Dentro del backlog de producto nos encontramos las famosas historias de usuario, la cual la podemos definir como un requisito del sistema expresado en el lenguaje común del usuario.
2. **Backlog de Sprint**: "El Backlog de Sprint es una lista reducida de ítems del Backlog de Producto, que han sido negociados entre el Dueño de Producto y el Scrum Team durante la planificación del Sprint." (Bahit, 2012, p. 45). Esta lista se crea al inicio de cada sprint y en ella se plasman las características que el equipo se compromete a desarrollar. Esta lista es actualizada diariamente por el equipo, en ella se muestra las tareas pendientes, en curso y finalizadas, así como una estimación del esfuerzo pendiente de cada tarea sin finalizar y el nombre de la persona del equipo dueño de dicha tarea.

3. **Scrum Taskboard:** Es un tablero, al que comúnmente le llamamos pizarra, el cual permite llevar un control de los trabajos que se encuentran pendientes, aquellos que están en procesos y los que están finalizados.
4. **Diagrama de Burndow:** Es un gráfico que permite medir visualmente, el progreso de los ítems del Backlog de Producto.

2.2.2.3 Sprint

El sprint es considerado el centro de todo Scrum, ya que está relacionado a todos los demás eventos existentes dentro de la metodología. Se lo puede asociar a la idea de mini proyectos con una duración corta variando entre una y cuatro semanas como máximo. Otra de sus características es que es secuencial, no hay tiempo intermedio entre cada sprint: "El objetivo principal es transformar un conjunto de ítems requeridos por el cliente en un incremento de funcionalidad 100% operativa para el software" (Bahit, 2012, p. 30)

2.2.2.4 Ceremonias en Scrum

En Scrum una ceremonia se refiere a una reunión particularmente cuando se refiere a las cuatro reuniones que se realizan de forma iterativa en cada Sprint:

1. *Planificación del Sprint*: Participan el Dueño de Producto, el Scrum Máster y el Scrum Team. El objetivo de esta ceremonia es que "el Dueño de Producto pueda presentar al equipo, las historias de usuario prioritarias, comprendidas en el Backlog de producto; que el equipo comprenda el alcance de estas mediante preguntas; y que ambos negocien cuáles pueden ser desarrolladas en el Sprint que se está planificando." (Bahit, 2012, p. 50)

2. *Reunión diaria:* Las reuniones diarias para Scrum son un tipo de conversación que no superan los 10 a15 minutos, las mismas son realizadas por el Scrum Máster con cada participante del equipo.
3. *Revisión:* Durante la ceremonia de revisión en Scrum, el equipo presentará al dueño de producto las funcionalidades que han sido desarrolladas. “El Dueño de Producto podrá sugerir mejoras a las funcionalidades desarrolladas, aprobarlas por completo o eventualmente, rechazarlas si considera que no se ha cumplido el objetivo.” (Bahit, 2012, p. 54)
4. *Retrospectiva:* El objetivo de esta ceremonia no es más que hacer una revisión, un análisis, de lo que se ha hecho, los resultados obtenidos a fin de identificar lo que se ha hecho bien y lo que se tiene que mejorar. “La retrospectiva en Scrum suele ser vista como una terapia de aprendizaje, donde la finalidad es aprender de los aciertos, de los errores y mejorar todo aquello que sea factible.” (Bahit, 2012, p. 55)

Scrum es actualmente de las metodologías más usadas en la industria, no solo de desarrollo, sino que, en proyectos de todo tipo, gracias a su fácil adaptación a las necesidades del proyecto o la empresa, eso nos invita a reflexionar sobre si es posible aplicar Scrum a proyectos de tipo personal, y la respuesta es sí.

En proyectos personales, una persona puede cumplir distintos roles. Por esta razón para el presente proyecto se tendrá los siguientes roles, que están especificados en la siguiente sección.

- *Producto Owner:* es o representa al cliente para el cuál se está desarrollando el proyecto. No siempre es el dueño del proyecto, pero es la persona capacitada para transferir las necesidades del cliente hacia el equipo de desarrollo conjunto con el

Scrum Máster. Para el presente proyecto el Product Owner es el gerente general de la empresa V-Imagen. En términos generales este rol es quién sabe qué quiere y cómo lo quiere.

- *Scrum Máster:* Angelo Benavidez
- *Equipo de Desarrollo:* Angelo Benavidez

2.3 Aplicaciones Web

2.3.1 Definición

Una aplicación o un sistema web "es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (HTTP) están estandarizados y no han de ser creados" (Luján Mora, 2002, p. 48), siendo una característica de un sistema web la posibilidad de usarlo desde cualquier navegador.

Otra de las características de los aplicativos webs son los protocolos que maneja para su funcionamiento, Lerma Blasco et al., (2013) afirmó lo siguiente:

> El principal protocolo utilizado en la Web es el HTTP, que es el Protocolo de Transferencia de Hipertexto (HyperText Transfer Protocol). Fue creado en 1989 en el CERN (Laboratorio Europeo de Física de las Partículas) como un medio para compartir los datos científicos a nivel internacional, rápidamente y a bajo coste. (p.10)

2.3.2 Lenguajes de programación orientados a la web

Gracias a la interoperabilidad, que ofrece la posibilidad de conectar servicios programados en distintos lenguajes a un servicio web, sin que el usuario final perciba esto, el lenguaje en que están construidos los sitios web ha pasado de ser un tema tedioso, a ser algo más sencillo para los desarrolladores, ya que pueden mezclar distintas

tecnologías, esto a su vez es cuestión del arquitecto de software quién toma esa decisión en base al proyecto. Sin embargo, existen lenguajes que actualmente son los más usados a la hora de programar un sitio web.

- **JavaScript:** JavaScript es un lenguaje de programación que se usa principalmente para crear páginas web dinámicas. “Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.” (Pérez, 2009, p. 5). A pesar de la similitud en su nombre con Java, no guardan una relación directa. Cabe destacar que su principal uso se da para el desarrollo Frontend con librerías como Vue, React, o en su defecto puro. Gracias a su popularidad, el uso para desarrollo Backend se ha incrementado con la llegada del framework Node.js.
- **Python:** Es un lenguaje de programación popular que tiene origen en 1991, su creador es Guido van Rossum, en su mayoría es usado para desarrollo web, para desarrollo de software (GUI), Matemáticas, Sistemas de encriptación, Redes.
- **PHP:** Es de los lenguajes de programación más populares para el desarrollo de lado del servidor, con su framework Laravel se han convertido es una las opciones más usadas por programadores.
- **Java:** Es uno de los lenguajes de programación más antiguos en el área de desarrollo web, a pesar de sus años muchos programadores lo siguen usando debido a su versatilidad, su estructura multiplataforma y su categoría de Open Source.

- **Go:** Uno de los lenguajes de programación más modernos es GO, fue creado en el año 2007 y lanzado en 2009 por Google, inspirado en C, su característica principal es el tipado estático. A pesar de ser de los más jóvenes, es bastante robusto, a tal punto que empresas gigantes como Netflix lo usa, el mismo Uber ha puesto en marcha el manejo de datos en tiempo real (georreferencia) con Go.
- **C#:** "Es un lenguaje de programación moderno, basado en objetos y con seguridad de tipos. C# permite a los desarrolladores crear muchos tipos de aplicaciones seguras y sólidas que se ejecutan en .NET." (Microsoft, 2016). Actualmente es uno de los lenguajes que más fuerza ha tomado entrando al campo del desarrollo web, gracias a su versatilidad le ha permitido incursionar en terrenos como los videojuegos, aplicaciones móviles y por supuesto aplicaciones web, una de sus tecnologías más usadas es ASP.NET

2.4 Arquitectura de Software

Cuando hablamos de arquitectura de software nos encontramos con que existan varias definiciones sin llegar a un consenso, pero sin importar los detalles todas las definiciones apuntan hacia que la arquitectura de software es la forma en la que está estructurado el sistema, sus componentes y las relaciones entre los mismos. "La arquitectura de software es el diseño de más alto nivel de la estructura de un sistema, el cual consiste en un conjunto de patrones y abstracciones que proporcionan un marco claro para la implementación del sistema" (Blancarte Iturralde, 2020, p. 30).

2.4.1 Estilos arquitectónicos

Cuando nos referimos a un estilo, podemos destacar como en cualquier ámbito profesional de aquello que hace la diferencia respecto a otro artefacto de su estilo, es decir es una tendencia que sigue un conjunto de estándares. "Un estilo arquitectónico establece

un marco de referencia a partir del cual es posible construir aplicaciones que comparten un conjunto de atributos y características mediante el cual es posible identificarlos y clasificarlos.” (Blancarte Iturralde, 2020, p.46). Es importante recordar que un estilo arquitectónico nos brinda los lineamientos a seguir en función de las funcionalidades o características del software, pero no nos dice que herramientas debemos usar. Existen varios estilos arquitectónicos detallados a continuación.

2.4.1.1 Monolítico

Tipo de arquitectura diseñada para ser autosuficiente, sin dependencias externas. El código en este tipo de arquitecturas está formado en una sola unidad al momento de compilar la aplicación, esto quiere decir que se puede construir código de forma separada por módulos o librerías, que al compilarse formarán una sola unidad. Se considera al estilo monolítico de los primeros en usarse no como fruto de un invento sino más bien por el hecho de que de esta manera se construían los sistemas al inicio de la computación. “Lo que define un estilo monolítico no es el número de clases, archivos o líneas de código, lo que define es que es autosuficiente, es decir, que tiene todo la funcionalidad para operar por sí mismo y sin depender de nadie más” (Blancarte Iturralde, 2020, p.135).

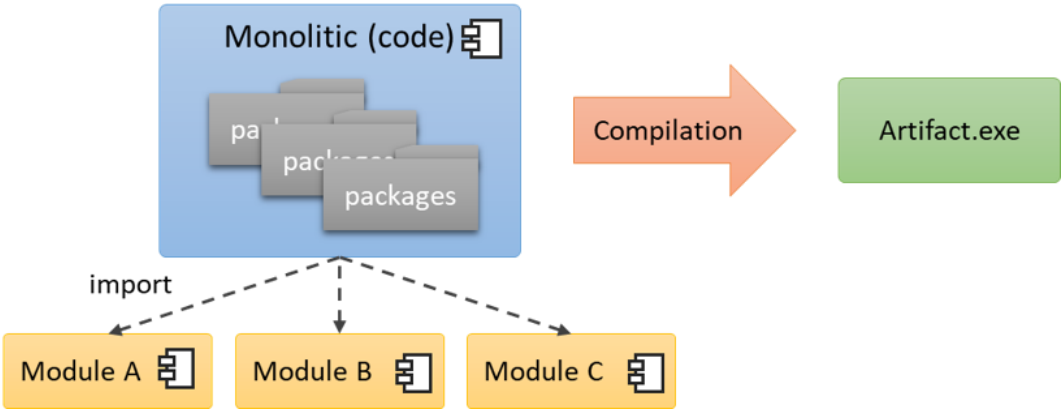
Una de las ventajas de este estilo, suele ser que los focos de errores no son muchos, puesto que es fácil de identificarlos, por otra parte, su desarrollo no suele ser complejo gracias a ello su proceso de pruebas al tratarse de una unidad de código disminuye en complejidad.

Del lado de sus desventajas debemos pensar que al tratarse de una unidad, si algo llega a corromperse puede ocasionar una falla general del sistema, además este tipo de arquitecturas no es recomendable en la actualidad para proyectos grandes que conlleven sistemas robustos. En la Figura 1 podemos observar cómo es el proceso de compilación

dentro de una arquitectura monolítica, todos los módulos son empaquetadas en un solo para producir el producto final.

Figura 1

Arquitectura monolítica estructura general.



Nota. Tomada de Blancarte Iturralde, 2020 (p.137)

2.4.1.2 Cliente – Servidor

Con el inicio de automatización de procesos los sistemas necesitaron de un modelo de arquitectura más eficiente que se adapte a los requerimientos de los usuarios, por ello los programadores pensaron como una forma de solución el modelo cliente-servidor, llamado de esta manera porque un usuario que hacía uso de los recursos del sistema hacía peticiones hacia un servidor y este último respondía con la solución al requerimiento, la particularidad es que no debía ser un único cliente, sino múltiples clientes pueden realizar solicitudes simultáneas, este modelo se convirtió en el más conocido y usado durante mucho tiempo. El cliente es básicamente la forma de darle al cliente una interfaz de interacción que solo muestra los datos, mientras que el servidor hace la ejecución de las tareas, una de las particularidades más interesantes de este modelo es que el cliente y el servidor no necesariamente deben estar en un mismo sitio por ello este estilo arquitectónico es considerado distribuido, y las tecnologías para construir cada una no necesariamente deben ser construidos bajo una misma tecnología. En cuanto a los

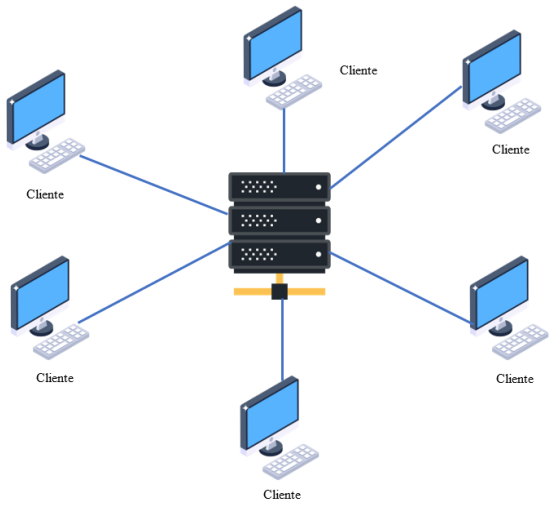
mecanismos de comunicación del cliente con el servidor se realiza a través de la red. “La forma más pura de realizar una comunicación entre un cliente y un servidor es mediante un canal de comunicación directo, por medio de un protocolo de comunicación en común, como podría ser el TCP/IP” (Blancarte Iturralde, 2020, p.148).

Una de los de beneficios de usar este tipo de estilo arquitectónico es sin duda la separación de responsabilidades, en tanto una de sus limitaciones es en el caso de una falla del servidor todo el sistema se verá afectado.

En la Figura 2 se puede apreciar cómo es la estructura de una arquitectura cliente servidor. Un cliente puede hacer una petición al servidor desde distintos sitios, y el servidor se encarga de procesar la petición y enviar una respuesta

Figura 2

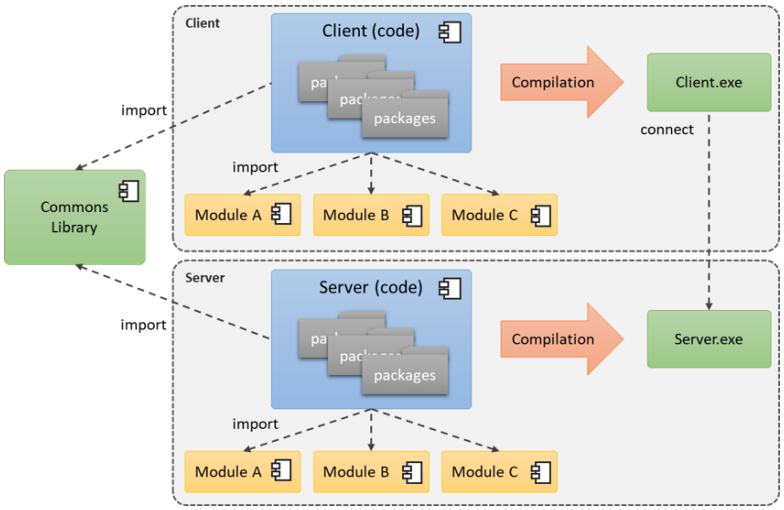
Arquitectura cliente-servidor



Nota. Adaptada de Blancarte Iturralde, 2020, (p. 143)

Figura 3

Estructura de una aplicación cliente-servidor



Nota. Tomada de (Blancarte Iturralde, 2020 (p. 147)

En tanto, en la Figura 3 observamos la estructura de una aplicación construida bajo de arquitectura cliente servidor y como se comunican los distintos objetos que intervienen.

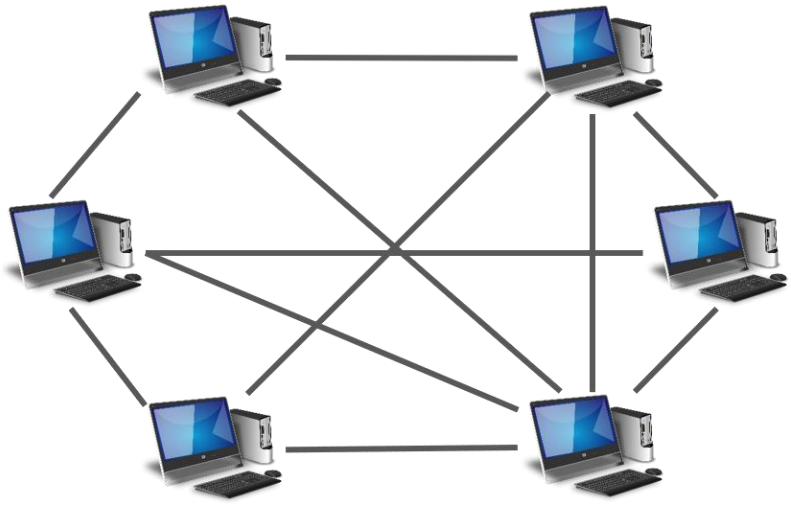
2.4.1.3 Peer-to-peer

El estilo arquitectónico Peer-to-peer (P2P) por sus siglas en inglés denominado de igual a igual hace referencia a una red entre iguales, “es una red de computadoras donde todos los dispositivos conectados a la red actúan como cliente y servidor al mismo tiempo” (Blancarte Iturralde, 2020, p.159), muchos la asocian como una variante del estilo arquitectónico cliente servidor, pero la realidad es que mientras P2P busca descentralizar el funcionamiento de las aplicaciones el primero las centraliza, además en el modelo P2P no se necesita de un servidor específico para que la aplicación pueda funcionar.

Dentro de las ventajas que tenemos en este estilo arquitectónico la tolerancia a un fallo es de las más destacables puesto que si un nodo de la red falla los demás pueden cubrir y permitir que la aplicación continue en funcionamiento, por otra parte, la seguridad es un tema del cual se debe tomar excesivas medidas en este tipo de arquitectura, el hecho de que puedan afectar un nodo perjudica a todo el sistema. En la Figura 4 se puede apreciar la estructura de una arquitectura P2P, teniendo a todos los nodos conectados.

Figura 4

Arquitectura peeer to peer



Nota. Tomada de (Blancarte Iturralde, 2020(p. 159)

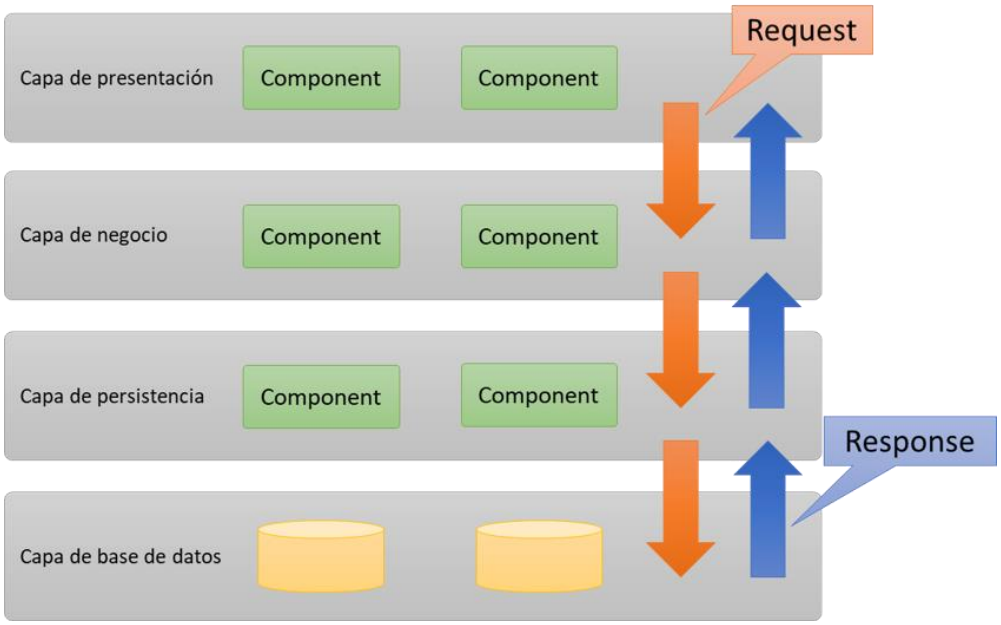
2.4.1.4 Arquitectura en Capas

Este tipo de estilo arquitectónico es de los más utilizados, de hecho, cuando se no tiene definido un tipo de arquitectura se elige esta por defecto. “La arquitectura en capas consta en dividir la aplicación en capas, con la intención de que cada tenga un rol muy definido” (Blancarte Iturralde, 2020, p.82). Las capas que se manejan en este tipo de estilos son la capa de presentación, capa de negocio y la capa de datos, sin embargo, se

puede añadir una capa denominada persistencia, que se acostumbre a unificar con la capa de datos. La Figura 5 muestra la arquitectura y las capas existentes, así como la comunicación entre las capas.

Figura 5

Arquitectura en capas y comunicación entre las mismas



Nota. Tomado de Blancarte Iturralde, 2020 (p. 183)

- **Capa de presentación**: es la capa que permite al usuario interactuar con el sistema, comúnmente conocida como la capa de usuario, a su vez esta capa lleva todo el flujo hacia la capa de negocio.
- **Capa de negocio:** contempla todas las reglas del negocio, es decir la lógica con la que se van a ejecutar las distintas funcionalidades de la aplicación
- **Capa de datos:** es la capa encargada del almacenamiento de los datos, está formada por uno o más motores de bases de datos como MYSQL, SQL Server, Oracle, MariaDB, MongoDB, etc.

Una de las ventajas de este estilo arquitectónico es la separación de responsabilidades a un nivel más detallado donde cada capa tiene asignada ciertas responsabilidades, a su vez realizar pruebas a los sistemas desarrollados bajo este estilo tiene menor complejidad, mientras que dentro de sus desventajas se tiene que si una capa falla el sistema falla en cascada.

2.4.1.5 Arquitectura Dirigida por Eventos (EDA)

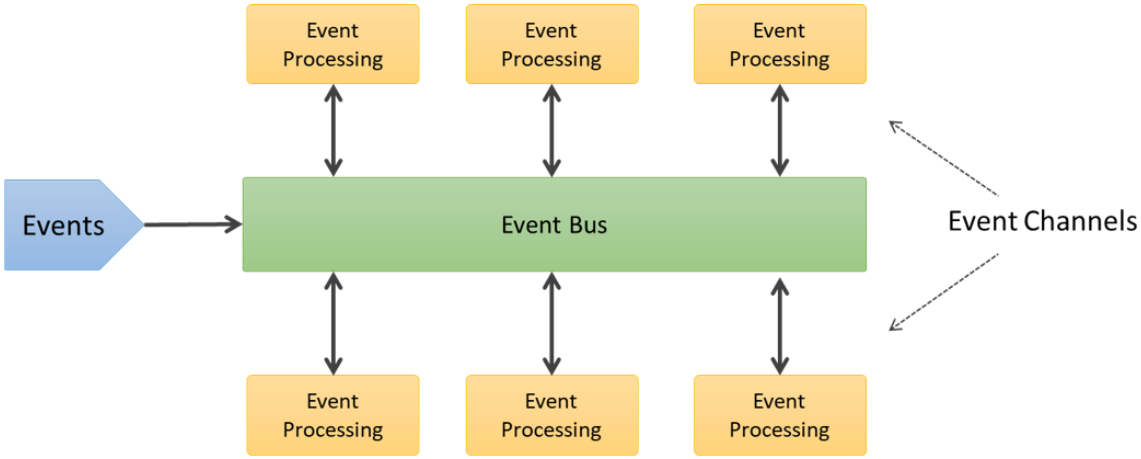
Conocida por sus siglas en inglés Event Driven Architecture (EDA), es un estilo arquitectónico dirigido por eventos. Como es de esperarse este tipo de arquitectura es distinta a las ya revisadas, el hecho de funcionar por eventos hace su esencia radique en esperar que algo pase para que suceda lo siguiente. “En un sistema como este, la captura, la comunicación, el procesamiento y la permanencia de los eventos son la estructura central de la solución. Esto difiere del modelo tradicional basado en solicitudes.” (Red Hat, 2019)

Dentro de las ventajas de este estilo arquitectónico es su fácil despliegue gracias al bajo acoplamiento de sus componentes, además su flexibilidad para responder a cambios. Por otra parte, una gran desventaja es hacer pruebas dentro de este tipo de arquitecturas, los casos de prueban deben ser más extensos para validar cada uno de los eventos.

En la Figura 6 se puede observar la arquitectura dirigida por eventos, donde se tiene el bus de eventos como base de la arquitectura.

Figura 6

Arquitectura dirigida por eventos



Nota. Tomada de Blancarte Iturralde, 2020 (p.245)

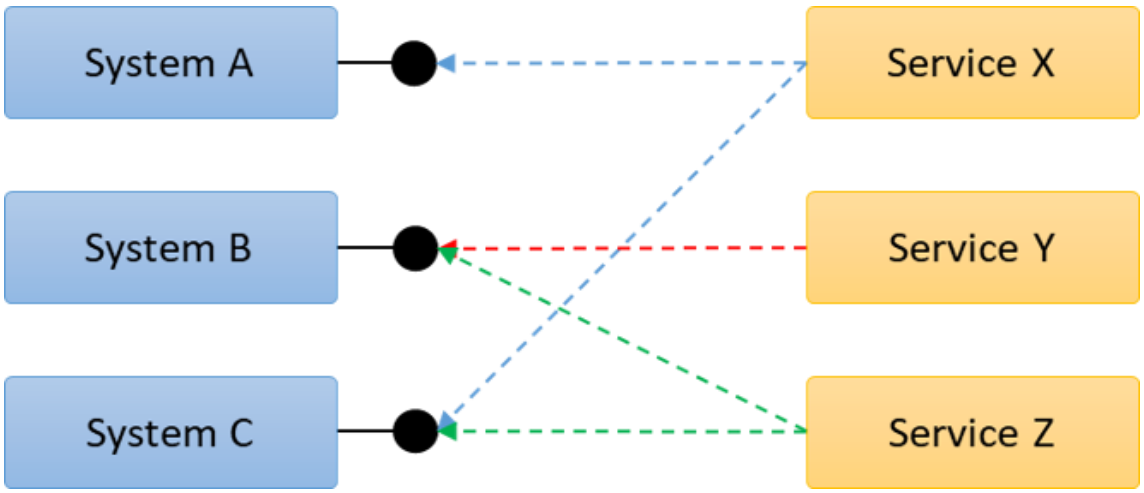
2.4.1.6 Arquitectura Orientada al Servicio

Conocida por sus siglas provenientes del inglés SOA (Service Oriented Architecture), este tipo de estilo resulta interesante conocerlo, puesto que una vez que las aplicaciones empezaron a soportar procesos muy complejos de las organizaciones encontraron un problema grande, muchas de ellas necesitaban módulos que precisamente no eran construidos sobre una misma tecnología, por ejemplo un módulo de un Sistema de Facturación podría ser construido en C y otro modulo en Java, es ahí cuando la arquitectura SOA entra y toma el protagonismo ya que permite que las aplicaciones sean interoperables, es decir, puedan consumir servicios sin importar en que tecnología se haya construido, para formar parte de un todo que sería un sistema o aplicación más compleja, “SOA integra los elementos del software que se implementan y se mantienen por separado, y permite que se comuniquen entre sí y trabajen en conjunto para formar aplicaciones de software en distintos sistemas.” (Red Hat, 2021).

En la Figura 7 se puede apreciar cómo se estructura una arquitectura orientada al servicio, los servicios X, Y, Z son consumidos por las aplicaciones A, B y C, aunque pueden ser consumidas por cualquier otra aplicación.

Figura 7

Arquitectura orientada al servicio



Nota. Tomada de Blancarte Iturralde, 2020 (p.215)

Dentro de las ventajas en este tipo arquitectura tenemos su escalabilidad, puesto que podemos construir nuestros servicios o simplemente tercerizar y consumir un servicio construido por alguien más que cumpla con nuestros requerimientos para el funcionamiento de un proceso, otro de los grandes aportes de usar SOA es la capacidad de reutilización, un mismo servicio puede ser usado para varios fines si el comportamiento entre procesos lo permite. Por el lado de las desventajas tenemos cierta perdida en el rendimiento de la aplicación, el hecho de comunicarse con otros servicios permite que el sistema pueda presentar ciertas fallas en temas de rendimiento.

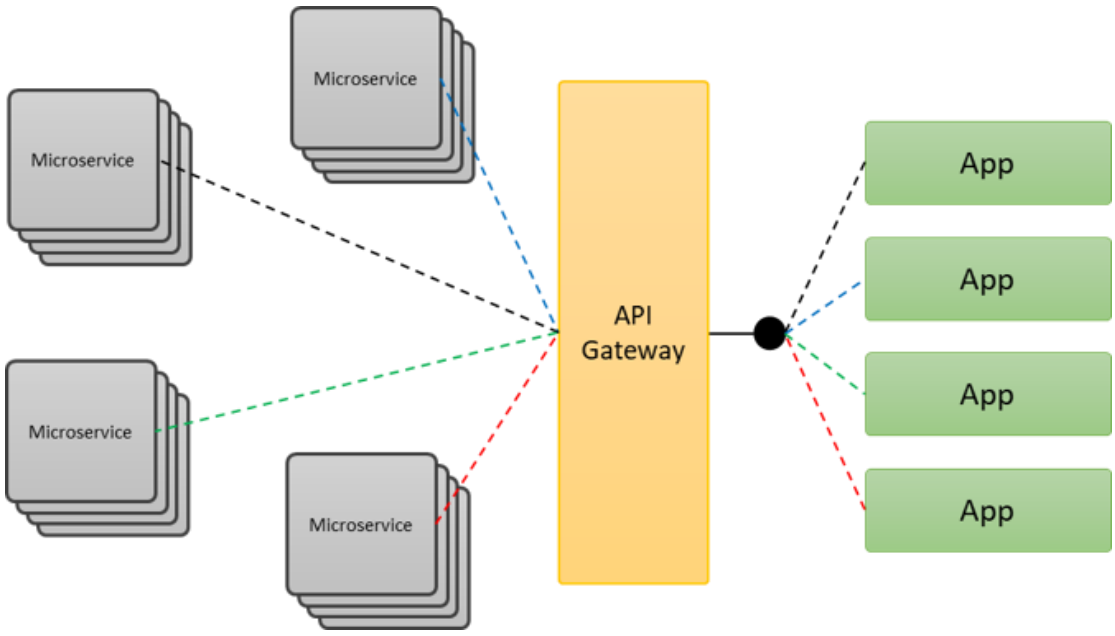
2.4.1.7 Microservicios

Este estilo arquitectónico es moderno y de los más utilizados por las grandes industrias tecnológicas en sus productos que usamos en el día a día, gracias a su forma de

funcionar básicamente consiste en crear componentes de software funcionales dedicados a solventar alguna tarea en específico. “Los microservicios son un enfoque arquitectónico y organizativo para el desarrollo de software donde el software está compuesto por pequeños servicios independientes que se comunican a través de API bien definidas. Los propietarios de estos servicios son equipos pequeños independientes.” (Amazon, 2021). Este estilo impulsa la innovación, permite que las aplicaciones sean escalables, se suele hacer un análisis con el estilo arquitectónico monolítico donde se hace referencia a que la aplicación se construye en un servidor que cubre todas funcionalidades para que un sistema funcione, en tanto, los microservicios funcionan totalmente contrario, haciendo uso del paradigma de divide y vencerás, se producen microservicios destinados a cumplir un proceso y el conjunto de todos los microservicios forman un sistema o aplicación robusta.

Figura 8

Arquitectura de microservicios



Nota. Tomada de Blancarte Iturralde, 2020 (p. 234)

Dentro de sus ventajas tenemos la posibilidad de escalar, es cuestión de añadir microservicios destinados a nuevas funcionalidades, gracias a tener un sistema distribuido hacer pruebas se vuelve manejable pudiendo detectar los microservicios que tienen problemas, además reusar microservicios es común. Por otra parte, al igual que en SOA tendemos a tener problemas de rendimiento por el hecho de consumir varios microservicios, y su trazabilidad es compleja, ya que para completar algún proceso se puede requerir pasar por múltiples microservicios.

2.5 Servicios Web

Con los avances tecnológicos, los sistemas se han visto en la necesidad de escalar, muchos han migrado hacia la web, y estando allí se han encontrado con una serie de diversas posibilidades como la de tener un servidor de base de datos alojado en un sitio distinto incluso lejano geográficamente y otro servidor de Backend igualmente separado y obtener la información necesaria en segundos, esto en gran parte se produjo por los llamados Servicios Web que han permitido que las arquitecturas distribuidas se popularicen. La W3C (2010) define a los servicios web de la siguiente manera: “Se refiere al diseño basado en mensajes que se encuentra con frecuencia en la Web y en el software empresarial. La Web de Servicios se basa en tecnologías como HTTP, XML, SOAP, WSDL, SPARQL y otras.” Existen varios tipos de servicios web que se resumen en las siguientes secciones.

2.5.1 Servicios Web SOAP

Dentro de los tipos de servicios tenemos a SOAP denominado así por sus siglas en inglés (Simple Object Access Protocol) protocolo simple de acceso a objetos, se define como “un formato de mensaje XML utilizado en interacciones de servicios web. Los mensajes SOAP habitualmente se envían sobre HTTP o JMS, pero se pueden utilizar otros

protocolos." (IBM, 2019a). Una de las grandes diferencias que SOAP mantiene con REST, es que el primero es un protocolo, por ello "impone reglas integradas que aumentan la complejidad y la sobrecarga, lo cual puede retrasar el tiempo que tardan las páginas en cargarse lo cual puede retrasar el tiempo que tardan las páginas en cargarse."(Red Hat, 2019)

2.5.2 Servicios Web RESTFUL

A menudo existe una confusión entre la arquitectura REST con el Servicio Web RESTFUL, son 2 cosas distintas. "Una API de REST, o API de RESTful, es una interfaz de programación de aplicaciones (API o API web) que se ajusta a los límites de la arquitectura REST y permite la interacción con los servicios web de RESTful." (Red Hat, 2020). Aquí se introduce un término bastante usado cuando se habla de un sistema o una aplicación Web denominado API, pues una API es una forma de comunicar los distintos nodos de una aplicación en caso de existir, es muy usado en la arquitectura REST bajo el servicio de RESTFULL. "Una API o interfaz de programación de aplicaciones es un conjunto de definiciones y protocolos que se usa para diseñar e integrar el software de las aplicaciones." (Red Hat, 2019). Para tener una idea, la API expone los servicios que han sido construidos del lado del servidor (Backend) para que estos puedan ser consumidos por el lado del cliente (Frontend) de esta manera permite el correcto funcionamiento del sistema en su totalidad.

2.5.3 GRAPHQL

GraphQL fue desarrollo por Facebook en el año 2012, nace como una solución de contrarrestar la sobrecarga de peticiones a sus servidores y respuesta de datos innecesarios para el termino de lo solicitado por el cliente sobre todo en su aplicación móvil, a su vez llega como una alternativa a las RESTAPIS y SOAP. "GraphQL es un lenguaje de

consulta y un tiempo de ejecución del servidor para las interfaces de programación de aplicaciones (API); su función es brindar a los clientes exactamente los datos que solicitan y nada más.” Uno de los mayores beneficios es que permite extraer datos desde varias fuentes haciendo uso de una sola llamada a la API, por otra parte, el almacenar los datos en caché resulta más complejo que por ejemplo si se usará REST.

2.6 Frameworks de Desarrollo

Al hablar de framework se hace referencia a una estructura de software, Gutiérrez (2010) menciona que un framework es una estructura que generalmente está: “compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta” (p. 1),

Los objetivos principales que persigue un framework son: “acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.” (Gutiérrez, 2010, p. 1)

Los frameworks web pueden usar usados tanto de lado del servidor como de lado del cliente. Por ello tenemos:

- **Frameworks del lado del servidor-Backend:** Cuando nos referimos a los frameworks del lado del servidor nos referimos al backend, es decir la lógica de la aplicación. En esta categoría podemos destacar 2 de los más conocidos, uno de ellos es Laravel que usa PHP para construir aplicaciones robustas, su manejador de paquetes Composer le permite agregar una serie de características. Por otro lado, está Node.js es la apuesta para programar del lado del servidor con JavaScript, su manejador de paquetes es NPM, el cual también permite agregar

paquetes o dependencias de forma rápida y segura, actualmente también se ha popularizado el uso de Yarn para el manejo de dependencias.

- **Frameworks del lado del cliente-Frontend:** El Frontend es la parte visual que el usuario percibe e interactúa por ello debe cumplir ciertos estándares de diseño y usabilidad para crear una experiencia agradable al usuario. Para este apartado la variedad de frameworks también es grande, puesto que existen muchas herramientas. Se debe destacar el uso de Next.js el cuál es un framework de React.js, que a su vez es una librería también bastante poderosa creada por Facebook, Next.js nos permite generar interfaces a base de componentes como React.js, los cuáles los podremos reutilizar. Otro de los frameworks populares para el Frontend es Vue.js: "Es un framework progresivo para construir interfaces de usuario. A diferencia de otros frameworks monolíticos, Vue está diseñado desde cero para ser utilizado incrementalmente." (Sitio oficial de Vue.js, 2014)

2.7 Base De Datos

Las bases de datos son la forma en las que las aplicaciones registran la información de manera permanente, además de realizar consultas de manera rápida. La necesidad de respaldar la información viene desde mucho tiempo atrás, empezando por la invención de la escritura, en principio respaldando información en materiales como papiro y posteriormente en papel, pasando por cintas magnéticas hasta llegar a medios digitales como discos duros e incluso a lo que hoy conocemos como la nube.

De esta forma se puede definir a una base de datos como "una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Normalmente, una base de datos está controlada por un sistema de gestión de bases de datos (DBMS)."(Oracle, 2019).

2.7.1 Tipos de Bases de Datos

Las tecnologías que usan las bases de datos han evolucionado y hoy podemos encontrar diversos tipos de base de datos que encajan mucho mejor con ciertas necesidades de la industria o del giro del negocio. A continuación, una clasificación de los tipos de base de datos haciendo énfasis en los más usados modelos relacionales y no relaciones.

- **Base de datos orientada a objetos:** Una base de datos orientada a objetos es el tipo de base que está orientada a trabajar con objetos en lugar de tablas como lo hacen las relacionales, las clases se convierten en las relaciones, las variables en los atributos y los métodos reemplazan a los populares procedimientos almacenados.
- **Base de datos jerárquicas:** Es los primeros tipos de base de datos en donde la información (datos) se establecían en forma de un árbol de forma inversa. Cada nodo puede tener uno o más registros secundarios considerados como hijos.
- **Base de datos relacionales:** Este tipo de base de datos es comúnmente el más conocido y el más usado, por su simplicidad basado en el modelo relacional. “Una base de datos relacional es una recopilación de elementos de datos con relaciones predefinidas entre ellos. Estos elementos se organizan como un conjunto de tablas con columnas y filas.” (Amazon, 2017) . En este estándar de base de datos las tablas se usan para guardan información de los objetos que contiene la base de datos, estos objetos tienen su representación en la vida real, las columnas guardan el tipo de dato, nombre y una columna o registro guarda la información como tal. Es común hablar de llaves primarias y foráneas, las primarias son

identificadores únicos de cada tabla, este valor no se puede repetir, para hablar de llaves foráneas hay que recalcar que están nacen productos de relaciones entre tablas, una llave foránea es el mecanismo que permite incluir una llave primaria de una tabla dentro de otro, permitiendo así relacionarse.

- **Base de datos NoSQL:** El avance agigantado de las aplicaciones ha permitido que se explore otras soluciones diferente a las bases de datos relacionales, y las bases de datos NoSQL han tomado protagonismo. Este tipo de base de datos "están diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas. Las bases de datos NoSQL son ampliamente reconocidas porque son fáciles de desarrollar, por su funcionalidad y el rendimiento a escala."(Amazon, 2020). Dentro de este grupo de base de datos existen clasificaciones de tipos de base de datos NoSQL o comúnmente conocidas como no relacionales. (MongoDB, 2021) destaca lo siguiente.
    - **Base de datos de documentos:** en estas bases de datos se asocia cada clave con una estructura de datos compleja que le designa el nombre de documento.
    - **Almacenes de grafos:** se utilizan para almacenar información sobre redes de datos, como las conexiones sociales. Ejemplos de almacenes de grafos son Neo4J y Giraph.
    - **Almacenes de clave-valor:** son las bases de datos NoSQL más simples. Cada elemento de la base de datos se almacena como un nombre de atributo (o «clave»), junto

con su valor. Ejemplos de almacenes de clave-valor son Riak y Berkeley DB.

- **Base de datos orientadas a columnas:** estas bases de datos, como Cassandra o HBase, permiten realizar consultas en grandes conjuntos de datos y almacenan los datos en columnas, en lugar de filas.

Una de las ventajas de usar Base de Datos NoSQL es que permiten que los equipos modernos de desarrollo realicen un Sprint de manera ágil, fácil estructuración de los esquemas, y su flexibilidad permite generar cambios sin tener que desestructurar el esquema, eso sin perder el orden que requiere un buen esquema. Adicional un dato a tomar en cuenta es que este tipo de base de datos tienen un escalado horizontal a diferencia de las relacionales que tienen un escalado vertical.

2.8 Herramientas

Las herramientas que se han elegido para el presente proyecto son las siguientes.

2.8.1 Backend

Dentro de un sistema web el Backend es el encargado de todos los procesos y servicios necesarios para que un sistema web funcione, esta parte del sistema juega un papel importante ya que contiene toda la lógica del sistema.

2.8.1.1 Node.JS

Node.JS es un ecosistema que trabaja en la capa del servidor fue “ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables.”(OpenJS, 2021). Fue creado en el año 2009 por Ryan Dahl basado en el motor V8 de Google, por tanto, es una tecnología moderna

que sigue evolucionando e imponiéndose en el desarrollo de los sistemas web, una de sus características a resaltar es que es de código abierto y es multiplataforma.

2.8.1.2 Express.Js

Express.JS es uno de los frameworks más populares para construir aplicaciones del lado del Backend con Node.JS. En su sitio web menciona que es "una infraestructura de aplicaciones web Node.js mínima y flexible dándonos así un conjunto de características sólidas para construir aplicaciones web y móviles." (Express.js, 2021). Adicional hay que destacar que Express.Js cuenta con cientos de métodos HTTP y middlewares que permiten que la tarea de construir una API y exponerla desde el Backend sea mucho una tarea más sencilla.

2.8.1.3 Mongoose

Es una biblioteca de JavaScript que permite interactuar con la base de datos de MongoDB, facilitando tareas de inserción, lectura, modificación y eliminación de datos. También se denomina biblioteca de modelado de datos de objetos (ODM) para MongoDB y Node.js. "Administra las relaciones entre los datos, proporciona validación de esquemas y se utiliza para traducir entre objetos en el código y la representación de esos objetos en MongoDB."(Karnik, 2018)

2.8.1.4 Nodemailer

Nodemailer es una biblioteca que nos permite él envió de correos desde nuestra aplicación de servidor de una forma simple, fácil de configurar. "El proyecto comenzó en 2010 cuando no había una opción sensata para enviar mensajes de correo electrónico, hoy en día es la solución a la que recurren la mayoría de los usuarios de Node.js de forma predeterminada." (Nodemailer, 2020). Un aspecto por destacar es su fácil configuración

con los servidores de correo, esto incluye un servidor de prueba que la biblioteca pone por defecto.

2.8.1.5 JSON Web Token

A menudo cuando trabajamos en la construcción de un sistema llegamos a encontrarnos con el tema de la autenticación y la autorización. Con autenticar nos referimos al hecho de dar la posibilidad al usuario de ingresar al sistema con sus credenciales, siempre y cuando sean válidas, precisamente el proceso de confirmar si las credenciales son válidas es el proceso de autenticar. Por otra parte, autorizar es la gestión de permisos que damos al usuario dentro de nuestro sistema, estos permisos pueden darse por ejemplo en función de roles. Precisamente aquí es donde JWT toma relevancia ya que nos permite gestionar el proceso de autenticación de manera segura. “JSON Web Token (JWT) es un estándar abierto (RFC 7519) que define una forma compacta y autónoma de transmitir información de forma segura entre las partes como un objeto JSON.”(JWT, 2017) (JWT, 2017). Para garantizar la seguridad del proceso JWT usa firmas digitales, son una especie de llave de seguridad que solo el origen del JWT conoce y bajo el cual está firmado un token, este normalmente tiene un tiempo de expiración de acuerdo con las necesidades del negocio.

2.8.2 Frontend

El Frontend es la parte gráfica, visual donde el usuario interactúa con el sistema o la aplicación, a diferencia del Backend que funciona del lado del servidor, el Frontend funciona del lado del cliente, al tratarse de sistemas web funcionan normalmente en un navegador web. El desarrollo del Frontend es un tema que se debe considerar al momento de planificar el desarrollo de un sistema, en cuanto a diseño y usabilidad del producto, tratando de construir sistemas fáciles de usar con interfaces amigables.

2.8.2.1 React.JS

React.JS se ha convertido sin duda en unas de las librerías más usadas para construir interfaces del lado del Frontend. Fue creada por Facebook en el año 2013 y es de código abierto. Una de las mayores bondades que ofrece React.JS es trabajar bajo componentes, de esta manera se puede creer un componente de un menú por ejemplo y ser utilizado en distintas vistas, con esto se logra un alto grado de reutilización de componentes.

2.8.2.2 Next.JS

Next.JS es un framework de React.JS creado por Vercel, permite desarrollar aplicaciones JavaScript. Next.JS a diferencia de React.JS usa el Server Side Rendering, que es una técnica de renderización que permite que ciertas páginas se rendericen del lado del servidor quitando carga al cliente, esto nos va a permitir que las páginas carguen más rápido ganando velocidad en nuestro sistema o aplicación.

2.8.3 Base de Datos

2.8.3.1 MongoDB

MongoDB es una base de datos NoSQL más comúnmente denominada no relacional, que ha ganado popularidad siendo una de las alternativas más usadas frente a las bases de datos relacionales. “MongoDB es una base de datos de documentos que ofrece una gran escalabilidad y flexibilidad, y un modelo de consultas e indexación avanzado.” (MongoDB, 2019). Una de sus características propias es que la información en lugar de guardarse en registros como lo hacen las bases de datos relacionales se guarda en documentos en una estructura parecida a JSON (JavaScript Objetc Notation) llamada BSON que es no más que un JSON binario.

2.8.4 Pruebas

Las pruebas dentro del proceso de desarrollo de un sistema juegan un papel importante ya que permite evaluar las distintas funcionalidades del sistema y a su vez detectar errores. Algo a tomar en cuenta es que existen varios tipos de pruebas que se le aplican a un sistema.

- **Pruebas unitarias:** Sirven para corroborar el correcto funcionamiento de cada componente del sistema, como clases, módulos.
- **Pruebas de humo:** Sirven para comprobar el funcionamiento básico del sistema, su objetivo es verificar que las funcionalidades siguen el flujo previste bajo el cual se construyeron.
- **Pruebas de aceptación:** Sirven para comprobar si los requerimientos del negocio han sido satisfechos.

Las pruebas que se aplicarán serán unitarias de forma automatizada, de humo de manera manual. Además de las pruebas de aceptación.

2.8.4.1 JEST

JEST es un framework de JavaScript bastante popular en el ámbito de las pruebas automatizadas. JEST está “diseñado para garantizar la corrección de cualquier código base de JavaScript. Le permite escribir pruebas con una API accesible, familiar y rica en funciones que le brinda resultados rápidamente.”(Facebook Inc., 2021)

2.8.5 Despliegue

El despliegue de una aplicación o sistema web consiste en llevar nuestra aplicación desde el entorno de desarrollo al entorno de producción sea esto local en un servidor dedicado o compartido o de otra forma en la nube, de cualquier manera, se debe garantizar la alta disponibilidad.

2.8.5.1 Heroku

"Heroku es una plataforma de servicios en la nube (conocidos como PaaS o Platform as a Service) que permite manejar los servidores y sus configuraciones, escalamiento y la administración."(Celis, 2018) Al ser un PasS no debemos preocuparnos por nada más que por el desarrollo, a Heroku solo le informamos cuál es el lenguaje en el que vamos a desarrollar (soporta una variedad de lenguajes de programación) y de ser el caso el motor de base de datos que vamos a elegir, y del despliegue se encarga Heroku. Unos los principales usos que se da a Heroku es la de almacenar el Backend de los sistemas compitiendo directamente con gigantes como Amazon Web Services.

2.8.5.2 Vercel

Vercel es una plataforma de código abierto basada en la nube, fundada en Argentina que ha tomado un protagonismo enorme cuando hablamos de despliegue de aplicaciones Frontend, y es que no solo es de las más usadas para desplegar aplicaciones de React, sino que construyó su propia solución de framework para desarrollar aplicaciones Frontend, que se introdujo en la sección anterior, conocido como Next.JS basado en React.JS.

2.8.5.3 MongoDB Atlas

MongoDB Atlas es una base de datos basada en la nube usada en sistemas u aplicaciones modernas. "Implementar una MongoDB completamente administrada garantiza la disponibilidad, la escalabilidad y la conformidad con las normas de seguridad gracias a la automatización inteligente para mantener el rendimiento a escala a medida que las aplicaciones evolucionan."(Amazon, 2019). Atlas está disponible en los servicios de la nube con más demanda como AWS, Azure, Google Cloud Platform.

2.8.6 Control de cambios y manejo de versiones

El control de versiones de software es una práctica muy valorada en el desarrollo de software puesto que permite llevar un orden y registros de los cambios a través del tiempo.

2.8.6.1 Git

Git es un sin duda el software de control de versiones más popular entre la comunidad de desarrolladores del mundo cabe destacar que fue creado por Linus Torvalds y es de código abierto. “Git es un sistema de control de versiones distribuido. Esto significa que un clon local del proyecto es un repositorio de control de versiones completo. Estos repositorios locales plenamente funcionales permiten trabajar sin conexión o de forma remota fácilmente.”(Microsoft, 2021). Git funciona como una base de datos que va guardando los cambios que se realizan al proyecto, se pueden crear ramas con las distintas funcionalidades de un proyecto, para que se puedan trabajar de forma separada y unificarlas a una rama principal solo cuando está ya ha sido probada y verificada, lo más importante es que en caso de que sucedan errores se puede regresar a través del tiempo hasta una versión que se encuentre estable.

2.8.6.2 GitHub

Si Git es la herramienta que permite llevar un registro histórico de los cambios a nivel de un repositorio local, GitHub es la herramienta que permite alojar esos repositorios locales en repositorios remotos donde se pueda colaborar de forma eficiente sobre proyectos. GitHub puede ser vista como la red social de los desarrolladores, ya que permite compartir proyectos, ideas y soluciones usando como base el sistema de Git. Sin embargo, GitHub no es la única forma de alojar código en la nube, existen otras

alternativas como Bitbucket, a pesar de ello la mayor cantidad de desarrolladores usan GitHub como alternativa principal a la hora de respaldar su código.

2.9 Justificación de Herramientas Seleccionadas

Para el presente trabajo de disertación la metodología de desarrollo del sistema web escogida es la de SCRUM, por la manera organizada de llevar a cabo proyectos de cualquier índole y tamaño, ofreciendo flexibilidad a la hora de realizar cambios y ajustes sobre la marcha, recibiendo retroalimentación constante de las partes involucradas.

Por otra parte, las herramientas técnicas seleccionadas para la implementación del sistema web son las siguientes

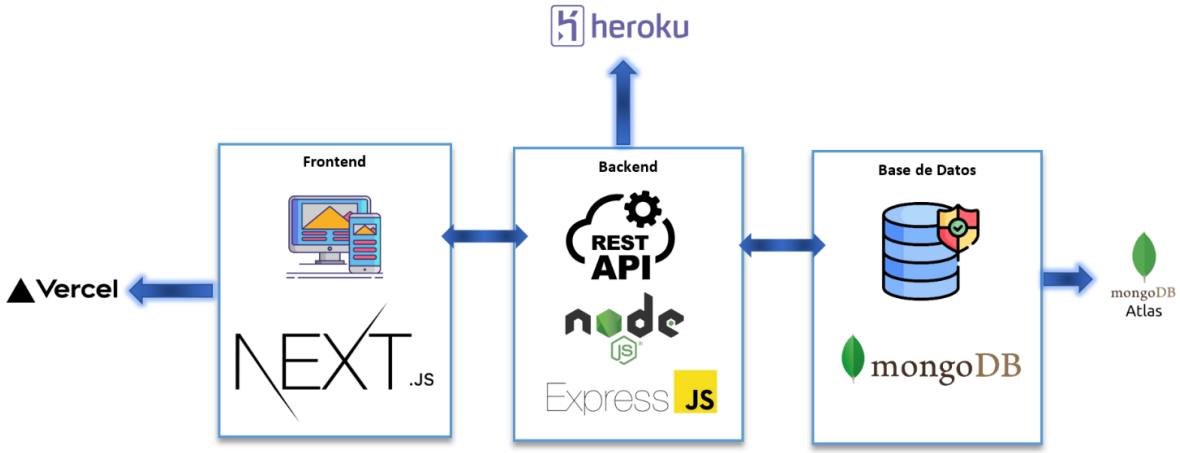
- **Backend:** Para la parte lógica del sistema, se decidió usar el entorno de ejecución Node.JS junto con su framework Express.JS y el ODM de Mongoose para las consultas hacia la base de datos debido a su alto grado de compatibilidad.
- **Frontend:** Se usará el framework de Vercel basado en React.JS, Next.JS que permite que el rendimiento de la aplicación mejore gracias a su renderizada del lado del servido, cargando así las vistas con mayor fluidez.
- **Base de datos:** Se usará base de datos NoSQL MongoDB debido a que ofrece alto rendimiento y escalabilidad permitiendo que el sistema crezca verticalmente sin ver afectado su rendimiento.
- **Pruebas:** Para la ejecución automática de pruebas se usará JEST gracias a su fácil configuración permitirá ejecutar pruebas unitarias y de integración, este framework cuenta con una alta compatibilidad con Next.JS, eso nos brindará retroalimentación inmediata bajo los parámetros establecidos para ejecutar las pruebas.

- **Despliegue:** Para el despliegue del Backend se usará Heroku, esta plataforma como servicio permitirá tener disponible los servicios del backend en todo momento. Por otra parte, para desplegar el Frontend se usará Vercel propietario del framework que se usará para construir la interfaz del sistema, esto permite que exista una alta compatibilidad ayudando al rendimiento del sistema. Para el despliegue de la base de datos se usará MongoDB Atlas a través de MongoDB cloud que ofrece una capa gratuita para usar los servicios en la nube de MongoDB.
- **Control de cambios:** Para llevar un control de cambios del software que se va a desarrollar se usará Git para el manejo de repositorios locales, y GitHub para repositorios remotos debido a que ofrecen gran compatibilidad.

A continuación, en la figura, se presenta la estructura general del sistema web.

Figura 9

Estructura general del sistema



3 CAPÍTULO III: ANÁLISIS Y LEVANTAMIENTO DE REQUERIMIENTOS

El objetivo de esta sección es desarrollar el proceso de levantamiento de requerimientos junto al usuario final, llevando así sus necesidades a requerimientos, todo esto apoyado en el marco de trabajo de Scrum, que nos permite a cada requerimiento asignarle una prioridad y un tiempo estimado para llevar a cabo el desarrollo.

3.1 Historias de Usuario

Las historias de usuario es una herramienta usada para extraer requisitos provenientes del cliente "que agiliza la administración de requisitos, reduciendo la cantidad de documentos formales y tiempo necesarios"(Menzinsky et al., 2020, p.6).

Una vez llevada a cabo una primera junta con los involucrados de los procesos, se logró comprender de manera exitosa el giro del negocio y el contexto sobre el cual nace la necesidad de construir un software que permite administrar el inventario, así como gestionar las ordenes que son emitidas por el administrador y accionados por los distintos colaboradores de la empresa en sus respectivas áreas.

Se estableció un acuerdo con el cliente, que existirán 2 tipos de usuarios denominados: administrador y operador. Además, se identificó 54 historias de usuarios descritas a continuación.

1. "Como administrador del sistema quiero ingresar al sistema con mis credenciales que los operadores pueden ingresar con sus credenciales y sus permisos asociados." (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).

2. “Como administrador del sistema quiero registrar nuevos usuarios en el sistema con un correo, nombres, cedula, el tipo de usuario y su contraseña” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
3. “Como administrador del sistema quiero editar usuarios, cambiando sus datos personales su correo o contraseña en caso de que el usuario la haya olvidado, sin poder observar sus contraseñas” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
4. “Como administrador del sistema quiero buscar usuarios por el nombre o por el correo registrado” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
5. “Como administrador del sistema quiero eliminar un usuario del sistema, cambiando su estado a inactivo y que de esa forma no pueda ingresar al sistema ni ejecutar ninguna operación” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
6. “Como administrador del sistema quiero que todos los usuarios puedan recuperar su contraseña enviando un enlace de recuperación a su correo” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
7. “Como administrador del sistema quiero poder, registrar los datos de los operadores.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
8. “Como administrador del sistema quiero poder actualizar los datos de los operadores” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
9. “Como administrador del sistema quiero ver una lista general de operadores, así como poder buscarlos por cédula, nombres o apellidos.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).

10. “Como administrador del sistema quiero poder eliminar operadores, pero no perder la información para tener un histórico de los operadores.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
11. “Como administrador del sistema quiero registrar los datos de las bodegas que tiene la empresa.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
12. “Como administrador del sistema quiero poder actualizar los datos de las bodegas” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
13. “Como administrador del sistema quiero ver una lista general de bodegas, así como poder buscarlos por su nombre.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
14. “Como administrador del sistema quiero poder eliminar bodegas.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
15. “Como administrador del sistema quiero registrar datos necesarios de los proveedores, así como datos de la persona de contacto frecuente” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
16. “Como administrador del sistema quiero actualizar los datos de los proveedores en caso de que el proveedor cambie la persona de contacto frecuente o se requiera actualizar otro dato en particular.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
17. “Como administrador del sistema quiero poder buscar proveedores por su nombre, razón social, cédula o ruc, así como tener una lista general de todos los proveedores.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).

18. “Como administrador del sistema quiero eliminar proveedores.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
19. “Como administrador del sistema quiero registrar unidades que son asociadas a los productos.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
20. “Como administrador del sistema quiero actualizar las unidades que son asociadas a los productos.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
21. “Como administrador del sistema quiero tener una lista general de unidades o buscar por su nombre.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
22. “Como administrador del sistema quiero eliminar unidades asociadas a productos.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
23. “Como administrador del sistema quiero registrar los datos de las distintas áreas de producción que tiene la empresa.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
24. “Como administrador del sistema quiero actualizar los datos de las distintas áreas de producción que tiene la empresa.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
25. “Como administrador del sistema quiero ver una lista de las distintas áreas de producción que tiene la empresa, además de buscarlas por su nombre.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
26. “Como administrador del sistema quiero eliminar un área de producción.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).

27. “Como administrador del sistema quiero registrar los datos de los clientes, esta acción también la pueden hacer los operadores.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
28. “Como administrador del sistema quiero actualizar la información de los clientes, esta acción también la pueden hacer los operadores.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
29. “Como administrador del sistema quiero buscar clientes por su cédula o documento de identificación o nombres y apellidos, esta acción también la pueden hacer los operadores.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
30. “Como administrador del sistema quiero poder eliminar clientes.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
31. “Como administrador del sistema quiero registrar productos y datos asociados al mismo.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
32. “Como administrador del sistema quiero actualizar datos de los productos” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
33. “Como administrador del sistema quiero ver una lista de los productos, además poder buscar productos por su sku, nombre, proveedor o categoría, esta acción también la pueden hacer los operadores.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
34. “Como administrador del sistema quiero poder eliminar un producto que ya no se requiera.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
35. “Como administrador del sistema quiero poder ingresar las categorías para manejar las ordenes de trabajos, estas categorías están directamente ligadas al giro

del negocio (áreas de producción).” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).

36. “Como administrador del sistema quiero poder actualizar los datos de categorías.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
37. “Como administrador del sistema quiero poder ver una lista general de categorías, así como poder buscarla por su nombre” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
38. “Como administrador del sistema quiero eliminar una categoría.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
39. “Como administrador del sistema quiero registrar una orden de trabajo con los datos necesarios, los más importantes serían la descripción, la fecha de recepción y el cliente asociado” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
40. “Como administrador del sistema quiero actualizar una orden de trabajo, esto será frecuente porque se debe cambiar el estado de, por ejemplo: pendiente-completado, además el operador que tenga asignada esa orden de trabajo podrá también actualizar el estado de esta” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
41. “Como administrador del sistema quiero buscar las ordenes de trabajo por cedula del cliente, apellido del operador o área a la que fue designada.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
42. “Como administrador del sistema quiero eliminar una orden de trabajo.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).

43. “Como administrador del sistema quiero registrar todas las entradas de los productos adquiridos por un proceso de compra.” (J.R. Chuquilla, comunicación personal, 18 de septiembre de 2021).
44. “Como administrador del sistema quiero actualizar las entradas de productos en caso de que se requiera modificar la cantidad en unidades o el proveedor o la fecha de ingreso” (J.R. Chuquilla, comunicación personal, 2 de octubre de 2021).
45. “Como administrador del sistema quiero poder buscar las entradas por concepto, nombre de bodega o productos específicos, además deseo visualizar todas las entradas listadas por fecha en orden descendente.” (J.R. Chuquilla, comunicación personal, 2 de octubre de 2021).
46. “Como administrador del sistema quiero eliminar una entrada de producto.” (J.R. Chuquilla, comunicación personal, 2 de octubre de 2021).
47. “Como administrador del sistema quiero registrar todas salidas de los productos a una determinada área de producción de la empresa.” (J.R. Chuquilla, comunicación personal, 2 de octubre de 2021).
48. “Como administrador del sistema quiero actualizar salidas de productos en caso de que se requiera modificar la cantidad en unidades, el área o la fecha de salida” (J.R. Chuquilla, comunicación personal, 2 de octubre de 2021).
49. “Como administrador del sistema quiero poder buscar salidas por productos específicos, área donde será destinada o nombre de la bodega, además deseo visualizar todas las salidas listadas por fecha en orden descendente.” (J.R. Chuquilla, comunicación personal, 2 de octubre de 2021).
50. “Como administrador del sistema quiero eliminar una salida de producto.” (J.R. Chuquilla, comunicación personal, 2 de octubre de 2021).

51. “Se requiere que el operador pueda actualizar su contraseña, desde la pestaña de cambiar contraseña.” (J.R. Chuquilla, comunicación personal, 14 de octubre de 2021).
52. “Como administrador del sistema quiero que al registrar un producto se cree un registro en el Kardex automáticamente, en el Kardex se requiere llevar un resumen de las entradas y salidas, así como el stock que se dispone en ese momento” (J.R. Chuquilla, comunicación personal, 14 de octubre de 2021).
53. “Como administrador del sistema quiero que cuando se realice una entrada o una salida de un producto el Kardex se actualice y se obtenga el nuevo Stock del producto automáticamente.” (J.R. Chuquilla, comunicación personal, 14 de octubre de 2021).
54. “Como administrador del sistema quiero poder ver un resumen de estadísticas de las principales entidades del sistema, total de órdenes, las de estado pendiente, completado y cancelado, así como el total de productos y la cantidad de productos sin existencias.” (J.R. Chuquilla, comunicación personal, 12 de marzo de 2022).

Una vez recolectadas las historias de usuario, se las analiza y se procede a documentarlas en la siguiente tabla con los campos:

- **Número**: Número que permite identificar a la historia de usuario
- **Nombre de historia**: Titulo pequeño que explica la historia de usuario
- **Descripción**: Detalles específicos de la historia de usuario, donde se cuenta lo que se pretende realizar.
- **Story Point:** Valor que se otorga a cada una de las historias de usuario basada en el esfuerzo y tiempo de desarrollo. Dándole una puntuación de 1 a 5, dónde

1 es bajo esfuerzo y poco tiempo, 5 es máximo esfuerzo con cantidad considerable de tiempo.

- **Rol**: Perfil del usuario autor de la historia de usuario.
- **Prioridad:** Nivel de preferencia que se le otorga a cada historia de usuario categorizada de 1 a 5, siendo 1 baja prioridad y 5 alta prioridad.

Tabla 3

Historias de Usuarios

Número	Nombre de Historia	Descripción	Story Point	Rol	Prioridad
1	Ingreso al Sistema	Como administrador del sistema quiero ingresar al sistema con mis credenciales y que los operadores pueden ingresar con sus credenciales y sus permisos asociados.	4	Administrador/Operador	5
2	Registro de Usuarios	Como administrador del sistema quiero registrar usuarios con sus nombres, correo, cedula, el tipo de usuario y contraseña genérica.	3	Administrador	5
3	Actualizar Usuarios	Como administrador del sistema quiero poder actualizar los datos de los usuarios incluida su contraseña, pero que esta no sea visible a fin de garantizar la seguridad de los usuarios.	3	Administrador	3
4	Consulta por Parámetros	Como administrador quiero poder buscar los usuarios por sus nombres o correos registrados.	3	Administrador	3
5	Eliminar Usuarios	Como administrador del sistema quiero eliminar usuarios de manera que no se pierda la información del usuario pero que no pueda acceder al sistema.	3	Administrador	3
6	Reestablecer contraseña	Como administrador del sistema quiero que tanto administrador como operadores puedan reestablecer su contraseña en caso de olvidarla, mediante enlaces de recuperación a su correo electrónico registrado.	5	Administrador/Operador	5

Número	Nombre de Historia	Descripción	Story Point	Rol	Prioridad
7	Registro de Operadores	Como administrador del sistema quiero poder, registrar los datos de los operadores.	2	Administrador	3
8	Actualizar Operadores	Como administrador del sistema quiero poder actualizar los datos de los operadores.	2	Administrador	3
9	Consulta General y por Parámetros de Operadores	Como administrador del sistema quiero ver una lista general de operadores, así como poder buscarlos por cédula, nombres o apellidos.	3	Administrador	3
10	Eliminar Operadores	Como administrador del sistema quiero poder eliminar operadores, pero no perder la información para tener un histórico de los operadores.	2	Administrador	3
11	Registrar Bodegas	Como administrador del sistema quiero registrar los datos de las bodegas que tiene la empresa.	2	Administrador	2
12	Actualizar Datos de Bodegas	Como administrador del sistema quiero poder actualizar los datos de las bodegas.	2	Administrador	2
13	Consulta General y por Parámetros de Bodegas	Como administrador del sistema quiero ver una lista general de bodegas, así como poder buscarlos por su nombre.	3	Administrador	2
14	Eliminar Bodegas	Como administrador del sistema quiero poder eliminar bodegas.	2	Administrador	2
15	Registrar Proveedores	Como administrador del sistema quiero registrar datos necesarios de los proveedores, así como información de la persona de contacto frecuente.	2	Administrador	3

Número	Nombre de Historia	Descripción	Story Point	Rol	Prioridad
16	Actualizar Proveedores	Como administrador del sistema quiero actualizar los datos de los proveedores en caso de que el proveedor cambie la persona de contacto frecuente o se requiera actualizar otro dato en particular.	2	Administrador	3
17	Consulta General y por Parámetros de Proveedores	Como administrador del sistema quiero poder buscar proveedores por su nombre, razón social, cédula o ruc, así como tener una lista general de todos los proveedores.	3	Administrador	3
18	Eliminar Proveedores	Como administrador del sistema quiero eliminar proveedores.	1	Administrador	3
19	Registrar Unidades	Como administrador del sistema quiero registrar unidades que son asociadas a los productos.	2	Administrador	2
20	Actualizar Unidades	Como administrador del sistema quiero actualizar las unidades que son asociadas a los productos.	2	Administrador	2
21	Consulta General y por Parámetros de Unidades	Como administrador del sistema quiero tener una lista general de unidades o buscar por su nombre.	3	Administrador	2
22	Eliminar Unidades	Como administrador del sistema quiero eliminar unidades asociadas a productos.	1	Administrador	1
23	Registrar Áreas	Como administrador del sistema quiero registrar los datos de las distintas áreas de producción que tiene la empresa.	2	Administrador	3

Número	Nombre de Historia	Descripción	Story Point	Rol	Prioridad
24	Actualizar Áreas	Como administrador del sistema quiero actualizar los datos de las distintas áreas de producción que tiene la empresa.	2	Administrador	3
25	Consulta General y por Parámetros de Áreas	Como administrador del sistema quiero ver una lista de las distintas áreas de producción que tiene la empresa, además de buscarlas por su nombre.	3	Administrador	3
26	Eliminar Área	Como administrador del sistema quiero eliminar un área de producción.	1	Administrador	3
27	Registrar Cliente	Como administrador del sistema quiero registrar los datos de los clientes, esta acción también la pueden hacer los operadores.	3	Administrador/Operador	4
28	Actualizar Cliente	Como administrador del sistema quiero actualizar la información de los clientes, esta acción también la pueden hacer los operadores.	3	Administrador/Operador	4
29	Consulta General y por Parámetros de Clientes	Como administrador del sistema quiero buscar clientes por su cédula o documento de identificación o nombres y apellidos esta acción también la pueden hacer los operadores.	3	Administrador/Operador	4
30	Eliminar Cliente	Como administrador del sistema quiero poder eliminar clientes.	2	Administrador	4
31	Registrar Productos	Como administrador del sistema quiero registrar y datos asociados al mismo.	2	Administrador	4
32	Actualizar Productos	Como administrador del sistema quiero actualizar datos de los productos.	2	Administrador	4

Número	Nombre de Historia	Descripción	Story Point	Rol	Prioridad
33	Consulta General y por Parámetros de Productos	Como administrador del sistema quiero ver una lista de los productos, además poder buscar productos por su nombre, proveedor o categoría, esta acción también la pueden hacer los operadores.	3	Administrador/Operador	5
34	Eliminar Productos	Como administrador del sistema quiero poder eliminar un producto que ya no se requiera.	2	Administrador	4
35	Registrar Categorías	Como administrador del sistema quiero poder ingresar las categorías para manejar las ordenes de trabajos, estas categorías están directamente ligadas al giro del negocio (áreas de producción).	2	Administrador	3
36	Actualizar Categorías	Como administrador del sistema quiero poder actualizar los datos de categorías.	2	Administrador	3
37	Consulta General y por Parámetros de Categorías	Como administrador del sistema quiero poder ver una lista general de categorías, así como poder buscarla por su nombre	3	Administrador	3
38	Eliminar una Categoría	Como administrador del sistema quiero eliminar una categoría.	1	Administrador	3
39	Registrar Orden de Trabajo	Como administrador del sistema quiero registrar una orden de trabajo con los datos necesarios, los más importantes serían la descripción, la fecha de recepción y el cliente asociado.	4	Administrador	5
40	Actualizar Orden de Trabajo	Como administrador del sistema quiero actualizar una orden de trabajo, esto será frecuente porque se debe cambiar el estado de, por ejemplo: pendiente-	4	Administrador/Operador	5

Número	Nombre de Historia	Descripción	Story Point	Rol	Prioridad
		completado, además el operador que tenga asignada esa orden de trabajo podrá también actualizar el estado de esta			
41	Consulta General y por Parámetros de Ordenes de Trabajo	Como administrador del sistema quiero buscar las ordenes de trabajo por cedula del cliente, apellido del operador o área a la que fue designada.	4	Administrador	5
42	Eliminar una orden de trabajo	Como administrador del sistema quiero eliminar una orden de trabajo.	2	Administrador	5
43	Registrar Entradas de Productos	Como administrador del sistema quiero registrar todas las entradas de los productos adquiridos por un proceso de compra.	4	Administrador	5
44	Actualizar Entradas de Productos	Como administrador del sistema quiero actualizar las entradas de productos en caso de que se requiera modificar la cantidad en unidades, el proveedor o la fecha de ingreso.	4	Administrador	4
45	Consulta General y por Parámetros de Entradas	Como administrador del sistema quiero poder buscar las entradas por concepto, nombre de bodega o productos específicos, además deseo visualizar todas las entradas listadas por fecha en orden descendente.	4	Administrador	5
46	Eliminar Entrada de Productos	Como administrador del sistema quiero eliminar una entrada de producto.	2	Administrador	4
47	Registrar Salida de Productos	Como administrador del sistema quiero registrar todas salidas de los productos a una determinada área de producción de la empresa.	4	Administrador	5

Número	Nombre de Historia	Descripción	Story Point	Rol	Prioridad
48	Actualizar Salida de Productos	Como administrador del sistema quiero actualizar salidas de productos en caso de que se requiera modificar la cantidad en unidades, el área o la fecha de salida.	4	Administrador	4
49	Consulta General y por Parámetros de Salidas	Como administrador del sistema quiero poder buscar salidas por productos específicos, área donde será destinada o nombre de la bodega, además deseo visualizar todas las salidas listadas por fecha en orden descendente.	4	Administrador	5
50	Eliminar Salida de Productos	Como administrador del sistema quiero eliminar una salida de producto.	2	Administrador	4
51	Actualizar su contraseña por parte del operador	Se requiere que el operador pueda actualizar su contraseña desde la pestaña de actualizar contraseña.	4	Operador	4
52	Registro de Kardex	Al ingresar un producto, se pueda crear una entrada de ese producto con la cantidad indicada	5	NA	5
53	Actualización de Kardex	Cuando se realice una entrada o una salida de un producto se requiere que el Kardex se actualice y se obtenga el nuevo Stock del producto automáticamente.	5	NA	5
54	Dashboard	Como administrador del sistema quiero poder ver un resumen de estadísticas de las principales entidades del sistema, total de órdenes, las de estado pendiente, completado y cancelado, así como el total de productos y la cantidad de productos sin existencias.	5	Administrador	5

3.2 Product Backlog

El Product Backlog es una lista de las principales funcionalidades del sistema, tienen su origen en las historias de usuarios provenientes de la sección anterior. Estas funcionalidades deben ser comprensibles al usuario final, por lo que será redactado en un lenguaje no técnico, sino más bien cotidiano.

3.2.1 Características del Product Backlog

El Product backlog es un artefacto bastante importante dentro del desarrollo del sistema, por su importancia debe cumplir con las siguientes características.

- El nivel de detalle debe ser alto, de tal manera que se entienda lo que el usuario final necesita y el equipo de desarrollo puede plasmarlo en el producto.
- El nivel de estimación debe ser adecuado, considerando el tiempo y la cantidad de esfuerzo que requiere invertir las personas del equipo de desarrollo para construir dicha funcionalidad
- Se debe establecer prioridad, dependiendo de la importancia para el negocio.

Una vez se ha analizado las historias de usuario, se establece una tabla con los siguientes campos:

- **Número:** Identificación del requerimiento.
- **Descripción:** Descripción del requerimiento.
- **Story Point:** Valor que se otorga a cada una de las historias de usuario basada en el esfuerzo y tiempo de desarrollo. Dándole una puntuación de 1 a 5, dónde 1 es bajo esfuerzo y poco tiempo, 5 es máximo esfuerzo con cantidad considerable de tiempo.
- **Rol:** Perfil del usuario que usará el requerimiento.

- **Prioridad:** Nivel de preferencia que se le otorga a cada historia de usuario categorizada de 1 a 5, siendo 1 baja prioridad y 5 alta prioridad.
- **Sprint:** Número de sprint en el que la funcionalidad será desarrollada.
- **Tiempo:** Estimado en horas para desarrollar la funcionalidad.
- **Criterio de aceptación:** Descripción de los requisitos que la funcionalidad debe cumplir para ser aceptado.

Tabla 4

Product Backlog

Número	Descripción	Story Point	Rol	Prioridad	Sprint	Tiempo	Criterio de aceptación
1	Crear una pantalla de ingreso al sistema donde el usuario pueda ingresar su usuario y contraseña para acceder al sistema.	4	Administrador/Operador	5	1	5 horas	El usuario podrá ingresar al sistema con sus credenciales y saldrá una pantalla con permisos asociados dependiendo si es administrador u operador.
2	Crear una pantalla donde el administrador pueda crear registrar los datos asociados a los usuarios	4	Administrador	5	1	4 horas	Únicamente el administrador podrá ingresar usuarios y sus datos asociados. El sistema guardará los datos para su posterior consulta.
3	Crear una pantalla donde el administrador tenga los datos del usuario seleccionado en la pantalla de consulta general, para posterior modificar sus datos.	4	Administrador	4	1	3 horas	El usuario con privilegios de administrador podrá buscar por nombres y correo de usuario, seleccionar uno y se abrirá una pantalla donde podrá modificar los datos del usuario y guardar los cambios.
4	Crear una pantalla donde el usuario con privilegios de administrador pueda ver una tener un cuadro de búsqueda por los nombres o correo asociado al usuario	3	Administrador	3	1	3 horas	El usuario con privilegios de administrador podrá realizar una búsqueda por parámetros de nombre y correo asociado al usuario.
5	Crear componente eliminar usuario, donde el usuario con privilegios de administrador pueda seleccionar en un cuadro de dialogo si desea eliminar o no el usuario.	2	Administrador	3	1	2 horas	El usuario con privilegios de administrador pulsará en eliminar desde la tabla resultante de la búsqueda por parámetros de usuarios, luego saldrá un cuadro emergente de diálogo donde podrá confirmar si desea eliminar el operador.

Número	Descripción	Story Point	Rol	Prioridad	Sprint	Tiempo	Criterio de aceptación
6	Crear las pantallas necesarias para generar el proceso de restauración de contraseña, el proceso iniciará en la pantalla de login donde el usuario administrador u operador presionen en olvidar contraseña y continuar el proceso. El usuario tendrá 10 minutos a partir del envío del correo de recuperación para introducir su nueva contraseña.	4	Administrador/Operador	5	1	6 horas	El usuario con cualquier perfil podrá restaurar su contraseña desde el login, presionando en el enlace de olvide mi contraseña, ahí deberá introducir el correo asociado a su perfil, posteriormente le llegará un correo donde debe seguir las indicaciones. Finalmente tendrá un cuadro donde debe poner su nueva contraseña, y ya podrá acceder al sistema con sus nuevas credenciales.
7	Crear una pantalla donde el usuario con privilegios de administrador pueda registrar los datos asociados a los operadores.	2	Administrador	3	1	3 horas	Únicamente el administrador podrá ingresar operadores y sus datos asociados. El sistema guardará los datos para su posterior consulta.
8	Crear una pantalla donde el administrador tenga los datos del operador seleccionado en la pantalla de consulta general, para posterior modificar sus datos.	2	Administrador	3	1	3 horas	El usuario con privilegios de administrador podrá ver una lista en una tabla de los operadores y seleccionar uno y se abrirá una pantalla donde podrá modificar los datos del operador y guardar los cambios. La lista de los operadores se encontrará paginada.
9	Crear una pantalla donde el usuario con privilegios de administrador pueda ver una lista general de operadores, y puede filtrar por cédula, nombres o apellidos.	3	Administrador	3	1	3 horas	El usuario con privilegios de administrador podrá ver los datos de los operadores en una lista ubicada en una tabla y tendrá un componente donde pueda buscar por cedula, nombres o apellidos, filtrando los operadores que requiera.
10	Crear componente eliminar operador, donde el usuario con privilegios de administrador pueda	2	Administrador	3	1	2 horas	El usuario con privilegios de administrador pulsará en eliminar desde la tabla donde se

Número	Descripción	Story Point	Rol	Prioridad	Sprint	Tiempo	Criterio de aceptación
	seleccionar en un cuadro de dialogo si desea eliminar o no el operador.						muestran todos los operadores también llamada consulta general, luego saldrá un cuadro emergente de diálogo donde podrá confirmar si desea eliminar el operador.
11	Crear una pantalla donde el administrador podrá registrar los datos asociados a la bodega.	2	Administrador	2	1	3 horas	El usuario con privilegios de administrador podrá registrar nuevas bodegas.
12	Crear una pantalla donde se encuentre los datos de la bodega a actualizar y el usuario con privilegios de administrador pueda modificar uno o varios datos y guardarlo.	2	Administrador	2	1	3 horas	El usuario con privilegios de administrador podrá actualizar los datos de una bodega seleccionada desde la tabla donde se listan todas las bodegas registradas que a su vez estarán paginadas.
13	Crear una pantalla donde el usuario con privilegios de administrador pueda ver una lista general de las bodegas registradas y pueda filtrarlas por nombre	3	Administrador	2	1	3 horas	El usuario con privilegios de administrador podrá consultar los datos de la bodega desde una lista desplegada en una tabla o su vez tendrá una caja en la parte superior de la interfaz donde podrá filtrarla por su nombre.
14	Crear componente eliminar bodega, donde el usuario con privilegios de administrador pueda seleccionar en un cuadro de dialogo si desea eliminar o no la bodega.	2	Administrador	2	1	2 horas	El usuario con privilegios de administrador pulsará en eliminar desde la tabla donde se muestran todas las bodegas también llamada consulta general, luego saldrá un cuadro emergente de diálogo donde podrá confirmar si desea eliminar la bodega.
15	Crear una pantalla donde el administrador podrá registrar los datos asociados al proveedor	2	Administrador	3	1	3 horas	El usuario con privilegios de administrador podrá registrar nuevos proveedores.
16	Crear una pantalla donde se encuentre los datos del proveedor a actualizar y el usuario con	2	Administrador	3	1	3 horas	El usuario con privilegios de administrador podrá actualizar los datos de un proveedor

Número	Descripción	Story Point	Rol	Prioridad	Sprint	Tiempo	Criterio de aceptación
	privilegios de administrador pueda modificar uno o varios datos y guardarlo.						seleccionado desde la tabla donde se listan todos los proveedores registrados que a su vez estarán paginados.
17	Crear una pantalla donde el usuario con privilegios de administrador pueda ver una lista general de los proveedores registrados y pueda filtrarlos por su razón social o nombres.	3	Administrador	3	1	3 horas	El usuario con privilegios de administrador podrá consultar los datos del proveedor desde una lista desplegada en una tabla o su vez tendrá una caja en la parte superior de la interfaz donde podrá filtrarla por su razón social.
18	Crear componente eliminar proveedor, donde el usuario con privilegios de administrador pueda seleccionar en un cuadro de dialogo si desea eliminar o no el proveedor.	1	Administrador	3	1	3 horas	El usuario con privilegios de administrador pulsará en eliminar desde la tabla donde se muestran todas los proveedores también llamada consulta general, luego saldrá un cuadro emergente de diálogo donde podrá confirmar si desea eliminar el proveedor.
19	Crear una pantalla donde el administrador podrá registrar los datos asociados a las unidades de los productos.	2	Administrador	2	1	2 horas	El usuario con privilegios de administrador podrá registrar nuevas unidades necesarias para posterior asociarla a un producto.
20	Crear una pantalla donde se encuentre los datos de la unidad a actualizar y el usuario con privilegios de administrador pueda modificar uno o varios datos y guardarlo.	2	Administrador	2	1	2 horas	El usuario con privilegios de administrador podrá actualizar los datos de una unidad seleccionada desde la tabla donde se listan todas las unidades registradas que a su vez estarán paginadas.
21	Crear una pantalla donde el usuario con privilegios de administrador pueda ver una lista general de las unidades registradas y pueda filtrarlas por nombre	3	Administrador	2	1	2 horas	El usuario con privilegios de administrador podrá consultar los datos de las unidades desde una lista desplegada en una tabla o su

Número	Descripción	Story Point	Rol	Prioridad	Sprint	Tiempo	Criterio de aceptación
							vez tendrá una caja en la parte superior de la interfaz donde podrá filtrarla por su nombre.
22	Crear componente eliminar unidad, donde el usuario con privilegios de administrador pueda seleccionar en un cuadro de dialogo si desea eliminar o no la unidad.	1	Administrador	3	1	1 hora	El usuario con privilegios de administrador pulsará en eliminar desde la tabla donde se muestran todas las unidades también llamada consulta general, luego saldrá un cuadro emergente de diálogo donde podrá confirmar si desea eliminar la unidad.
23	Crear una pantalla donde el administrador podrá registrar los datos asociados a las distintas áreas de producción de la empresa.	2	Administrador	3	1	3 horas	El usuario con privilegios de administrador podrá registrar nuevas áreas.
24	Crear una pantalla donde se encuentre los datos del área a actualizar y el usuario con privilegios de administrador pueda modificar uno o varios datos y guardarlo.	2	Administrador	3	1	3 horas	El usuario con privilegios de administrador podrá actualizar los datos de un área seleccionada desde la tabla donde se listan todas las áreas registradas que a su vez estarán paginadas.
25	Crear una pantalla donde el usuario con privilegios de administrador pueda ver una lista general de las áreas registradas y pueda filtrarlos por nombre.	3	Administrador	3	1	3 horas	El usuario con privilegios de administrador podrá consultar los datos de un área desde una lista desplegada en una tabla o su vez tendrá un recuadro en la parte superior de la pantalla donde podrá filtrarla por su nombre.
26	Crear componente eliminar área, donde el usuario con privilegios de administrador pueda seleccionar en un cuadro de dialogo si desea eliminar o no el área.	1	Administrador	3	1	3 horas	El usuario con privilegios de administrador pulsará en eliminar desde la tabla donde se muestran todas las áreas también llamada consulta general, luego saldrá un cuadro

Número	Descripción	Story Point	Rol	Prioridad	Sprint	Tiempo	Criterio de aceptación
							emergente de diálogo donde podrá confirmar si desea eliminar el área.
27	Crear una pantalla donde el administrador podrá registrar los datos asociados al cliente.	3	Administrador / Operador	4	1	3 horas	El usuario con privilegios de administrador u operador podrá registrar nuevos clientes.
28	Crear una pantalla donde se encuentre los datos del cliente a actualizar y el usuario con privilegios de administrador u operador pueda modificar uno o varios datos y guardarlo.	3	Administrador / Operador	4	1	3 horas	El usuario con privilegios de administrador u operador podrá actualizar los datos de un cliente seleccionado desde la tabla donde se listan todos los clientes registrados que a su vez estarán paginados.
29	Crear una pantalla donde el usuario con privilegios de administrador u operador pueda ver una lista general de los clientes registrados y pueda filtrarlos por su cédula, nombres o apellidos.	3	Administrador / Operador	4	1	3 horas	El usuario con privilegios de administrador u operador podrá consultar los datos del cliente desde una lista desplegada en una tabla o su vez tendrá un recuadro en la parte superior de la pantalla donde podrá filtrarla por su cédula, nombres o apellidos.
30	Crear componente eliminar cliente, donde el usuario con privilegios de administrador pueda seleccionar en un cuadro de dialogo si desea eliminar o no el cliente.	2	Administrador	4	1	3 horas	El usuario con privilegios de administrador pulsará en eliminar desde la tabla donde se muestran todos los clientes también llamada consulta general, luego saldrá un cuadro emergente de diálogo donde podrá confirmar si desea eliminar el cliente.
31	Crear una pantalla donde el administrador podrá registrar los datos asociados a los productos.	2	Administrador	4	2	3 horas	El usuario con privilegios de administrador podrá registrar nuevos productos.
32	Crear una pantalla donde se encuentre los datos del producto a actualizar y el usuario con	2	Administrador	4	2	3 horas	El usuario con privilegios de administrador podrá actualizar los datos de un producto

Número	Descripción	Story Point	Rol	Prioridad	Sprint	Tiempo	Criterio de aceptación
	privilegios de administrador pueda modificar uno o varios datos y guardarlo.						seleccionado desde la tabla donde se listan todos los productos registrados que a su vez estarán paginados.
33	Crear una pantalla donde el usuario con privilegios de administrador pueda ver una lista general de los productos registrados y pueda filtrarlos por sku, nombre, proveedor o categoría.	3	Administrador	5	2	3 horas	El usuario con privilegios de administrador podrá consultar los datos del producto desde una lista desplegada en una tabla o su vez tendrá un recuadro en la parte superior de la pantalla donde podrá filtrarlo por su nombre.
34	Crear componente eliminar producto, donde el usuario con privilegios de administrador pueda seleccionar en un cuadro de dialogo si desea eliminar o no el producto.	2	Administrador	4	2	3 horas	El usuario con privilegios de administrador pulsará en eliminar desde la tabla donde se muestran todos los productos también llamada consulta general, luego saldrá un cuadro emergente de diálogo donde podrá confirmar si desea eliminar el producto.
35	Crear una pantalla donde el administrador podrá registrador los datos asociados a las categorías que maneja las áreas de producción de la empresa	2	Administrador	3	2	3 horas	El usuario con privilegios de administrador podrá registrar nuevas categorías al sistema.
36	Crear una pantalla donde se encuentre los datos de la categoría a actualizar y el usuario con privilegios de administrador pueda modificar uno o varios datos y guardarlo.	2	Administrador	3	2	3 horas	El usuario con privilegios de administrador podrá actualizar los datos de una categoría seleccionando desde la tabla donde se listan todas las categorías registradas que a su vez estarán paginadas.
37	Crear una pantalla donde el usuario con privilegios de administrador pueda ver una lista general de las categorías registradas y pueda filtrarlas por su nombre.	3	Administrador	3	2	3 horas	El usuario con privilegios de administrador podrá consultar los datos de las categorías desde una lista desplegada en una tabla o su vez tendrá un recuadro en la parte superior de

Número	Descripción	Story Point	Rol	Prioridad	Sprint	Tiempo	Criterio de aceptación
							la pantalla donde podrá filtrarla por su nombre.
38	Crear componente eliminar categoría, donde el usuario con privilegios de administrador pueda seleccionar en un cuadro de dialogo si desea eliminar o no la categoría.	1	Administrador	3	2	3 horas	El usuario con privilegios de administrador pulsará en eliminar desde la tabla donde se muestran todas las categorías también llamada consulta general, luego saldrá un cuadro emergente de diálogo donde podrá confirmar si desea eliminar la categoría.
39	Crear una pantalla donde el administrador podrá registrar los datos asociados a las órdenes de trabajo.	4	Administrador	5	2	3 horas	El usuario con privilegios de administrador podrá registrar nuevas órdenes de trabajo al sistema.
40	Crear una pantalla donde se encuentre los datos de la orden de trabajo a actualizar y el usuario con privilegios de administrador u operador pueda modificar uno o varios datos y guardarlo.	4	Administrador / Operador	5	2	3 horas	El usuario con privilegios de administrador u operador podrá actualizar los datos de una orden de trabajo seleccionando desde la tabla donde se listan todas las ordenes de trabajo registradas que a su vez estarán paginadas.
41	Crear una pantalla donde el usuario con privilegios de administrador pueda ver una lista general de las ordenes registradas y pueda filtrarlas por cedula del cliente, apellido del operador o área.	4	Administrador	5	2	3 horas	El usuario con privilegios de administrador podrá consultar los datos de las ordenes desde una lista desplegada en una tabla o su vez tendrá un recuadro en la parte superior de la pantalla donde podrá filtrarla por apellidos del operador, cedula del cliente o nombre del área.
42	Crear componente eliminar orden de trabajo, donde el usuario con privilegios de administrador pueda seleccionar en un cuadro de dialogo si desea eliminar o no la orden de trabajo.	2	Administrador	5	2	3 horas	El usuario con privilegios de administrador pulsará en eliminar desde la tabla donde se muestran todas las ordenes también llamada consulta general, luego saldrá un cuadro

Número	Descripción	Story Point	Rol	Prioridad	Sprint	Tiempo	Criterio de aceptación
							emergente de diálogo donde podrá confirmar si desea eliminar la orden.
43	Crear una pantalla donde el administrador podrá registrar los datos asociados a las entradas de productos por un proceso de compra.	4	Administrador	5	3	4 horas	El usuario con privilegios de administrador podrá registrar nuevas entradas de productos al sistema.
44	Crear una pantalla donde se encuentre los datos de la entrada de productos a actualizar y el usuario con privilegios de administrador pueda modificar uno o varios datos y guardarlo.	4	Administrador	4	3	4 horas	El usuario con privilegios de administrador podrá actualizar los datos de una entrada de producto seleccionando desde la tabla donde se listan todas las entradas registradas que a su vez estarán paginadas.
45	Crear una pantalla donde el usuario con privilegios de administrador pueda ver una lista general de las entradas registradas y pueda filtrarlas por concepto, nombre de bodega o productos específicos.	4	Administrador	5	3	4 horas	El usuario con privilegios de administrador podrá consultar los datos de las entradas desde una lista desplegada en una tabla o su vez tendrá un recuadro en la parte superior de la pantalla donde podrá filtrarla por concepto, nombre de producto o bodega.
46	Crear componente eliminar entrada, donde el usuario con privilegios de administrador pueda seleccionar en un cuadro de dialogo si desea eliminar o no la entrada.	2	Administrador	4	3	2 horas	El usuario con privilegios de administrador pulsará en eliminar desde la tabla donde se muestran todas las entradas también llamada consulta general, luego saldrá un cuadro emergente de diálogo donde podrá confirmar si desea eliminar la entrada.
47	Crear una pantalla donde el administrador podrá registrar los datos asociados a las salidas de productos a una determinada área de la empresa para su uso en producción.	4	Administrador	5	3	4 horas	El usuario con privilegios de administrador podrá registrar salidas de productos en él sistema.

Número	Descripción	Story Point	Rol	Prioridad	Sprint	Tiempo	Criterio de aceptación
48	Crear una pantalla donde se encuentre los datos de la salida de productos a actualizar y el usuario con privilegios de administrador pueda modificar uno o varios datos y guardarlo.	4	Administrador	4	3	4 horas	El usuario con privilegios de administrador podrá actualizar los datos de una salida de producto seleccionando desde la tabla donde se listan todas las salidas registradas que a su vez estarán paginadas.
49	Crear una pantalla donde el usuario con privilegios de administrador pueda ver una lista general de las salidas registradas y pueda filtrarlas por productos específicos, área donde será destinada o nombre de la bodega.	4	Administrador	5	3	4 horas	El usuario con privilegios de administrador podrá consultar los datos de las salidas desde una lista desplegada en una tabla o su vez tendrá un recuadro en la parte superior de la pantalla donde podrá filtrarla por el concepto de salida, nombre del producto o nombre de la bodega.
50	Crear componente eliminar salida, donde el usuario con privilegios de administrador pueda seleccionar en un cuadro de dialogo si desea eliminar o no la salida.	2	Administrador	4	3	2 horas	El usuario con privilegios de administrador pulsará en eliminar desde la tabla donde se muestran todas las salidas también llamada consulta general, luego saldrá un cuadro emergente de diálogo donde podrá confirmar si desea eliminar la entrada.
51	Crear una pantalla donde el usuario con privilegios de administrador u operador pueda actualizar su contraseña.	4	Administrador/Operador	4	3	5 horas	El usuario con privilegios de administrador u operador podrá actualizar su contraseña desde la pestaña de cambiar contraseña.
52	Crear una pantalla donde se pueda ver el Kardex con el resumen de cada producto, constatando las entradas, salidas y el resumen del stock que se tiene a la fecha que se consulta.	5	Administrador	5	3	12 horas	El usuario con privilegios de administrador podrá revisar el stock de los productos, así como ver el resumen de entradas y salidas de los productos.

Número	Descripción	Story Point	Rol	Prioridad	Sprint	Tiempo	Criterio de aceptación
53	Crear la lógica de programación en el backend que permita que cuando se cree o actualice una entrada/salida de un producto el stock se actualice automáticamente y presente el nuevo stock disponible del producto.	5	Administrador	5	3	12 horas	El usuario con privilegios de administrador podrá ingresar o actualizar una entrada/salida de producto y el Kardex se actualizará automáticamente.
54	Crear validaciones para todos los campos de las distintas pantallas creadas.	5	Administrador / Operador	5	3	5 horas	El usuario con privilegios de administrador u operador no podrá registrador datos que no sean válidos, se obligará a llenar los datos en caso de ser necesarios: validar correos, números de teléfono.
55	Crear componentes de alerta para las distintas pantallas.	4	Administrador / Operador	4	3	5 horas	El usuario con privilegios de administrador u operador tendrá una notificación antes de ejecutar una acción.
56	Crear una pantalla donde se puede ver un dashboard con resumen de estadísticas de las principales entidades del sistema.	5	Administrador	5	3	3 horas	El usuario con privilegios de administrador podrá observar un tablero con las principales estadísticas del sistema.

Una vez se ha documentado cada requerimiento en el producto backlog, los cuales fueron obtenidos en la fase previa denominadas historias de usuario, se estimó que el tiempo necesario para que el sistema se encuentre funcional es de 194 horas. Esta cantidad de horas será dividida en tres Sprint, las funcionalidades se irán desarrollando en base a su prioridad desde la más alta a la más pequeña como se indica en el Product backlog.

3.3 Formalización de Requerimientos

3.3.1 Requerimientos Funcionales y No Funcionales

Los requerimientos de un sistema son las funcionalidades o servicios que expondrá el sistema, además nos expresa las restricciones que tendrá el mismo. Se clasifican en

- **Requerimientos funcionales:** De acuerdo con Blancarte Iturralde (2020) los requerimientos funcionales son todos aquellos "que nacen de la necesidad de los usuarios para cubrir un requerimiento de negocio y son solicitados explícitamente" (p. 19), por ejemplo, si se tomamos como objeto de estudio un sistema de gestión de una institución educativa como colegios o universidades, un requisito funcional indispensable será el proceso de matrículas.
- **Requerimientos no funcionales:** Por el contrario de los funcionales, estos no son solicitados explícitamente por el cliente, pero deben ser considerados como cualidades del sistema, entre ellas pueden estar: la seguridad, y su desempeño.

A continuación, en la tabla 5 se establecen los requerimientos funcionales, y su respectivo requerimiento de diseño. En tanto en la tabla 6 se observa los requerimientos no funcionales del producto y su criterio de evaluación.

Finalmente, en la tabla 7 se observa los requerimientos mínimos para el correcto funcionamiento del sistema.

Tabla 5

Requerimientos funcionales

Requerimiento funcional	Requerimientos de diseño
Módulo de autenticación y autorización	Ingreso al sistema
Módulo de gestión de usuarios	Administración de usuarios
Módulo de operadores	Administración de operadores
Módulo de categorías	Administración de categorías
Módulo de unidades	Administración de unidades
Módulo de bodegas	Administración de bodegas
Módulo de proveedores	Administración de proveedores.
Módulo de áreas	Administración de áreas
Módulo de productos	Administración de productos
Módulo de ordenes de trabajo.	Administración de ordenes de trabajo.
Módulo de entradas	Administración de entradas
Módulo de salidas	Administración de salidas
Módulo de Kardex	Administración de Kardex

Tabla 6

Requerimientos no funcionales de producto

Requerimiento no funcional	Criterio de evaluación
Usabilidad	• El sistema debe presentar mensajes informativos al usuario sobre una acción realizada, ya sean de error o informativos. • El sistema debe presentar interfaces intuitivas que disminuyan esfuerzos al usuario al querer completar una acción. • El sistema se adaptará a dispositivos de distintos tamaños de pantalla, aplicando el diseño responsivo, garantizando la correcta visualización desde computadores, teléfonos inteligentes o tabletas.
Eficiencia	• El sistema debe responder adecuadamente a las peticiones del usuario, se busca obtener tiempos de respuesta cortos.

Mantenibilidad	• El sistema será capaz de recibir modificaciones para escalar las funcionalidades o el número de usuarios concurrentes
Seguridad	• El sistema tendrá restricciones de acceso. • Las contraseñas de los usuarios estarán cifradas y no podrán ser observadas por nadie, ni administradores ni acudiendo a registros de la base de datos.

Tabla 7

Requerimientos mínimos

Requerimientos mínimos	Detalle
Hardware	No Aplica
Software	Navegador web que soporte JavaScript, de preferencias últimas versiones de los navegadores.

3.4 Descripción del Sistema

3.4.1 Descripción General

De acuerdo con las necesidades de los usuarios y las funcionalidades que deberá cumplir este sistema, se ha establecido 2 perfiles con diferentes privilegios.

En función del rol que desempeña cada usuario para la empresa V-Imagen:

- **Administrador**: El usuario con privilegios de administrador es en el encargado de las principales funciones del sistema. Ingresar Operadores, Categorías, Unidades, Áreas, Productos, Clientes, Entrada y Salida de Productos, Generar Ordenes de Trabajo, por ello a este usuario se le otorgan todos los permisos sobre los datos.
- **Operador**: El usuario con perfil de operador es encargado de gestionar las ordenes de trabajo y podrán actualizar su estatus en función del trabajo realizado, además tendrán ciertos permisos en cuanto a ingresar, consultar, actualizar clientes, pero no podrán eliminarlos. Este tipo de usuario está asociado a los colaboradores de la empresa.

3.4.2 Funcionalidades Generales del Sistema

Al finalizar el desarrollo del presente sistema, la empresa V-Imagen contará con un sistema web, el cual estará desplegado en la nube y podrá ser accedido desde un dispositivo electrónico con acceso a Internet y que disponga del software requerido para navegar, gracias a estar asociado a un dominio. El sistema web está enfocado en proveer una herramienta moderna que permita automatizar los procesos de la institución otorgando celeridad y protección a los datos. A su vez, mejorará la eficiencia en el manejo de las ordenes de trabajo e inventario que actualmente son manejados de forma manual.

El sistema web permitirá registrar entradas de productos que son adquiridos por un proceso de compra a un proveedor determinado, de igual forma se conocerá con detalle la salida de materia prima de bodega a una determinada área, conociendo así que área ocupa en mayor cantidad cierto tipo de material, de esta manera se aporta con datos que ayuden a tomar las decisiones del negocio.

Por otra parte, el sistema web permitirá registrar y llevar un control de las ordenes de trabajo o también llamados tiques, asignando una orden a determinado operador y obtener reportes que permitan observar el desempeño de la empresa.

3.4.3 Funcionalidades enfocadas al Administrador

El sistema le presentará un dashboard completo al administrador con todas las principales funcionalidades del sistema. La pantalla principal contendrá estadísticas generales de la empresa. En la barra lateral izquierda se encontrará los distintos botones que permitan acceder a las distintas funcionalidades, entre las que se encuentran.

- Ingreso al Sistema
- Operador: Ingresar, Actualizar, Consultar y Eliminar
- Bodega: Ingresar, Actualizar, Consultar y Eliminar

- Proveedor: Ingresar, Actualizar, Consultar y Eliminar
- Unidad: Ingresar, Actualizar, Consultar y Eliminar
- Producto: Ingresar, Actualizar, Consultar y Eliminar
- Área: Ingresar, Actualizar, Consultar y Eliminar
- Cliente: Ingresar, Actualizar, Consultar y Eliminar
- Entrada de Productos: Ingresar, Actualizar, Consultar y Eliminar
- Salida de Productos: Ingresar, Actualizar, Consultar y Eliminar
- Kardex: Consultar
- Categoría: Ingresar, Actualizar, Consultar y Eliminar
- Orden de trabajo: Ingresar, Actualizar, Consultar y Eliminar
- Recuperar contraseña
- Tablero de estadísticas

3.4.4 Funcionalidades enfocadas al Operador

El operador es la persona colaboradora de la empresa, su perfil le otorgará ciertos permisos sobre determinadas funcionalidades que pueda ejecutar. Entra las funcionalidades enfocadas al Operador se encuentran:

- Ingreso al Sistema
- Actualizar contraseña
- Cliente: Ingresar, Actualizar, Consultar
- Orden de Trabajo: Actualizar, Consultar
- Producto: Consultar
- Recuperar contraseña

4 CAPÍTULO IV: DISEÑO

El objetivo de esta sección es desarrollar los diagramas de casos de uso, modelo de base de datos y arquitectura de sistema que permitirá tener una idea de cómo se integrará los distintos componentes del sistema web.

4.1 Casos de Uso

Un caso de uso es “un artefacto que define una secuencia de acciones que da lugar a un resultado de valor observable. Los casos de uso proporcionan una estructura para expresar requisitos funcionales en el contexto de procesos empresariales y de sistema.” (IBM, 2019b)

Los casos de usos son un tipo de diagrama que capturan los requisitos funcionales a desarrollar, en él se representa los actores externos que harán uso de la funcionalidad que se conectan con el requisito funcional mediante líneas de comunicación.

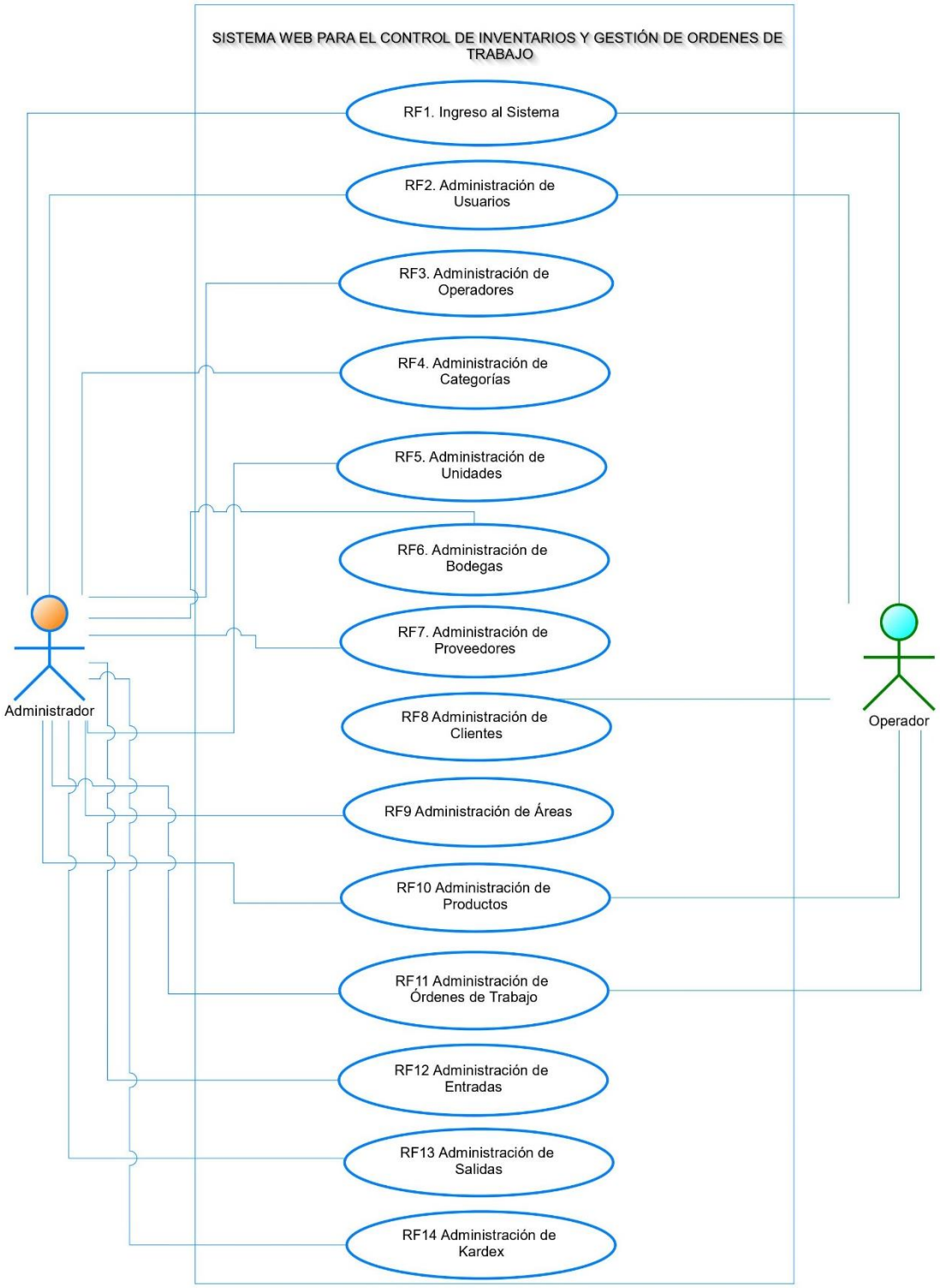
De acuerdo con los requerimientos funcionales descritos en la tabla 5, los diagramas de caso de uso del sistema web son:

4.1.1 Nivel General

El diagrama de casos de uso a nivel general describe todos los requisitos funcionales del sistema y los actores que intervienen con cada uno de estos requisitos.

Figura 10

Casos de uso a nivel general



4.1.2 Siguiente Nivel

Los diagramas de siguiente nivel presentan con detalle cada una de las funcionalidades y sus principales operaciones a nivel de datos que fueron presentadas en el nivel general.

Figura 11

RF1. Ingreso al sistema

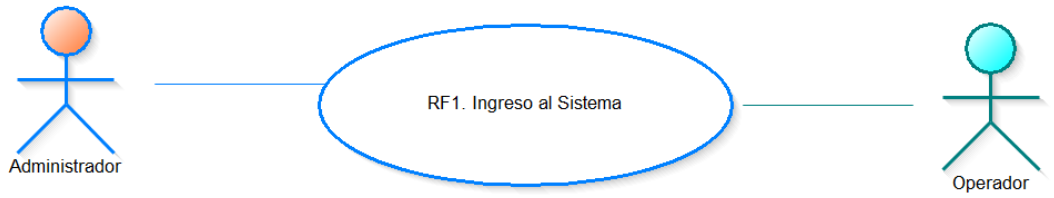


Figura 12

RF2. Administración de usuarios

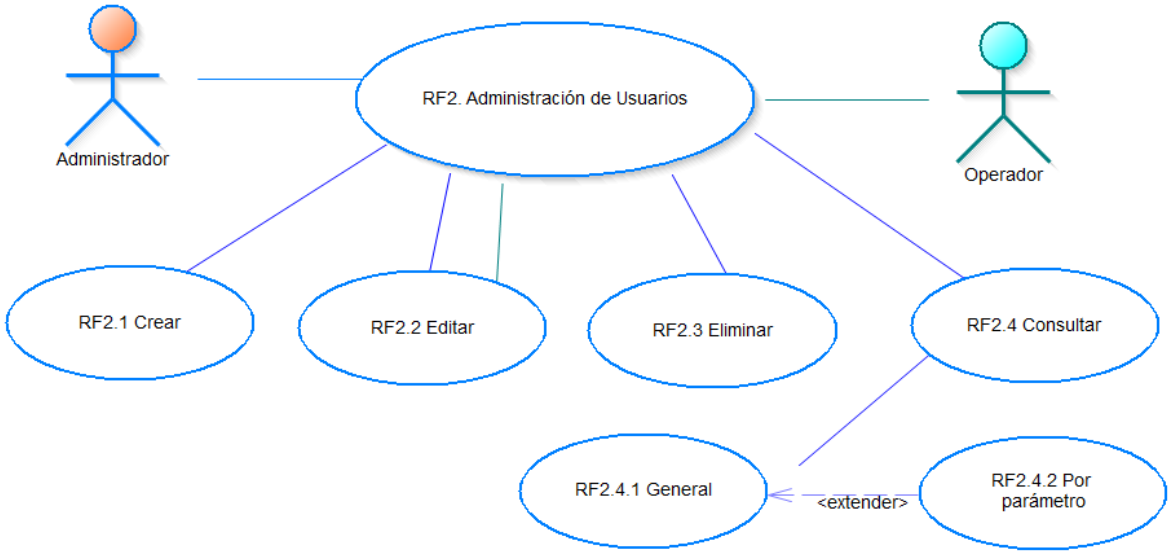


Figura 13

RF3. Administración de operadores

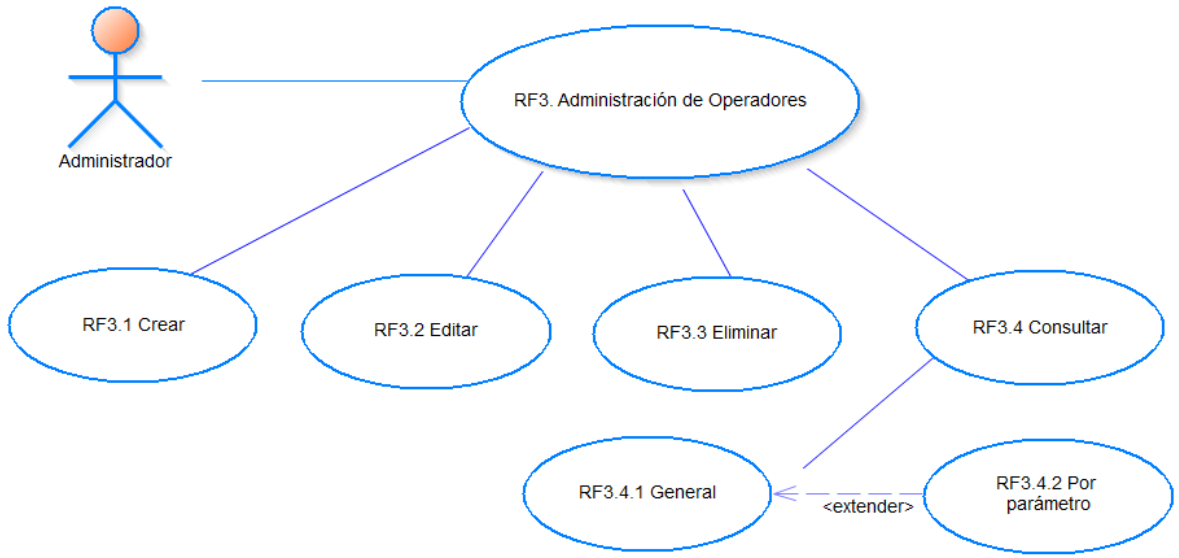


Figura 14

RF4. Administración de categorías

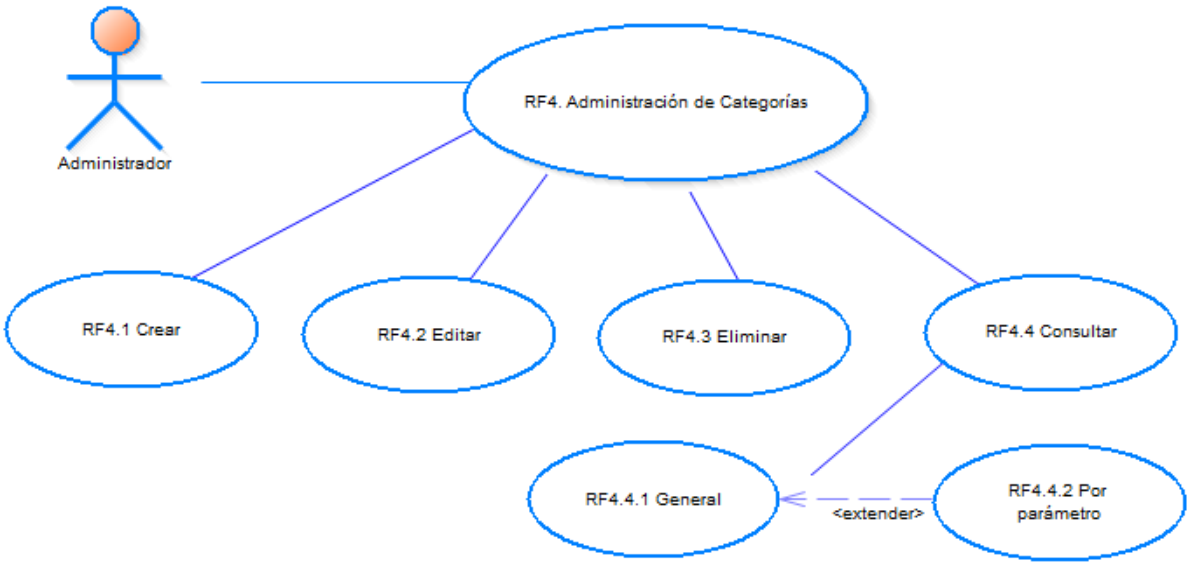


Figura 15

RF5. Administración de unidades

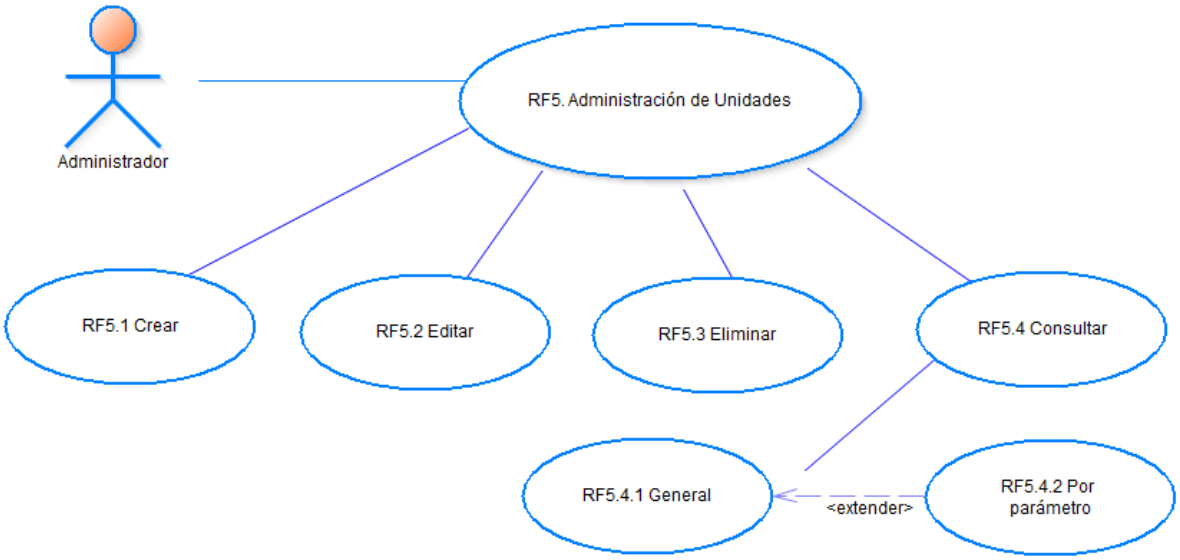


Figura 16

RF6. Administración de bodegas

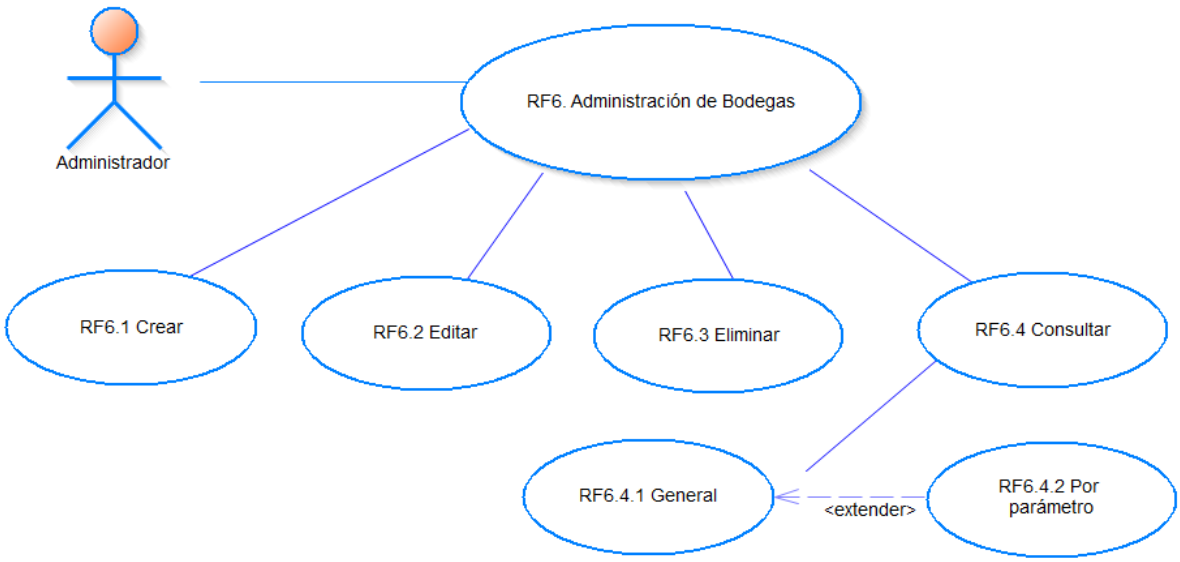


Figura 17

RF7. Administración de proveedores

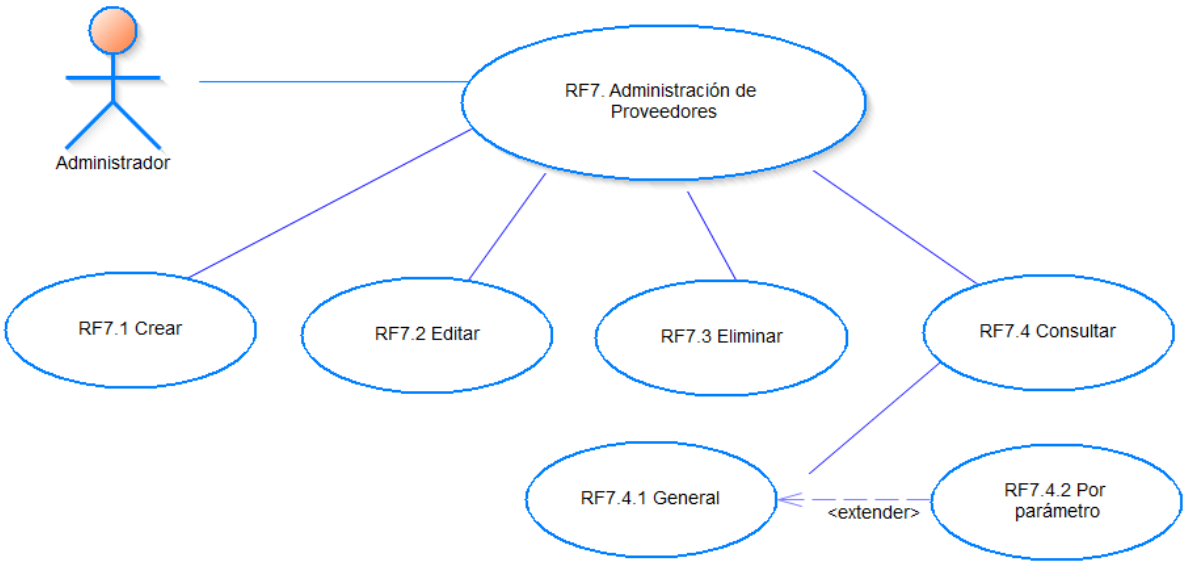


Figura 18

RF8. Administración de clientes

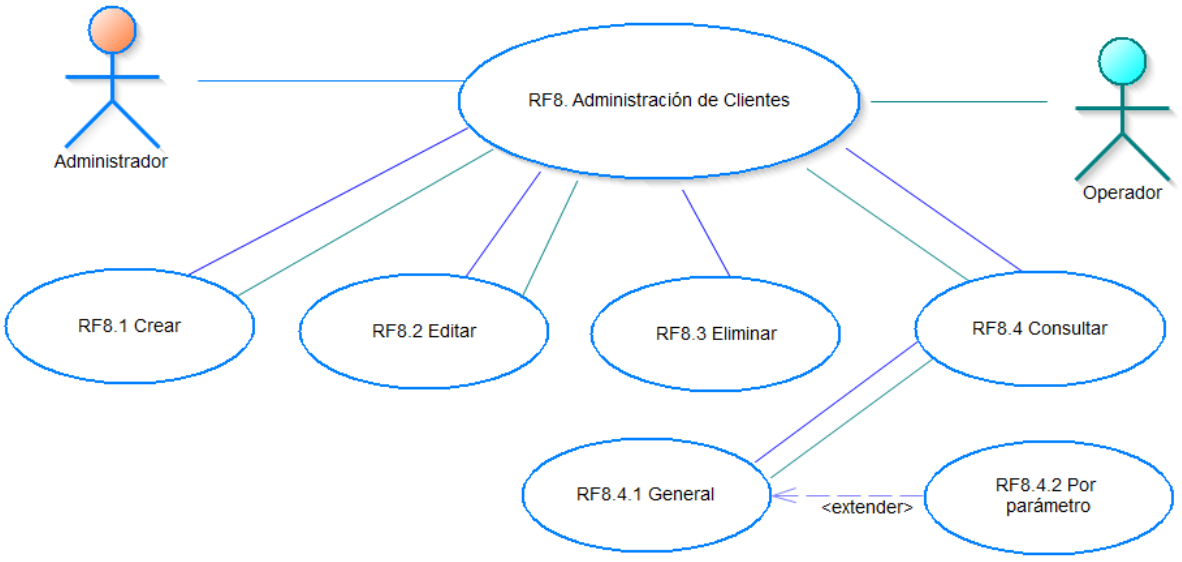


Figura 19

RF9. Administración de áreas

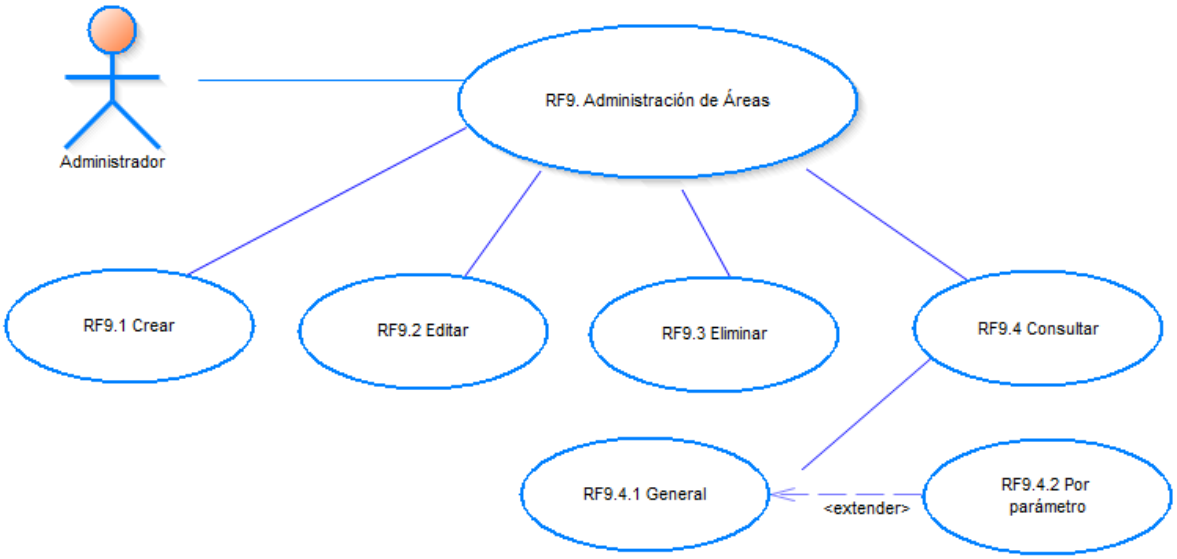


Figura 20

RF10. Administración de productos

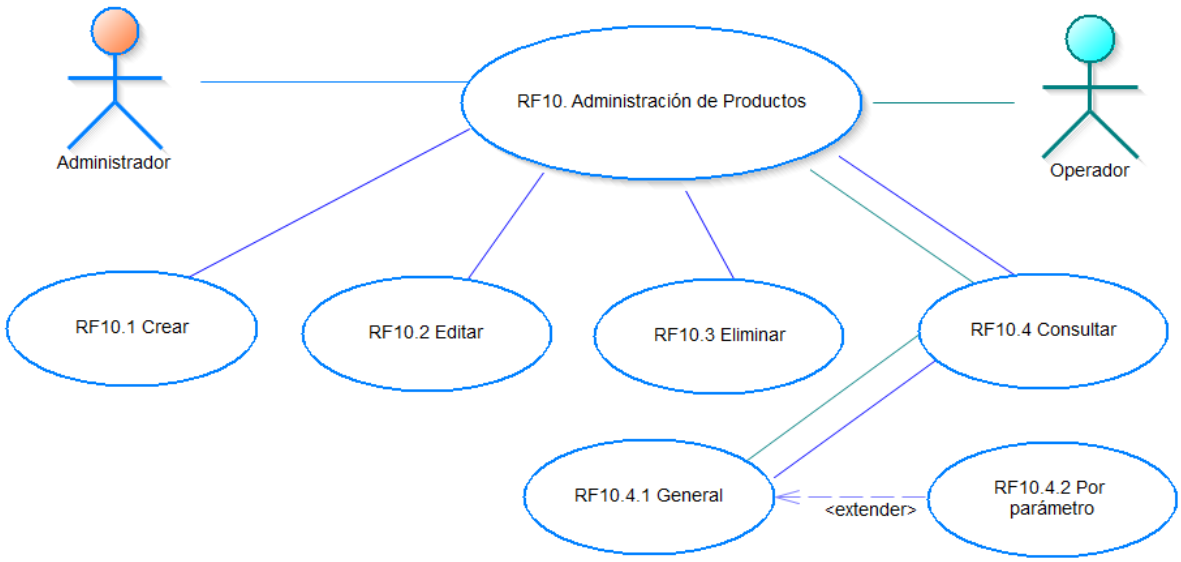


Figura 21

RF11. Administración de ordenes de trabajo

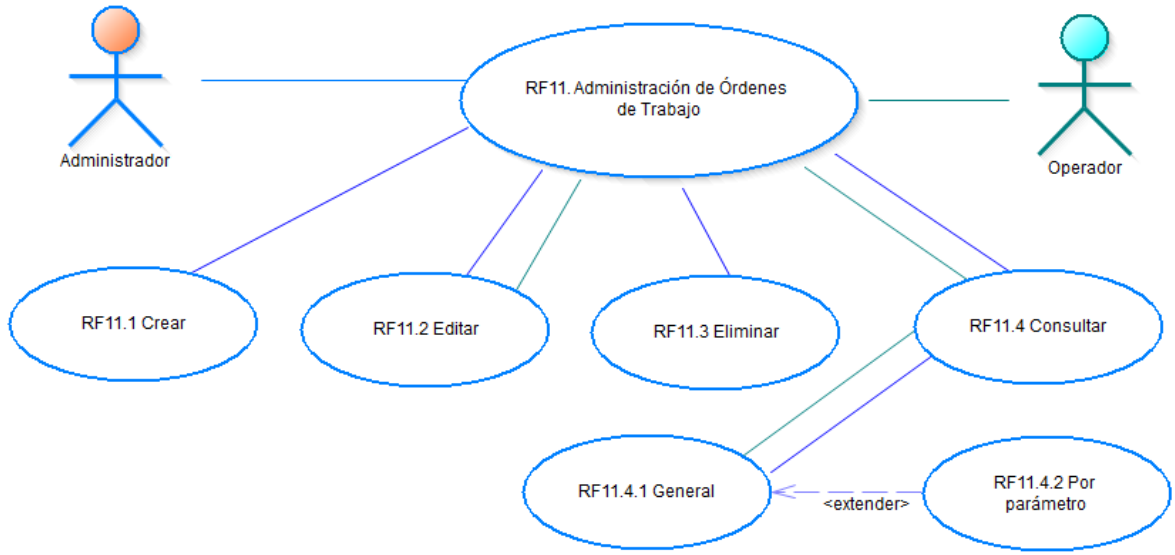


Figura 22

RF12. Administración de entradas

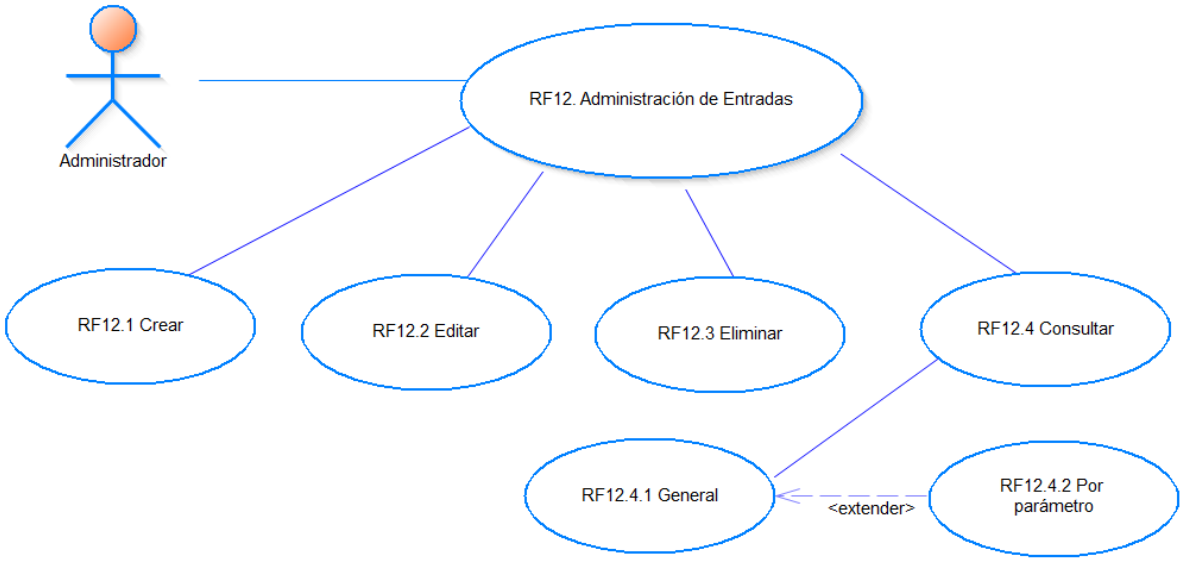


Figura 23

RF13. Administración de salidas

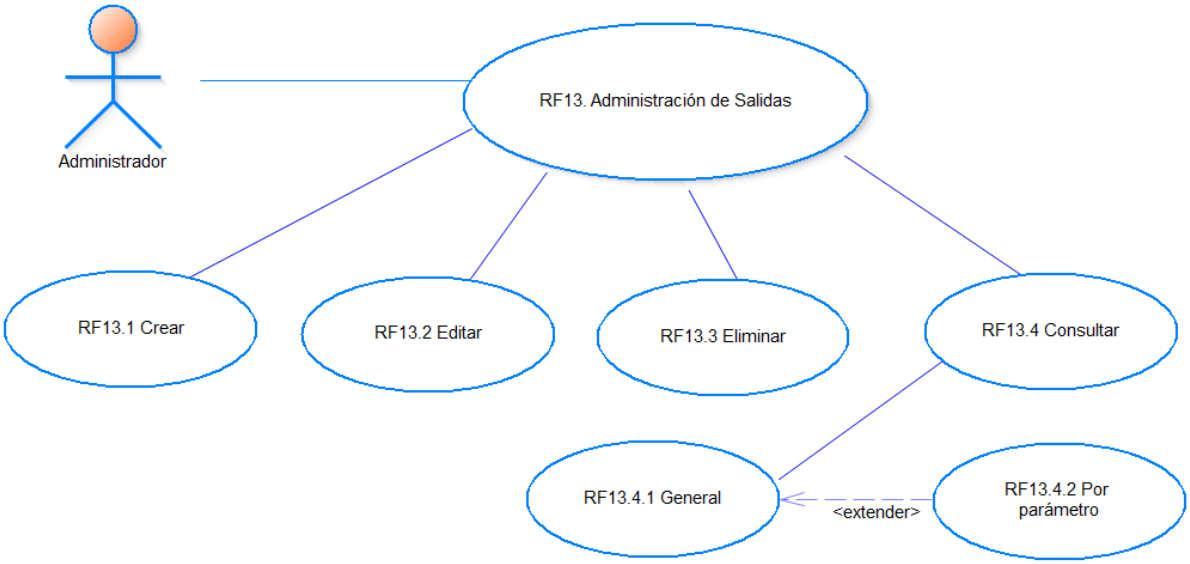
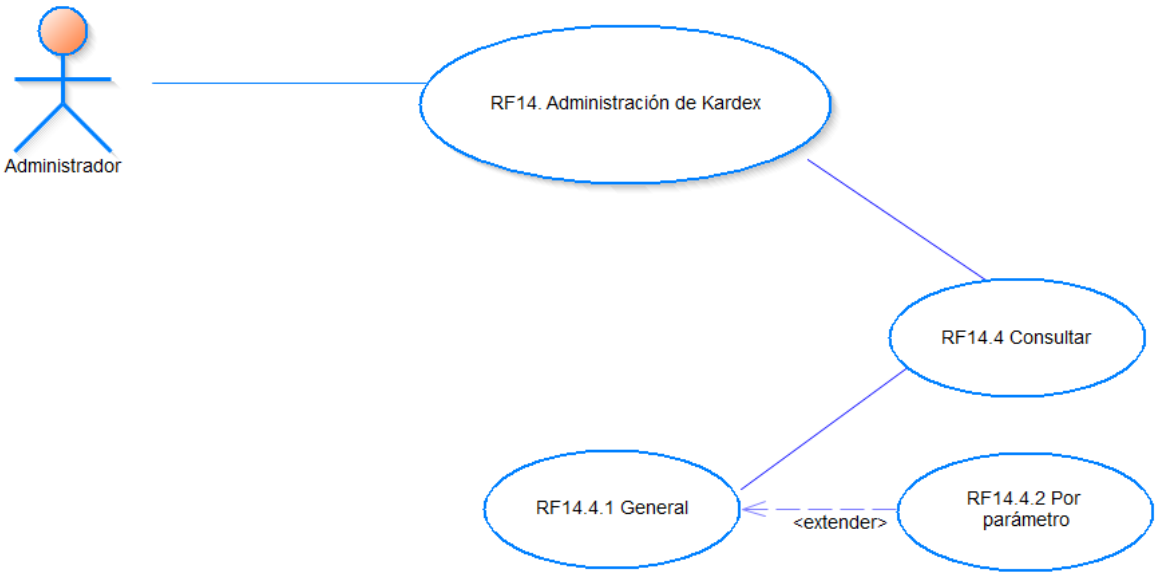


Figura 24

RF14. Administración de Kardex



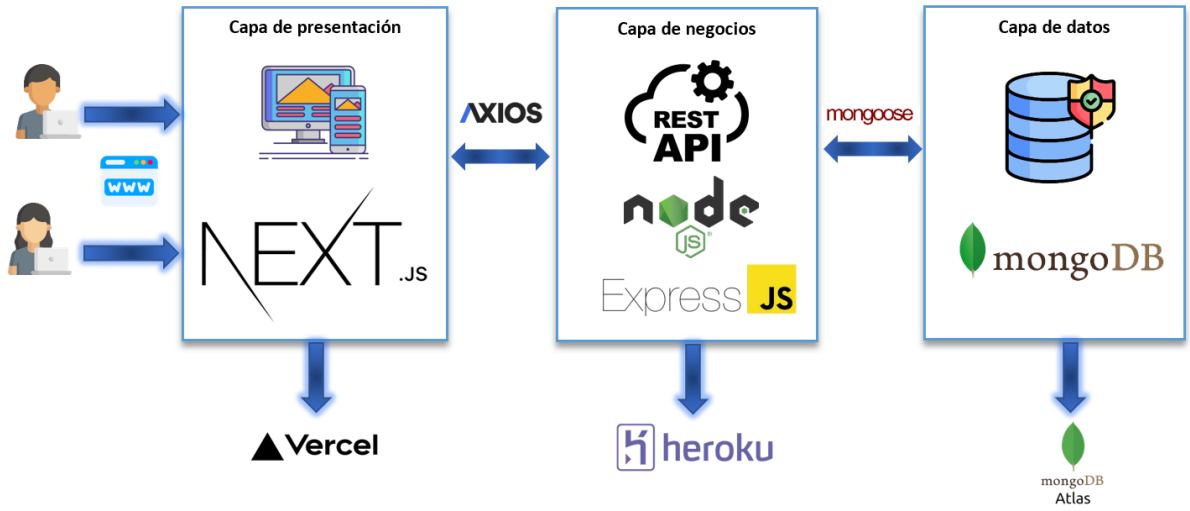
4.2 Arquitectura del sistema

La arquitectura de software establecida para el presente desarrollo web será la de capas, siendo compuesta por la capa de presentación (interfaz gráfica), la capa de negocios (lógica del negocio) y la capa de datos (acceso a datos). La arquitectura puede ser representada por el Frontend asociada a la capa de presentación, el Backend asociada a la capa de negocio y acceso a datos. La comunicación entre el Frontend y el Backend a través del API REST.

La figura 25 muestra la arquitectura del sistema web junto con las herramientas utilizadas para el proceso de desarrollo del sistema web.

Figura 25

Arquitectura del sistema



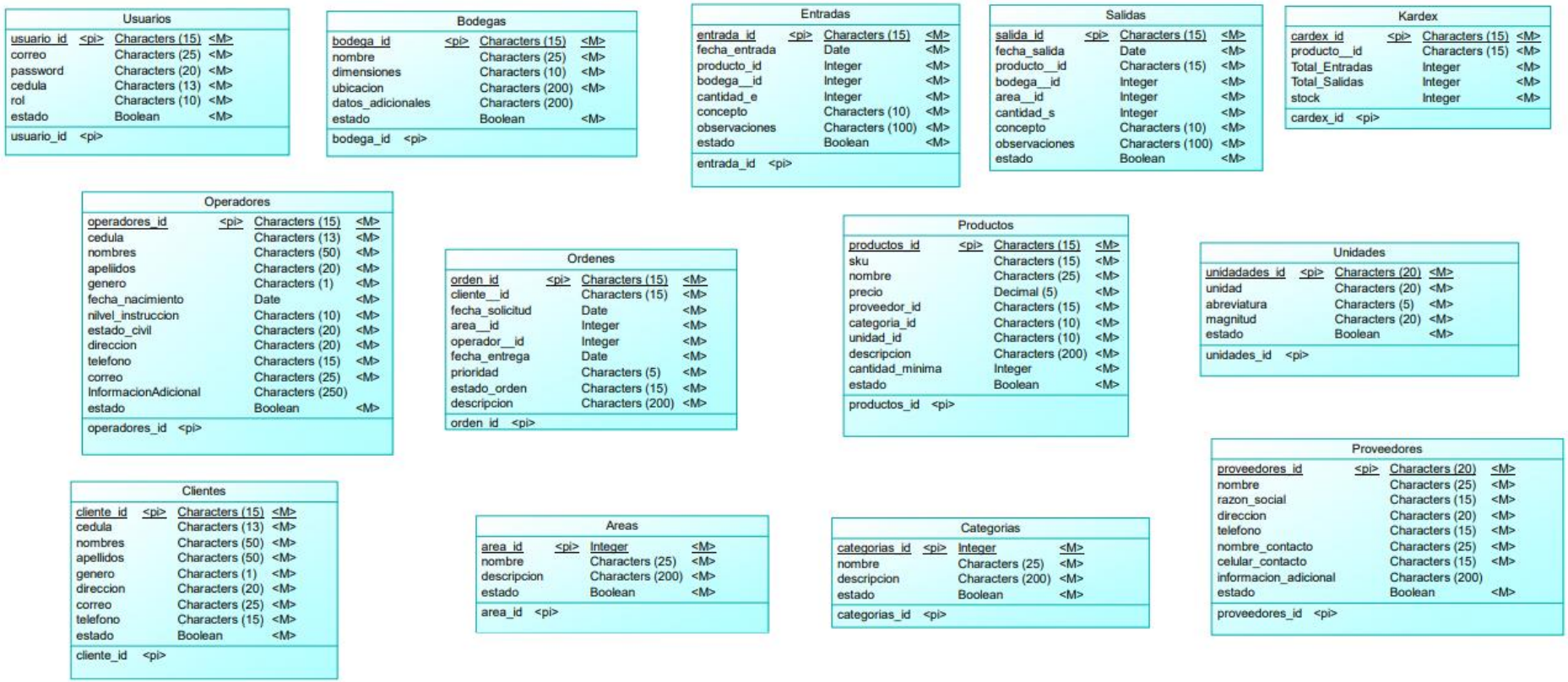
4.3 Base de Datos

El modelo de base de datos a continuación se ha construido basado en los requisitos y necesidades del usuario descrito en las historias de usuario. El diagrama no tiene relaciones puesto que se usará una base de datos no relacional que guarda los registros en forma de objeto.

La figura 26 muestra el diagrama de base de datos que se usará para la construcción del sistema.

Figura 26

Modelo de base de datos



5 CAPÍTULO V: DESARROLLO Y PRUEBAS

El objetivo de esta sección es construir de forma iterativa los requisitos funcionales en producto. El desarrollo del sistema web se ha dividido en tres Sprint.

5.1 Sprint 1

5.1.1 Planificación del Sprint

En base a la planificación realizada para el sprint 1 se tiene las siguientes historias de usuario descritas en la tabla 3. En la tabla 8 se puede observar un resumen de las historias de usuario que serán construidas en el Sprint 1. Es necesario introducir el concepto de épica, que es una forma que Scrum tiene para llamar a los grandes módulos que por su tamaño deben ser descompuestos en varias historias de usuario. Llamemos épica al módulo de usuarios que se descompone en la historia asociada a la creación de usuario, una historia asociada a la actualización y así como se requiera en función de los requerimientos.

Tabla 8

Resumen de historias de usuario de sprint 1

Número	Nombre de Historia	Rol
1	Ingreso al Sistema	Administrador/Operador
2	Registro de Usuarios	Administrador
3	Actualizar Usuarios	Administrador
4	Consulta por Parámetros	Administrador
5	Eliminar Usuarios	Administrador
6	Reestablecer contraseña	Administrador/Operador

Número	Nombre de Historia	Rol
7	Registro de Operadores	Administrador
8	Actualizar Operadores	Administrador
9	Consulta General y por Parámetros de Operadores	Administrador
10	Eliminar Operadores	Administrador
11	Registrar Bodegas	Administrador
12	Actualizar Datos de Bodegas	Administrador
13	Consulta General y por Parámetros de Bodegas	Administrador
14	Eliminar Bodegas	Administrador
15	Registrar Proveedores	Administrador
16	Actualizar Proveedores	Administrador
17	Consulta General y por Parámetros de Proveedores	Administrador
18	Eliminar Proveedores	Administrador
19	Registrar Unidades	Administrador
20	Actualizar Unidades	Administrador
21	Consulta General y por Parámetros de Unidades	Administrador
22	Eliminar Unidades	Administrador
23	Registrar Áreas	Administrador
24	Actualizar Áreas	Administrador

Número	Nombre de Historia	Rol
25	Consulta General y por Parámetros de Áreas	Administrador
26	Eliminar Área	Administrador
27	Registrar Cliente	Administrador/Operador
28	Actualizar Cliente	Administrador/Operador
29	Consulta General y por Parámetros de Clientes	Administrador/Operador
30	Eliminar Cliente	Administrador

5.1.2 Sprint backlog

El sprint backlog correspondiente al sprint 1 contempla en su mayoría tablas bases que servirán para construir los procesos de negocio. Se puede ver los requerimientos a trabajar en la tabla 8.

5.1.3 Duración del Sprint

El sprint 1 en base a la planificación y estimación del proceso de desarrollo tendrá una duración de 88 horas de trabajo.

5.1.4 Desarrollo

Para el desarrollo se empezará construyendo los Endpoints en el Backend y luego las pantallas asociadas a la historia de usuario.

5.1.4.1 Backend: Resumen de épica de Usuarios

En la tabla 9 se muestran los recursos del backend para las historias de usuarios 1,2,3,4 y 5 correspondiente a la épica de Usuarios.

Tabla 9

Historia de usuario 1, 2, 3, 4, 5 y 6. Endpoints

Historia de usuario	Método	URI	Descripción
1	POST	/api/auth/login	Autenticación de usuarios
2	POST	/api/usuarios	Esté método nos permite registrar un nuevo usuario, con sus datos asociados. La contraseña predeterminada será el número de cédula del usuario, que posteriormente podrá ser cambiada, las acciones de crear se realizan desde el perfil de administrador.
3.1	GET	/api/usuarios/id	Este método permite obtener un usuario por su id.
3.2	GET	/api/buscar/usuarios/valor	Este método permite buscar un usuario por sus nombres o correo registrado desde el perfil de administrador.
4	PUT	/api/usuarios/id	Este método permite actualizar los datos de un usuario incluida su contraseña desde el perfil de administrador.
5	DELETE	/api/usuarios/id	Este método permite eliminar de manera lógica un usuario desde el perfil de administrador.
6	POST	/api/auth/recovery	Este método permite al usuario reestablecer su contraseña desde la cuenta de un administrador u operador.

5.1.4.2 Frontend: Historia de usuario 1

La historia de usuario 1 corresponde a la construcción del inicio de sesión para los usuarios con perfil administrador u operador.

Figura 27

Historia de usuario 1, iniciar sesión

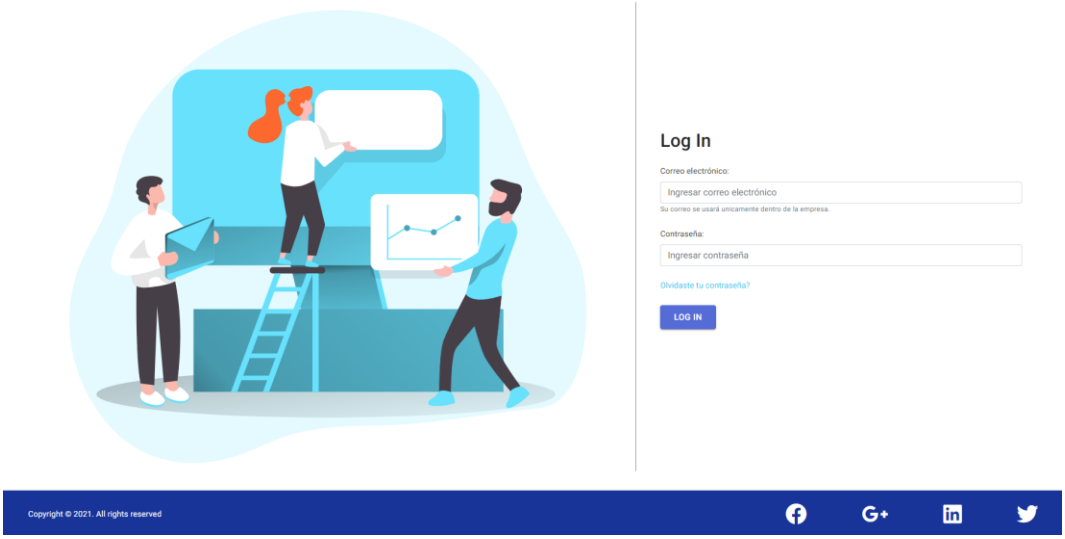
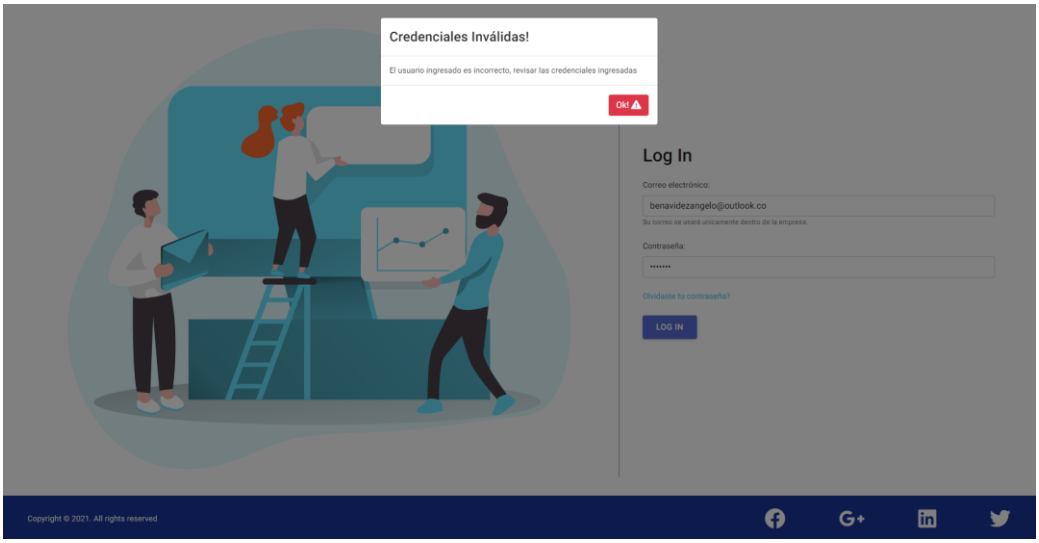


Figura 28

Historia de usuario 1, iniciar sesión: credenciales inválidas.

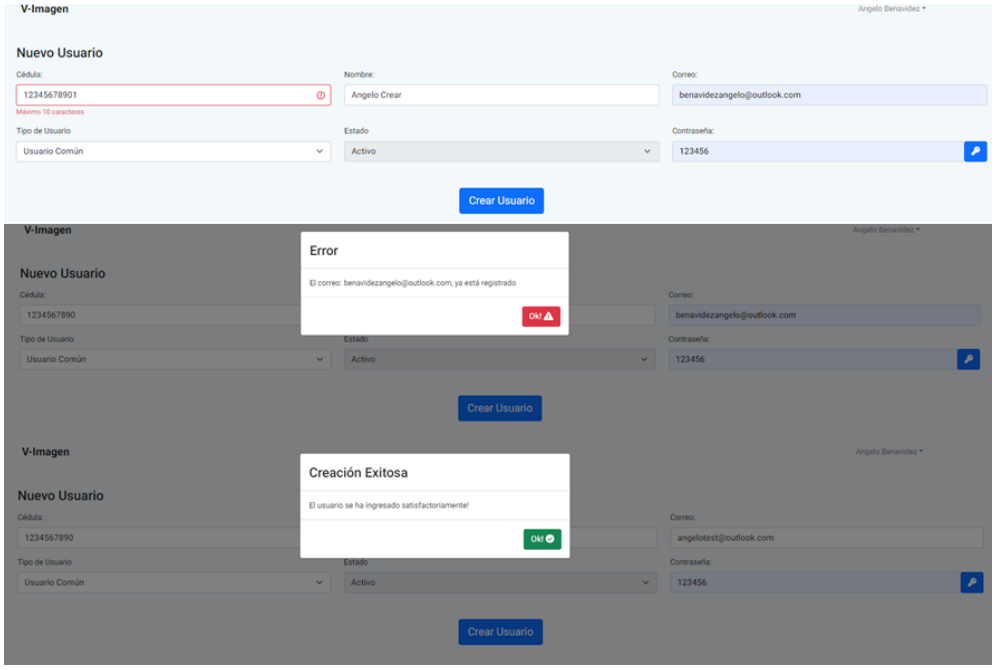


5.1.4.3 Frontend: Historia de usuario 2

La historia de usuario 2 corresponde a la creación de nuevos usuarios, esta acción es realizada por parte del administrador.

Figura 29

Historia de usuario 2, crear usuarios

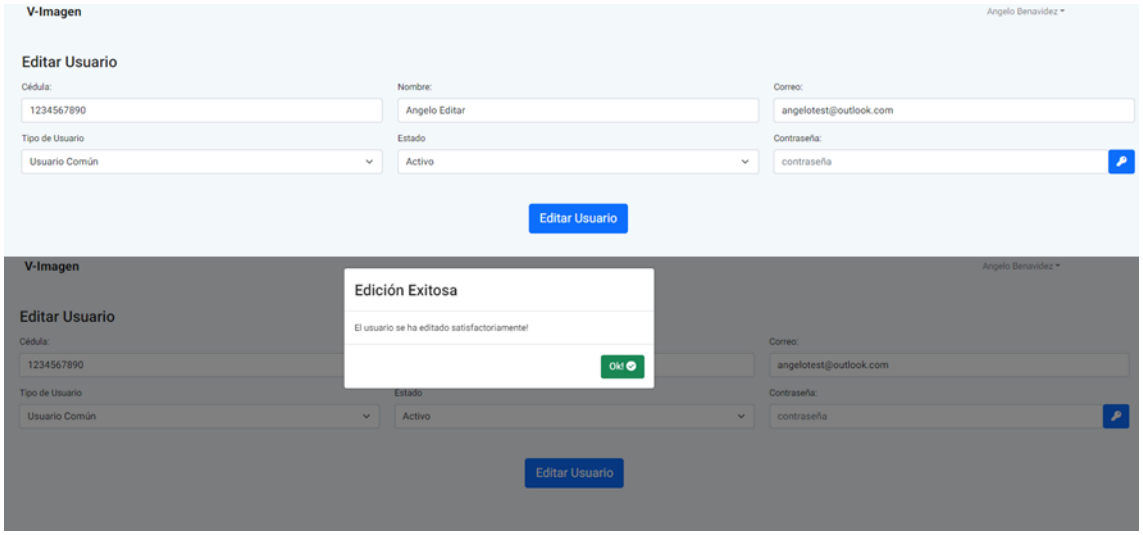


5.1.4.4 Frontend: Historia de usuario 3

La historia de usuario 3 corresponde a la actualización usuarios, esta acción es realizada por parte del administrador.

Figura 30

Historia de usuario 3, actualizar usuarios

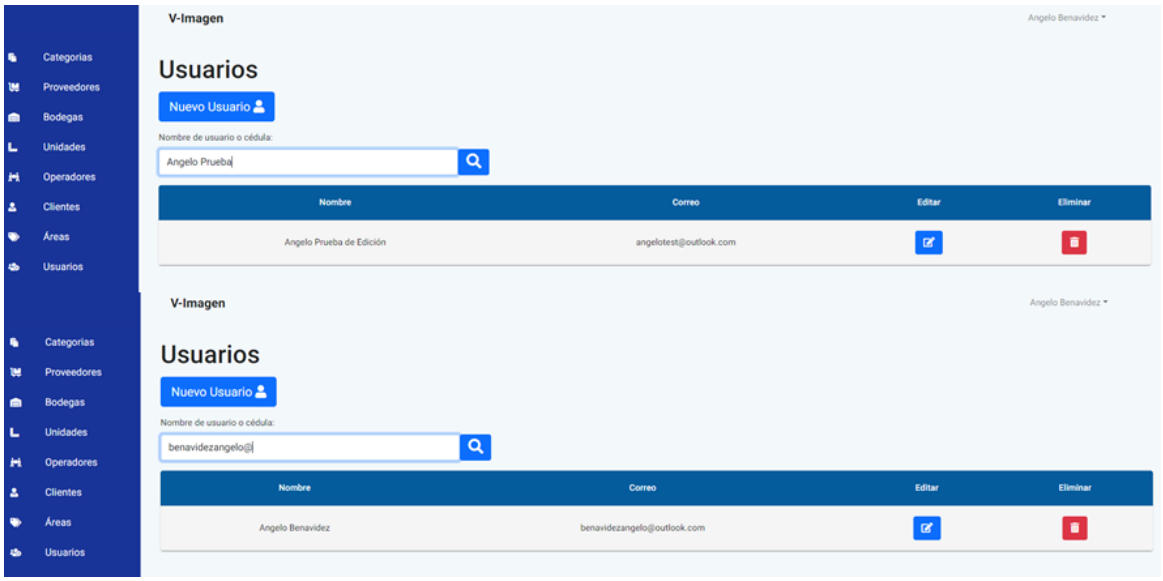


5.1.4.5 Frontend: Historia de usuario 4

La historia de usuario 4 corresponde a la consulta por parámetros de usuarios, esta acción es realizada por parte del administrador.

Figura 31

Historia de usuario 4, consultar usuarios por parámetros

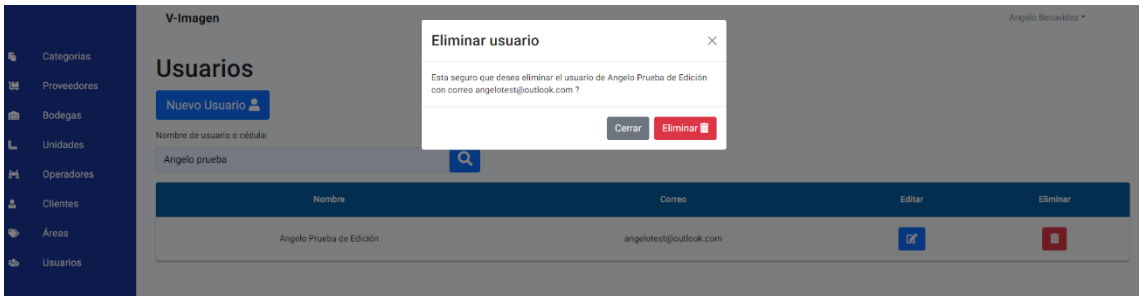


5.1.4.6 Frontend: Historia de usuario 5

La historia de usuario 5 corresponde a la eliminación de usuarios, esta acción es realizada por parte del administrador.

Figura 32

Historia de usuario 5, eliminar usuarios



5.1.4.7 Frontend: Historia de usuario 6

La historia de usuario 6 corresponde a reestablecer la contraseña de usuarios, esta acción es realizada un administrador u operador. Es necesario realizar tres pasos descritos en las figuras 33, 34 y 35.

Figura 33

Historia de usuario 6, paso 1 para reestablecer contraseña

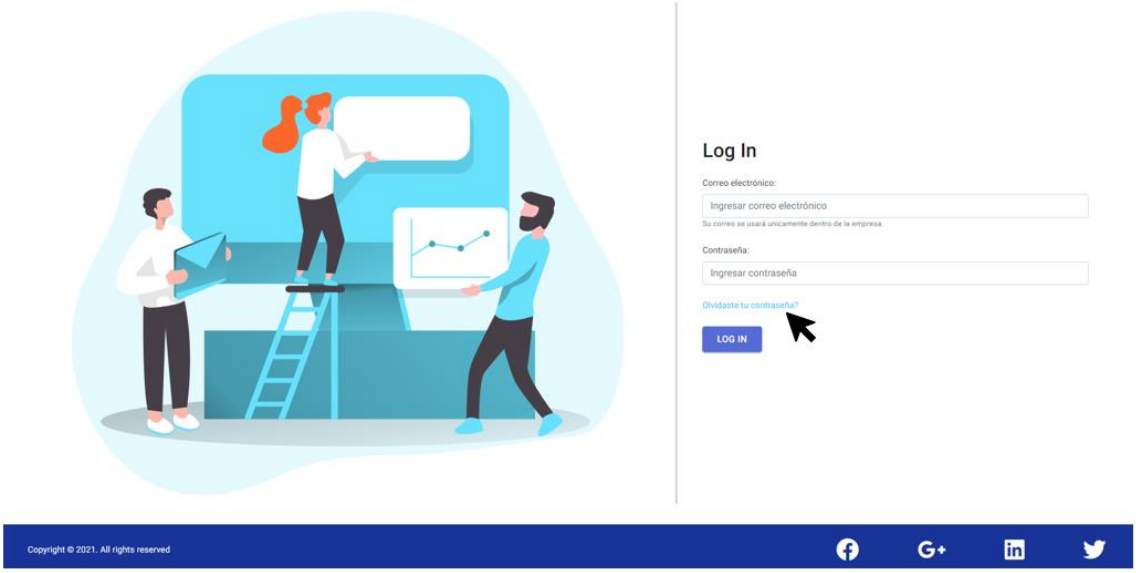


Figura 34

Historia de usuario 6, paso 2 para reestablecer contraseña

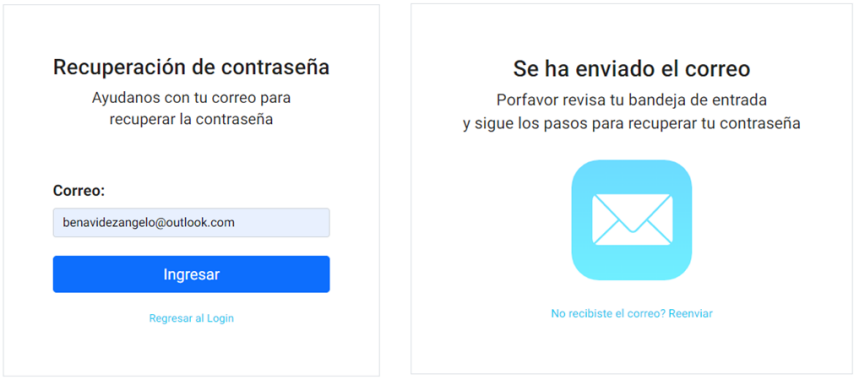
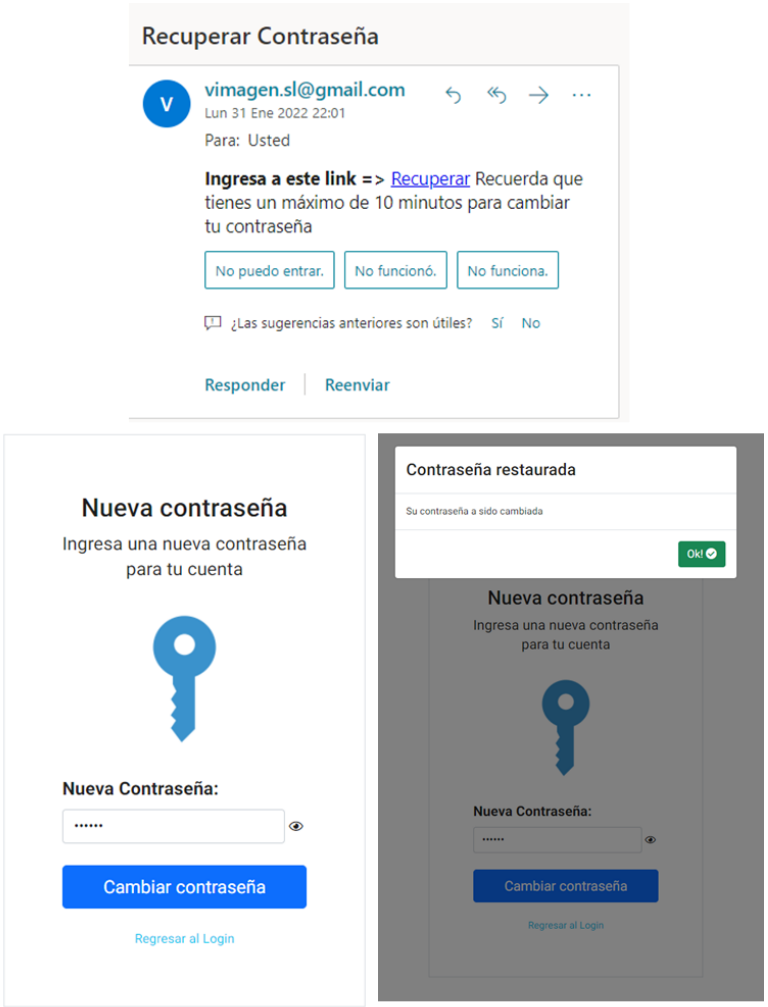


Figura 35

Historia de usuario 6, paso 3 para reestablecer contraseña



5.1.4.8 Backend: Resumen de épica de Operadores

En la tabla 10 se muestran los recursos del backend para las historias de usuarios 7, 8, 9 y 10 correspondiente a la épica de Operadores.

Tabla 10

Historia de usuario 7, 8, 9 y 10. Endpoints

Historia de usuario	Método	URI	Descripción
7	POST	/api/operadores	Esté método nos permite registrar un nuevo operador, con sus datos asociados desde el perfil de administrador.

8	PUT	/api/operadores	Este método permite actualizar los datos de un operador desde el perfil de administrador.
9.1	GET	/api/operadores/ /api/operadores/id	El primer método permite obtener una lista de operadores. El segundo método permite obtener un operador por su id.
9.2	GET	/api/buscar/operadores/valor	Este método permite buscar un operador por su cédula, nombres, apellidos o correo registrado.
10	DELETE	/api/operadores/id	Este método permite eliminar de manera lógica un operador desde el perfil de administrador.

5.1.4.9 Frontend: Historia de usuario 7

La historia de usuario 7 corresponde a la creación de nuevos operadores, esta acción es realizada por parte del administrador.

Figura 36

Historia de usuario 7, validaciones para crear operador

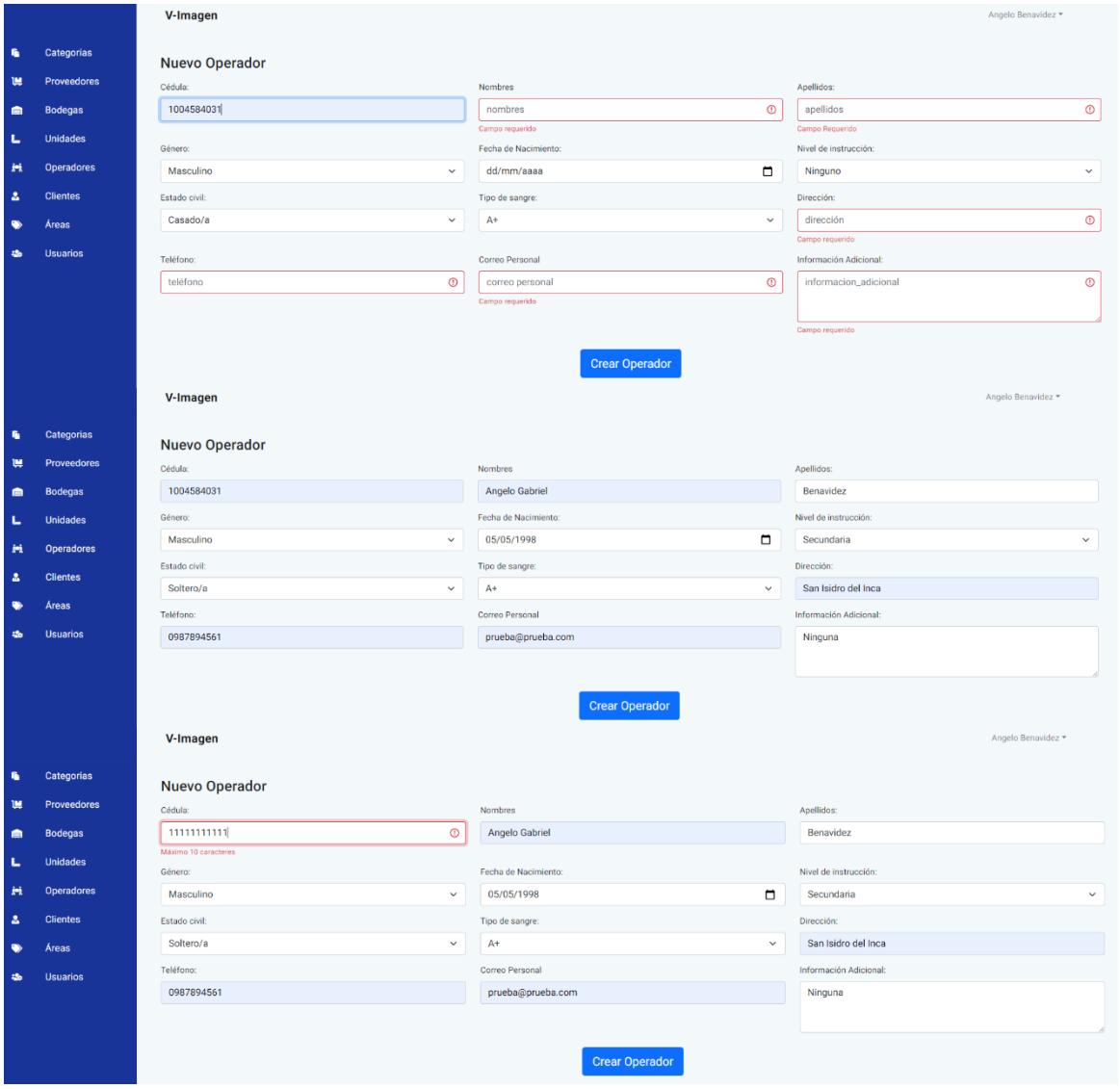
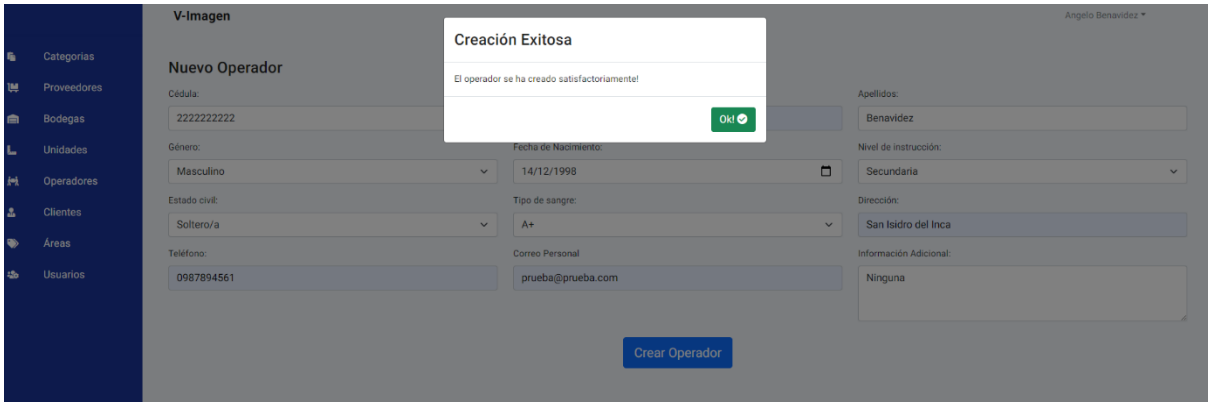


Figura 37

Historia de usuario 7, crear operador



5.1.4.10 **Frontend:** Historia de usuario 8

La historia de usuario 8 corresponde a la actualización de operadores, esta acción es realizada por parte del administrador.

Figura 38

Historia de usuario 8, actualizar operador: seleccionar.

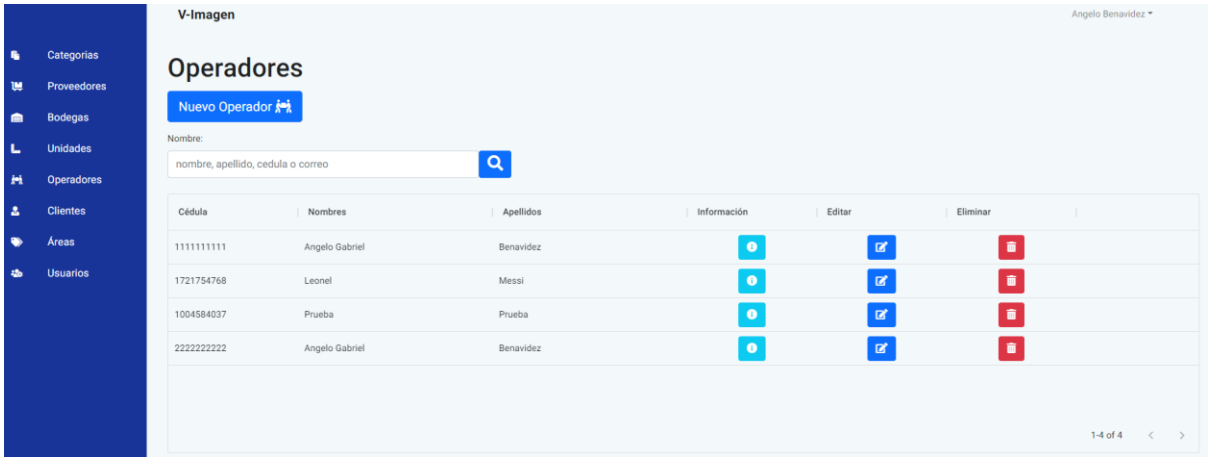
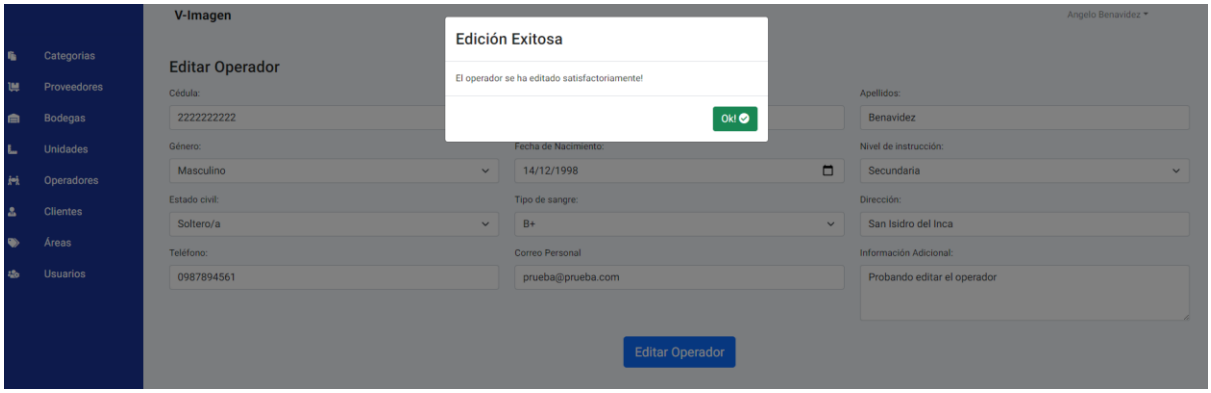


Figura 39

Historia de usuario 8, actualizar operador: editar campos.



5.1.4.11 **Frontend:** Historia de usuario 9

La historia de usuario 9 corresponde a la consulta general y por parámetros de operadores, esta acción es realizada por parte del administrador.

Figura 40

Historia de usuario 9, consulta general operador

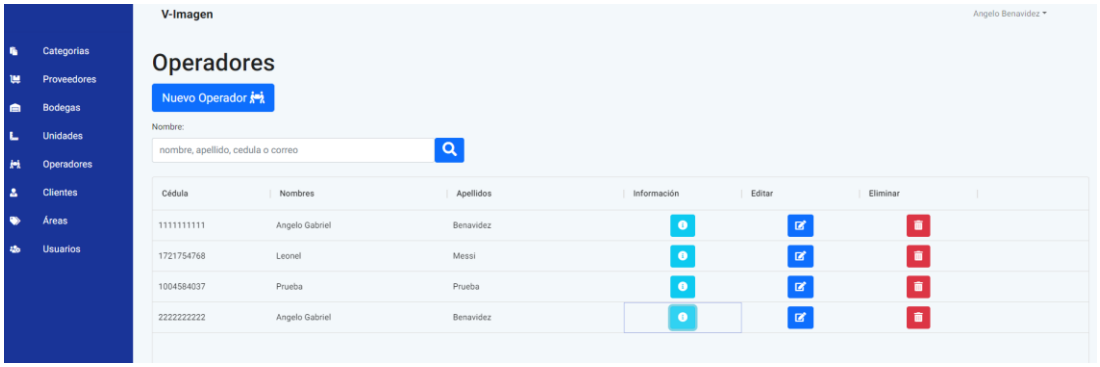


Figura 41

Historia de usuario 9, consulta detallada de operador

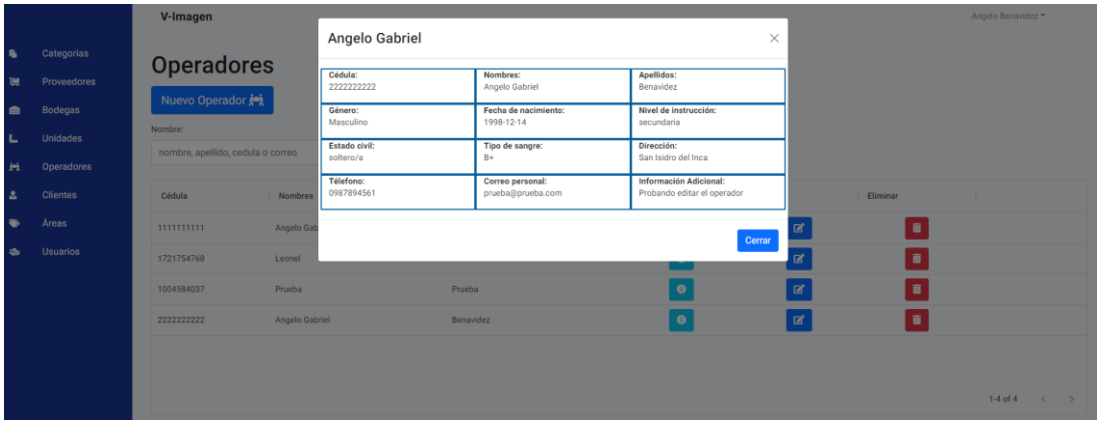
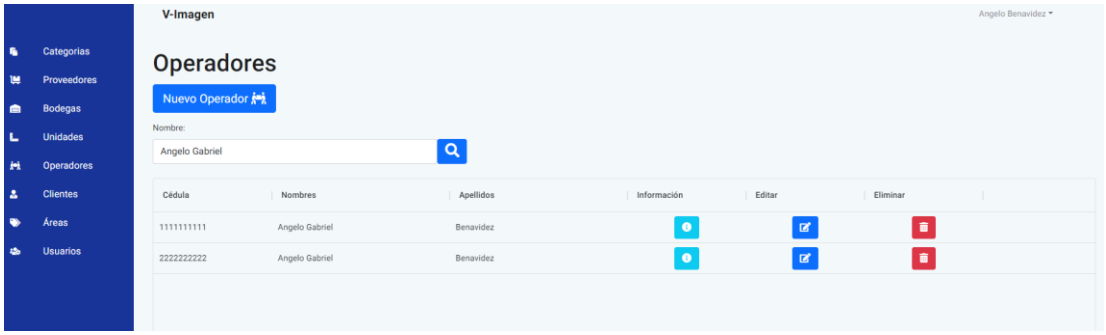


Figura 42

Historia de usuario 9, consulta por parámetro de operador

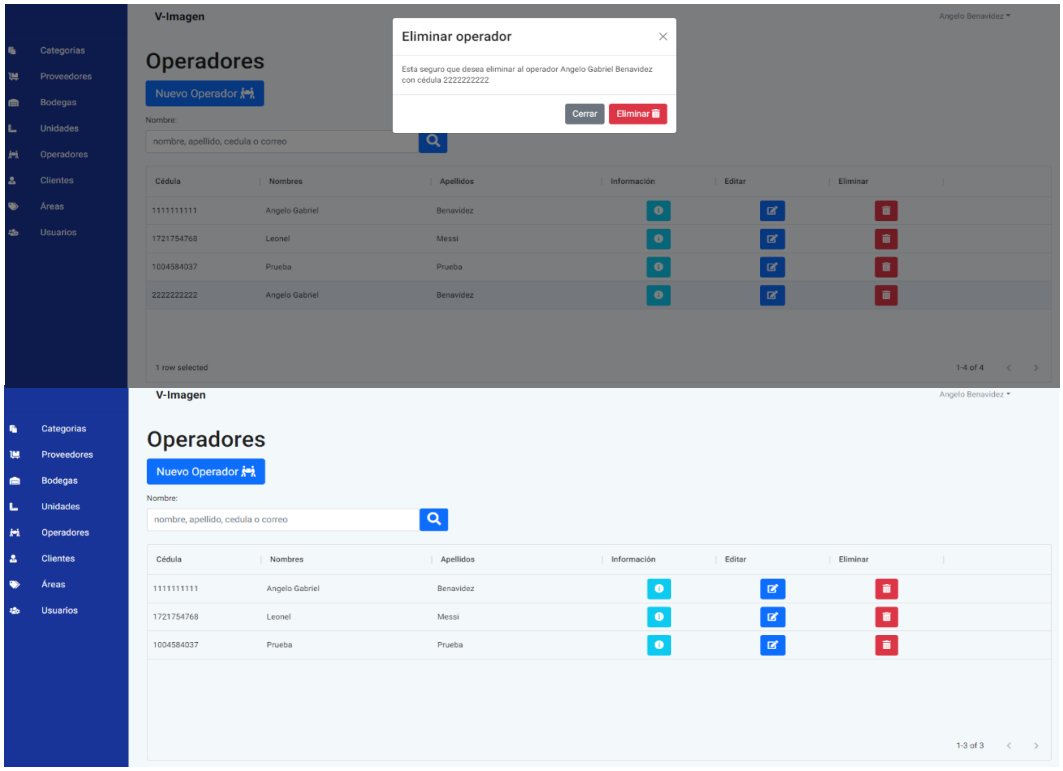


5.1.4.12 Historia de usuario 10

La historia de usuario 10 corresponde a la eliminación de un operador, esta acción es realizada por parte del administrador.

Figura 43

Historia de usuario 10, eliminar un operador



5.1.4.13 **Backend:** Resumen épica de Bodegas

En la tabla 11 se muestran los recursos del backend para las historias de usuarios 11, 12, 13 y 14 correspondiente a la épica de Bodegas.

Tabla 11

Historia de usuario 11, 12, 13 y 14: Endpoints

Historia de usuario	Método	URI	Descripción
11	POST	/api/bodegas	Esté método nos permite registrar una nueva bodega, con sus datos asociados desde el perfil de administrador.
12	PUT	/api/bodegas	Este método permite actualizar los datos de una bodega desde el perfil de administrador.
13.1	GET	/api/bodegas/ /api/bodegas/id	El primer método permite obtener una lista de las bodegas disponibles. El segundo método permite obtener una bodega por su id.

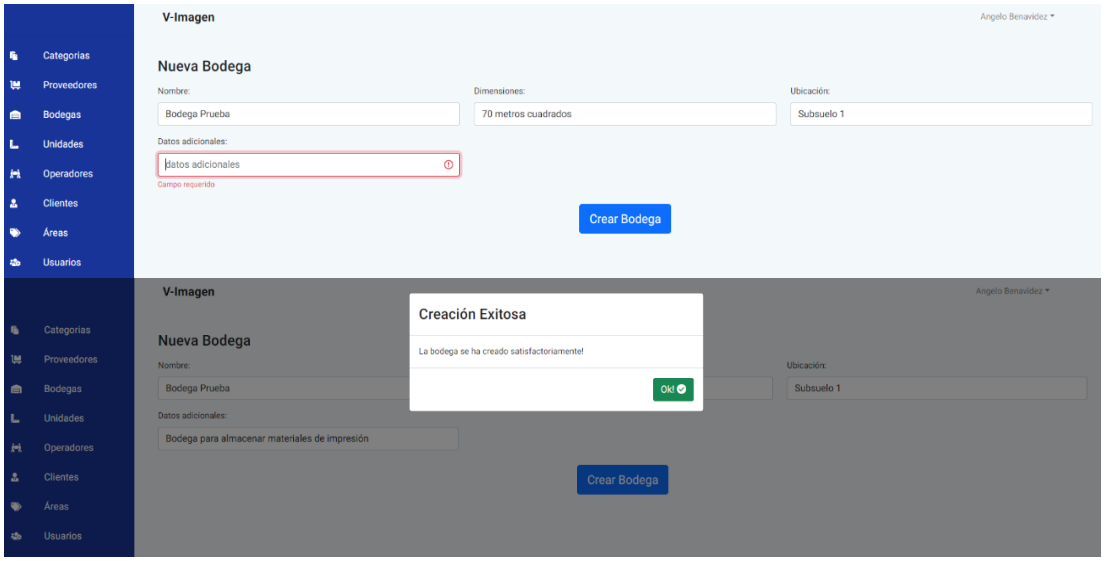
13.2	GET	/api/buscar/bodegas/valor	Este método permite buscar una bodega por su nombres, ubicación o dimensión.
14	DELETE	/api/bodegas/id	Este método permite eliminar de manera lógica una bodega desde el perfil de administrador.

5.1.4.14 **Frontend:** Historia de usuario 11

La historia de usuario 11 corresponde a creación de bodegas, esta acción es realizada por parte del administrador.

Figura 44

Historia de usuario 11, crear una bodega

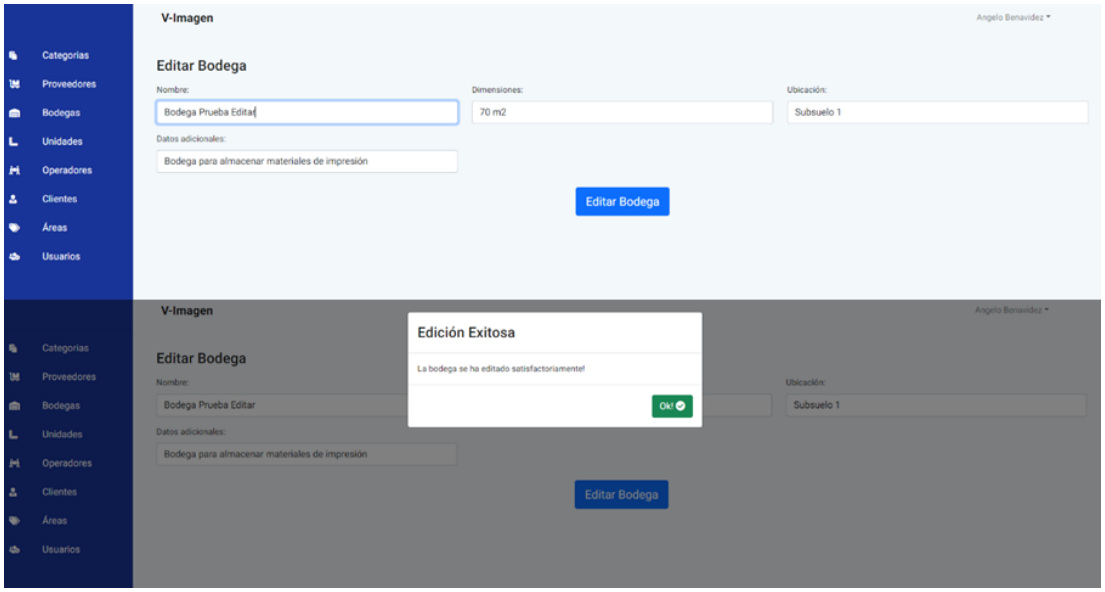


5.1.4.15 **Frontend:** Historia de usuario 12

La historia de usuario 12 corresponde a actualización de bodegas, esta acción es realizada por parte del administrador.

Figura 45

Historia de usuario 12, actualizar una bodega



5.1.4.16 **Frontend:** Historia de usuario 13

La historia de usuario 13 corresponde a consulta general y por parámetros de bodegas, esta acción es realizada por parte del administrador.

Figura 46

Historia de usuario 13, consultar bodegas de forma general

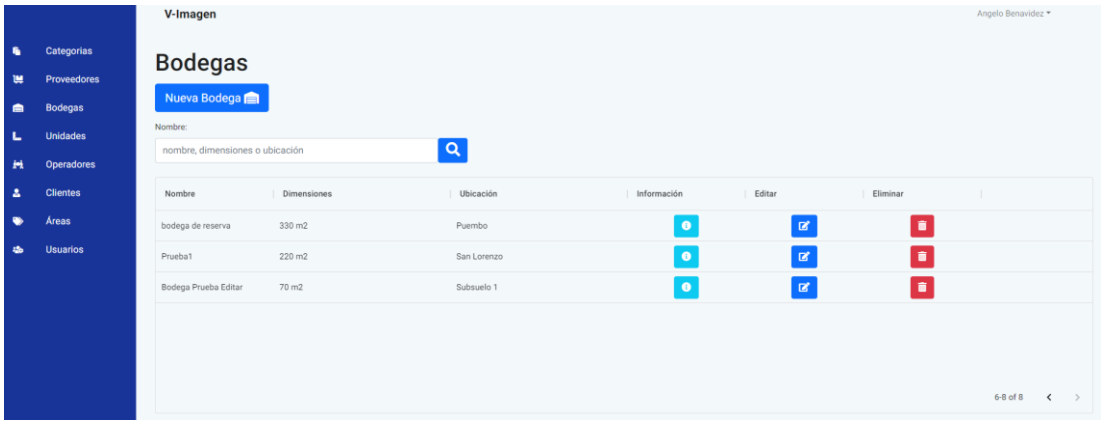


Figura 47

Historia de usuario 13, consultar detalles de una bodega

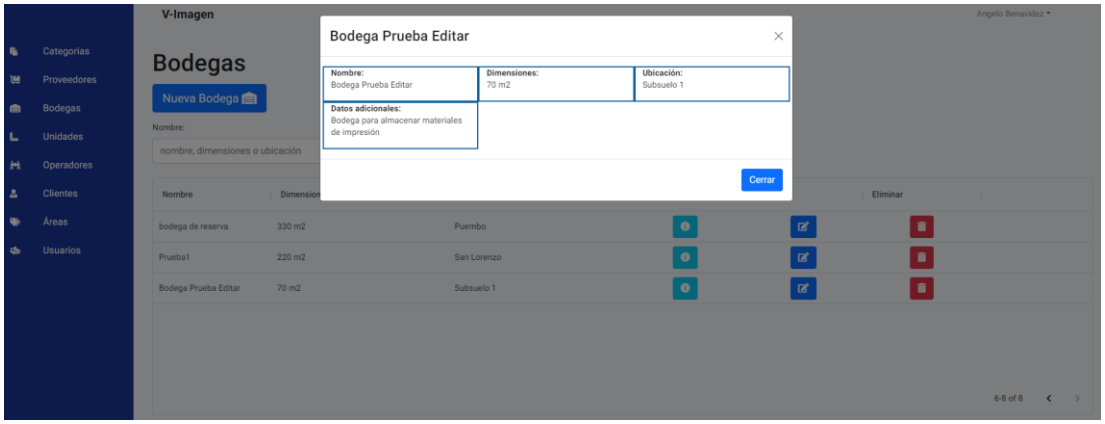
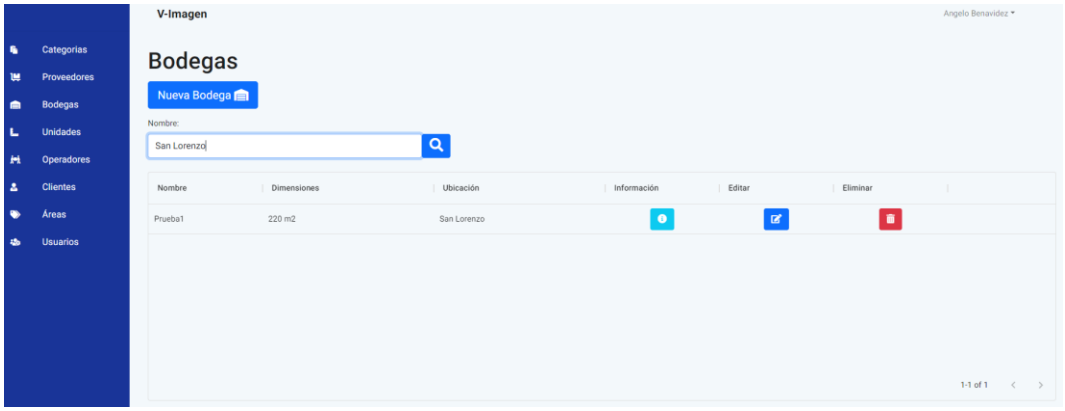


Figura 48

Historia de usuario 13, consulta por parámetros de una bodega

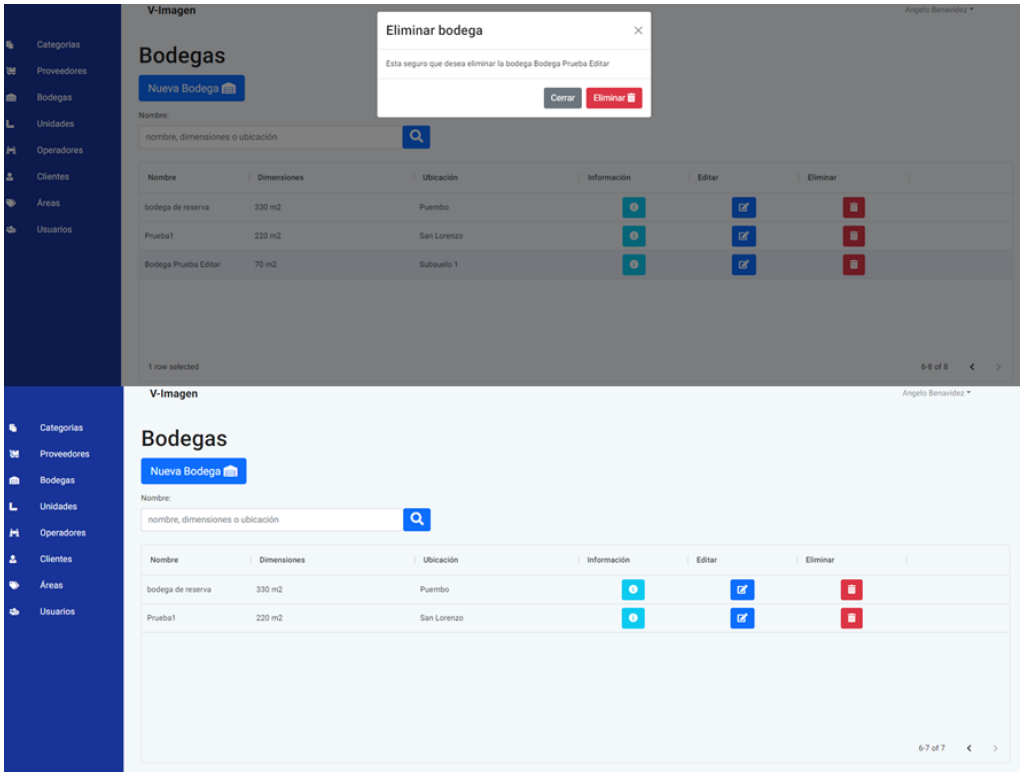


5.1.4.17 **Frontend:** Historia de usuario 14

La historia de usuario 14 corresponde a eliminar una bodega, esta acción es realizada por parte del administrador.

Figura 49

Historia de usuario 14, eliminar una bodega



5.1.4.18 **Backend:** Resumen épica de Proveedores

En la tabla 12 se muestran los recursos del backend para las historias de usuarios 15, 16, 17 y 18 correspondiente a la épica de Proveedores.

Tabla 12

Historia de usuario 15, 16, 17 y 18: Endpoints

Historia de usuario	Método	URI	Descripción
15	POST	/api/proveedores	Esté método nos permite registrar un nuevo proveedor, con sus datos asociados desde el perfil de administrador.
16	PUT	/api/proveedores	Este método permite actualizar los datos de un proveedor desde el perfil de administrador.
17.1	GET	/api/proveedores/ /api/proveedores/id	El primer método permite obtener una lista de todos los proveedores. El segundo método permite obtener un proveedor por su id.

17.2	GET	/api/buscar/proveedores/valor	Este método permite buscar un proveedor por su nombre, razón social o teléfono.
18	DELETE	/api/proveedores/id	Este método permite eliminar un proveedor de manera lógica de la aplicación.

5.1.4.19 **Frontend:** Historia de usuario 15

La historia de usuario 15 corresponde a creación de proveedores, esta acción es realizada por parte del administrador.

Figura 50

Historia de usuario 15, validaciones de crear proveedores

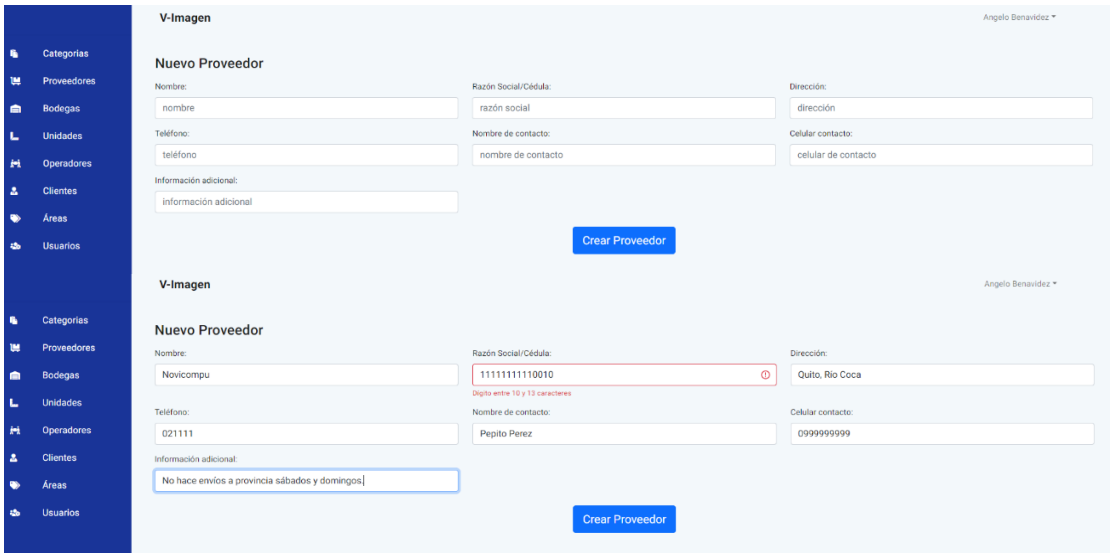
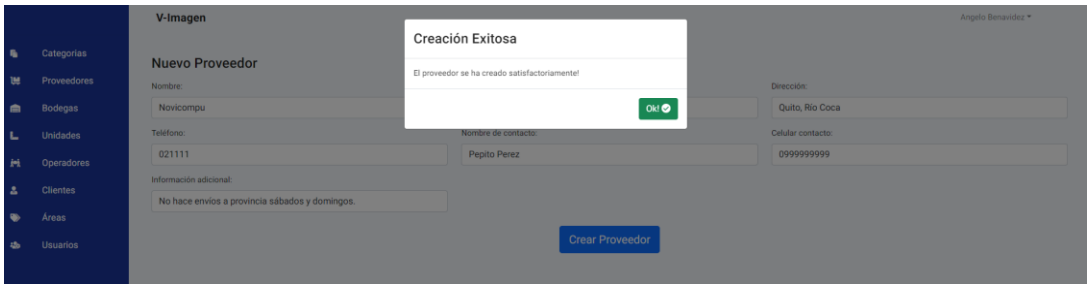


Figura 51

Historia de usuario 15, crear proveedor

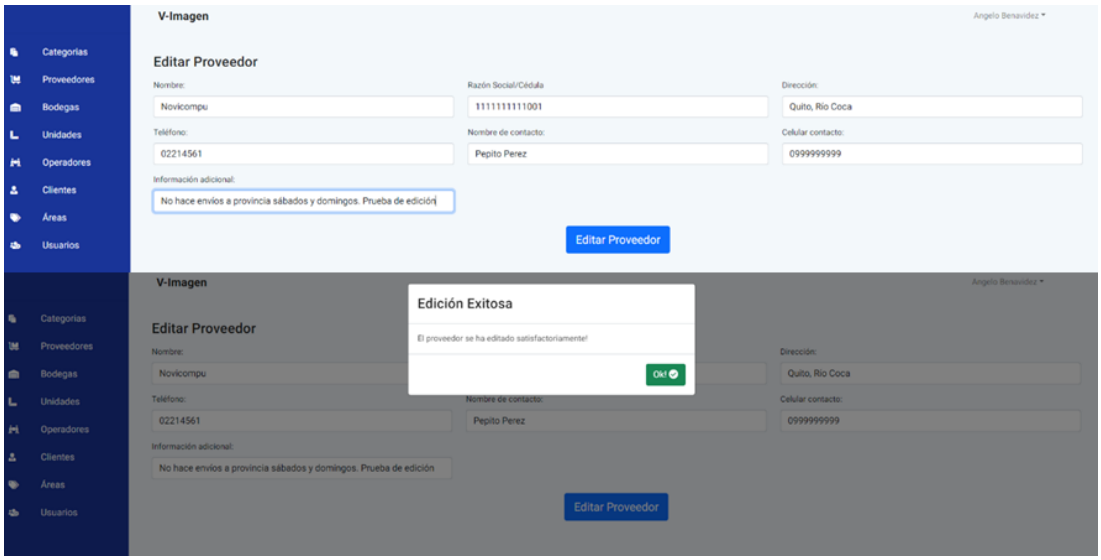


5.1.4.20 **Frontend:** Historia de usuario 16

La historia de usuario 16 corresponde a actualizar proveedores, esta acción es realizada por parte del administrador.

Figura 52

Historia de usuario 16, actualizar proveedor



5.1.4.21 **Frontend:** Historia de usuario 17

La historia de usuario 17 corresponde a consulta de proveedores a nivel general y de parámetros, esta acción es realizada por parte del administrador.

Figura 53

Historia de usuario 17, consulta general de proveedor

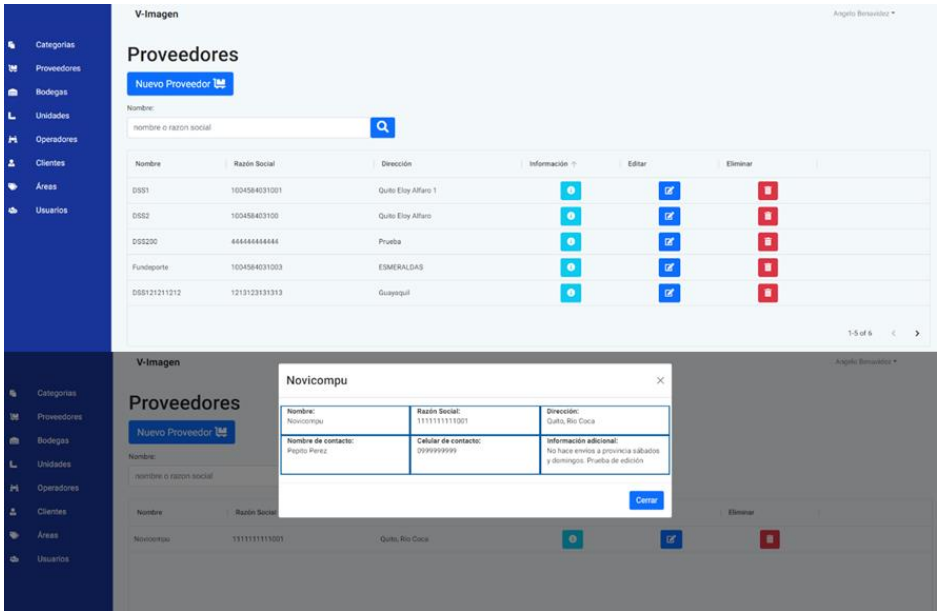
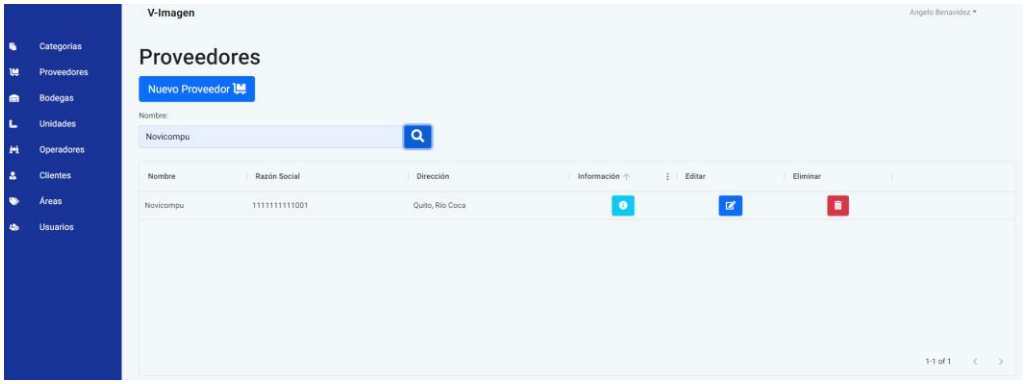


Figura 54

Historia de usuario 17, consulta por parámetros de proveedor

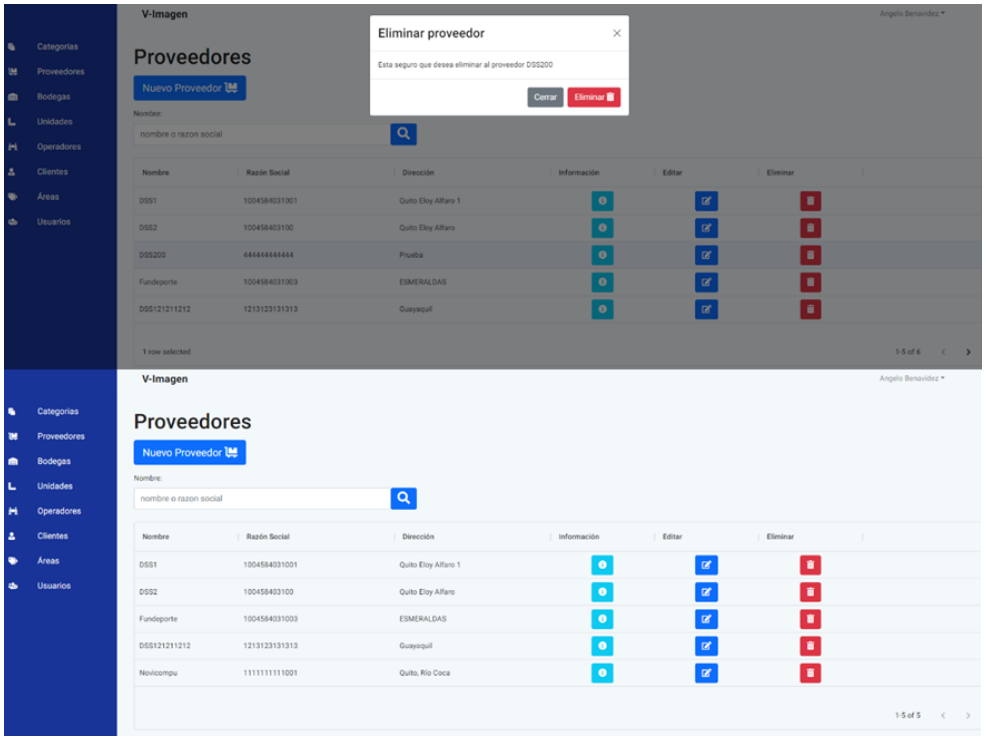


5.1.4.22 **Frontend:** Historia de usuario 18

La historia de usuario 18 corresponde a eliminación de proveedores, esta acción es realizada por parte del administrador.

Figura 55

Historia de usuario 18, eliminación de proveedor



5.1.4.23 **Backend**: Resumen de épica de Unidades

En la tabla 13 se muestran los recursos del backend para las historias de usuarios 19, 20, 21 y 22 correspondiente a la épica de Unidades.

Tabla 13

Historia de usuario 19, 20, 21 y 22: Endpoints

Historia de usuario	Método	URI	Descripción
19	POST	/api/unidades	Esté método nos permite registrar una nueva unidad, con sus datos asociados desde el perfil de administrador.
20	PUT	/api/unidades	Este método permite actualizar los datos de una unidad desde el perfil de administrador.
21.1	GET	/api/unidades/ /api/unidades/id	El primer método permite obtener una lista de todas las unidades registradas. El segundo método permite obtener una unidad por su id.

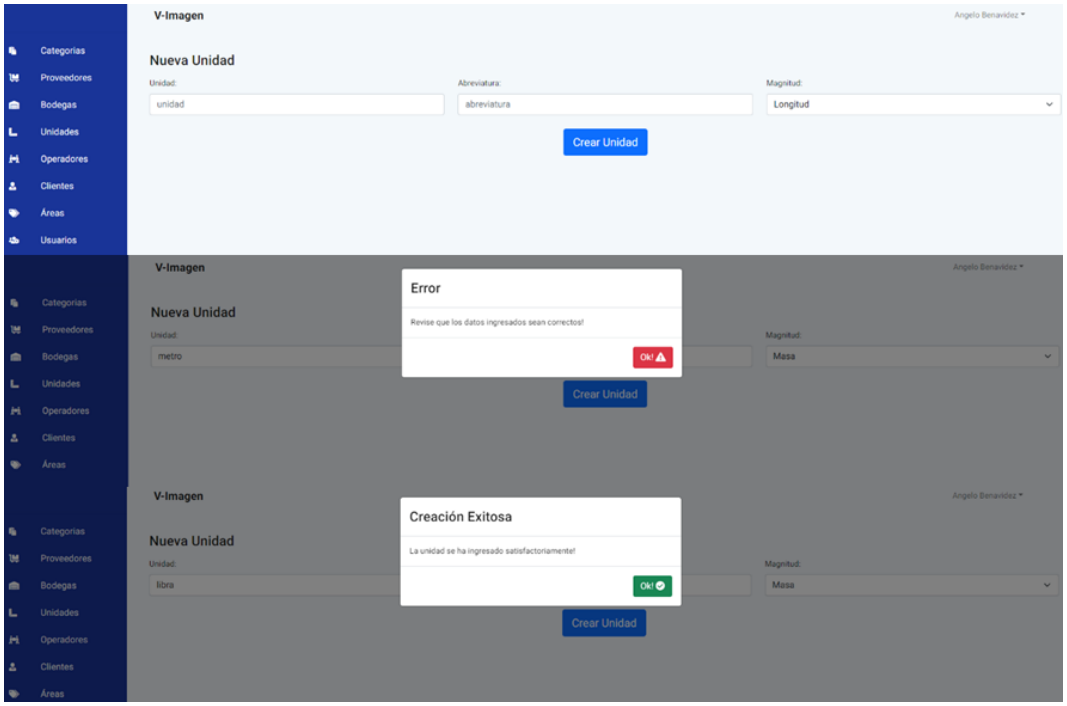
21.2	GET	/api/buscar/unidades/valor	Este método permite buscar una unidad por su unidad y abreviatura desde el perfil de administrador.
22	DELETE	/api/unidades/id	Este método permite eliminar una unidad de manera lógica de la aplicación desde el perfil de administrador.

5.1.4.24 **Frontend:** Historia de usuario 19

La historia de usuario 19 corresponde a creación de unidades, esta acción es realizada por parte del administrador.

Figura 56

Historia de usuario 19, creación de unidades

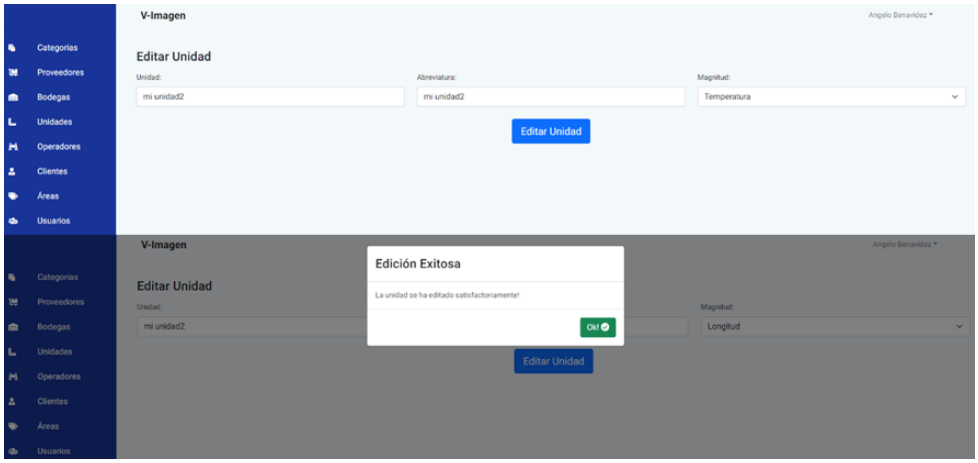


5.1.4.25 **Frontend:** Historia de usuario 20

La historia de usuario 20 corresponde a actualización de unidades, esta acción es realizada por parte del administrador.

Figura 57

Historia de usuario 20, actualización de unidades



5.1.4.26 **Frontend:** Historia de usuario 21

La historia de usuario 21 corresponde a consulta de unidades a nivel general y por parámetros de unidades, esta acción es realizada por parte del administrador.

Figura 58

Historia de usuario 21, consulta a nivel general de unidades

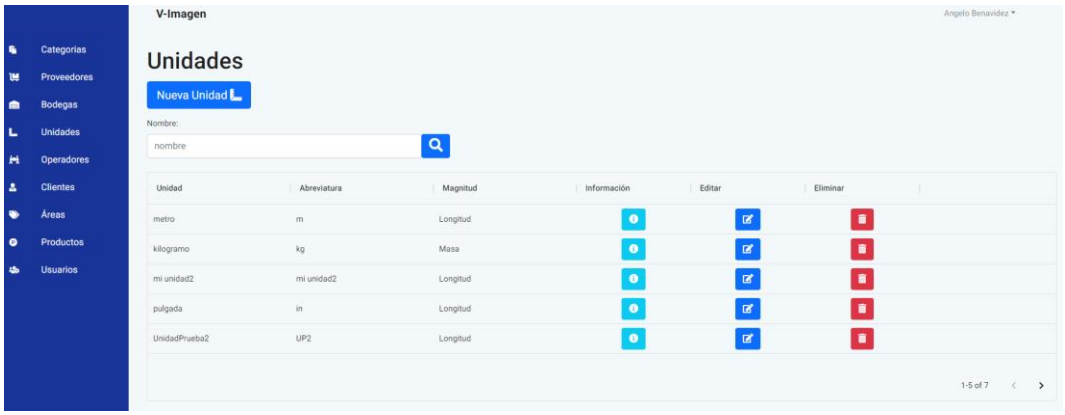
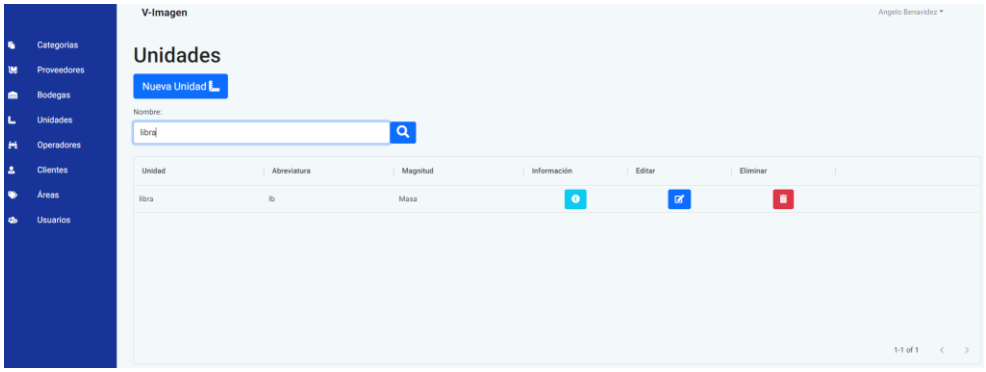


Figura 59

Historia de usuario 21, consulta por parámetros de unidades

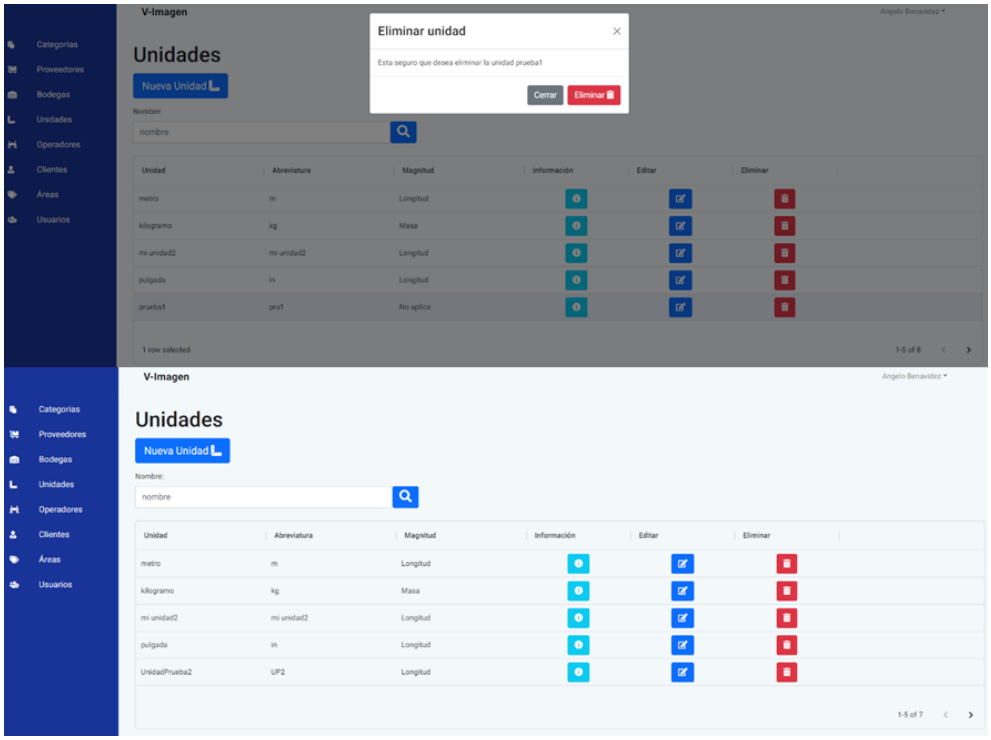


5.1.4.27 **Frontend:** Historia de usuario 22

La historia de usuario 22 corresponde eliminación de unidades, esta acción es realizada por parte del administrador.

Figura 60

Historia de usuario 22, eliminación de unidades



5.1.4.28 **Backend:** Resumen de épicas de Áreas

En la tabla 14 se muestran los recursos del backend para las historias de usuarios 23, 24, 25 y 26 correspondiente a la épica de Áreas.

Tabla 14

Historia de usuario 23, 24, 25 y 26: Endpoints

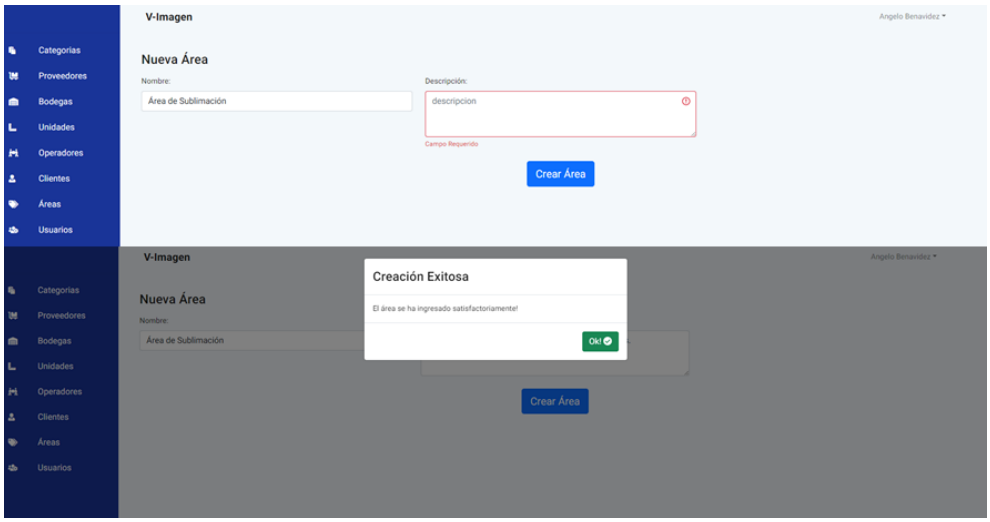
Historia de usuario	Método	URI	Descripción
23	POST	/api/areas	Esté método nos permite registrar una nueva área, con sus datos asociados desde el perfil de administrador.
24	PUT	/api/areas	Este método permite actualizar los datos un área desde el perfil de administrador.
25.1	GET	/api/areas /api/areas/id	El primer método permite obtener una lista de todas las áreas. El segundo método permite obtener un área por su id.
25.2	GET	/api/buscar/areas/valor	Este método permite buscar un área por su nombre desde el perfil de administrador.
26	DELETE	/api/areas/id	Este método permite eliminar un área de manera lógica de la aplicación desde el perfil de administrador.

5.1.4.29 **Frontend:** Historia de usuario 23

La historia de usuario 23 corresponde a la creación de áreas, esta acción es realizada por parte del administrador.

Figura 61

Historia de usuario 23, creación de áreas

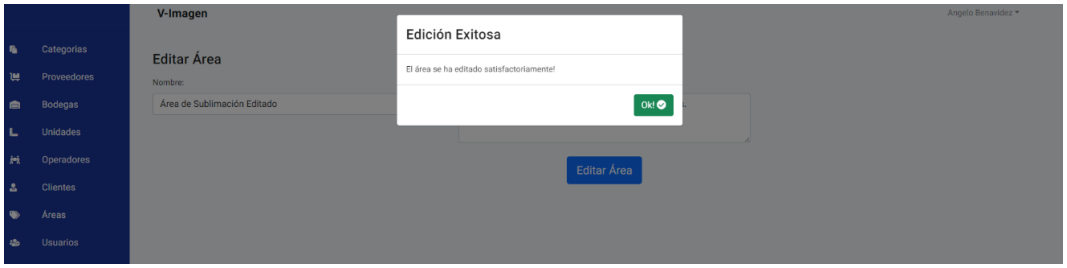


5.1.4.30 **Frontend:** Historia de usuario 24

La historia de usuario 24 corresponde a la actualización de áreas, esta acción es realizada por parte del administrador.

Figura 62

Historia de usuario 24, actualización de áreas



5.1.4.31 **Frontend:** Historia de usuario 25

La historia de usuario 25 corresponde a la consulta de áreas a nivel general y por parámetros de áreas, esta acción es realizada por parte del administrador.

Figura 63

Historia de usuario 25, consulta a nivel general de áreas

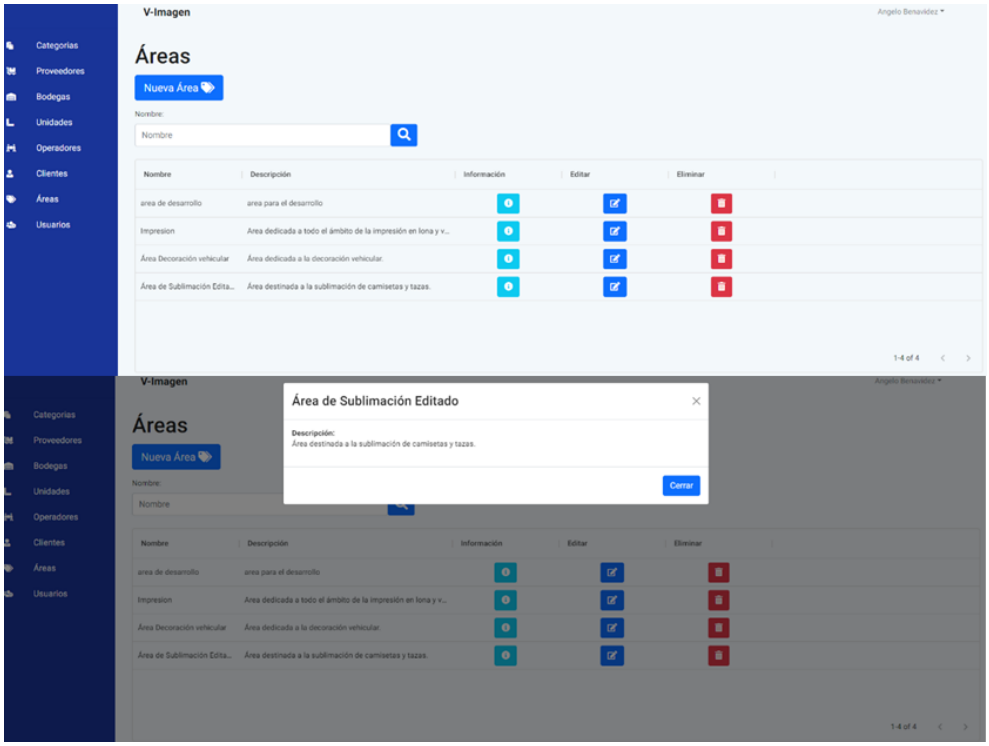
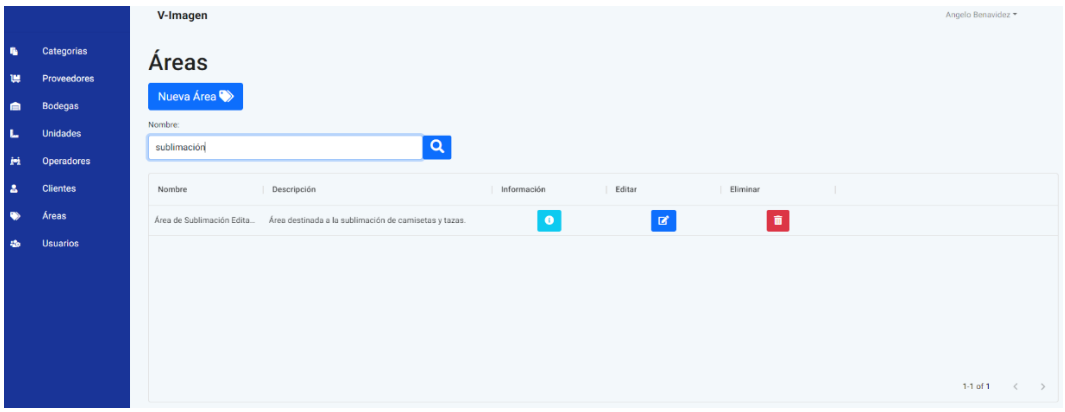


Figura 64

Historia de usuario 25, consulta por parámetro de áreas

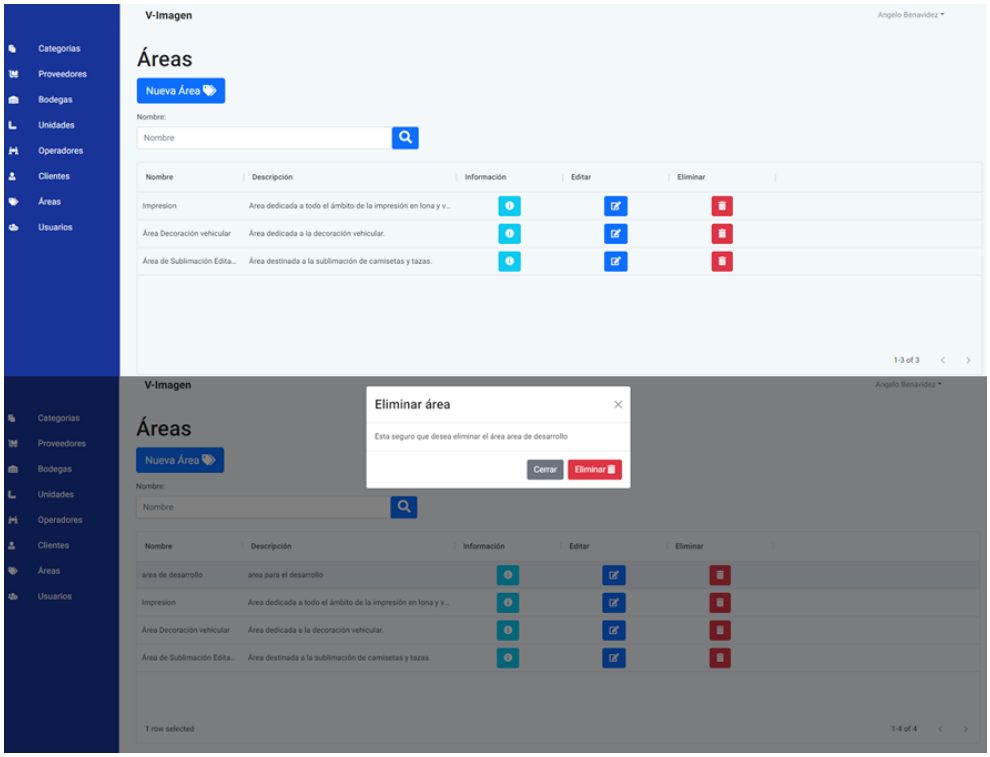


5.1.4.32 **Frontend:** Historia de usuario 26

La historia de usuario 26 corresponde a la eliminación de áreas, esta acción es realizada por parte del administrador.

Figura 65

Historia de usuario 26, eliminación de áreas



5.1.4.33 **Backend:** Resumen de épicas de Clientes

En la tabla 15 se muestran los recursos del backend para las historias de usuarios 27, 28, 29 y 30 correspondiente a la épica de Clientes.

Tabla 15

Historia de usuario 27, 28, 29 y 30: Endpoints

Historia de usuario	Método	URI	Descripción
27	POST	/api/clientes	Esté método nos permite registrar un nuevo cliente, con sus datos asociados desde el perfil de administrador y operador.
28	PUT	/api/clientes	Este método permite actualizar los datos un cliente desde el perfil de administrador y operador.
29.1	GET	/api/clientes /api/clientes/id	El primer método permite obtener una lista de todos los clientes registrados.

			El segundo método permite obtener un cliente por su id.
29.2	GET	/api/buscar/clientes/valor	Este método permite buscar un cliente por su cedula, nombres, apellidos y correo desde el perfil de administrador y operador.
30	DELETE	/api/clientes/id	Este método permite eliminar un cliente de manera lógica de la aplicación desde el perfil de administrador.

5.1.4.34 **Frontend:** Historia de usuario 27

La historia de usuario 27 corresponde a la creación de clientes, esta acción es realizada por parte del administrador y operador.

Figura 66

Historia de usuario 27, validaciones para creación de clientes.

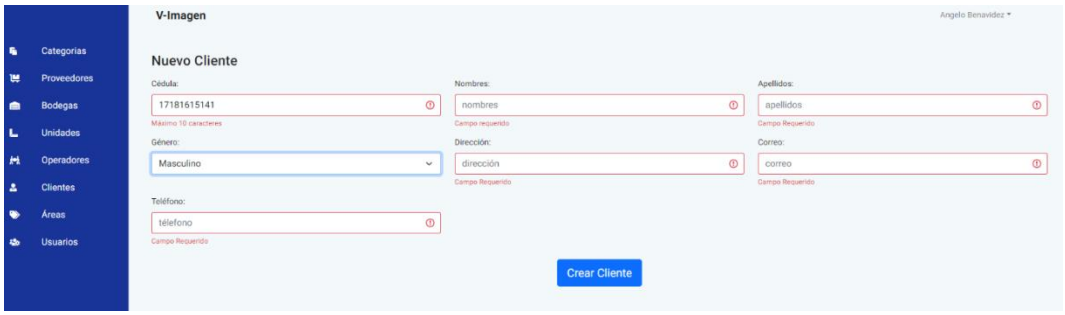
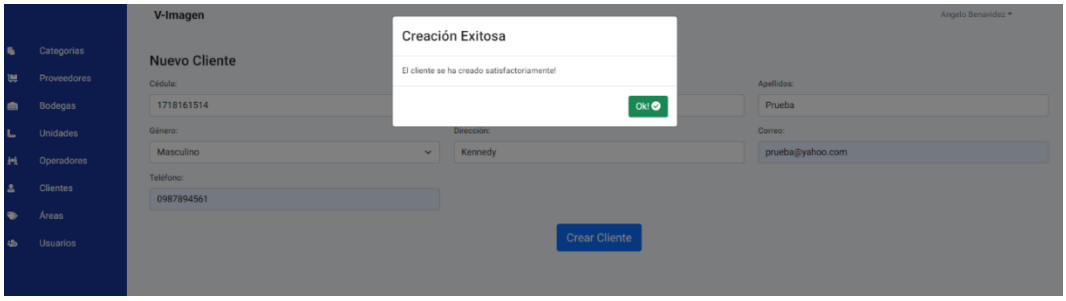


Figura 67

Historia de usuario 27, validaciones para creación de clientes.

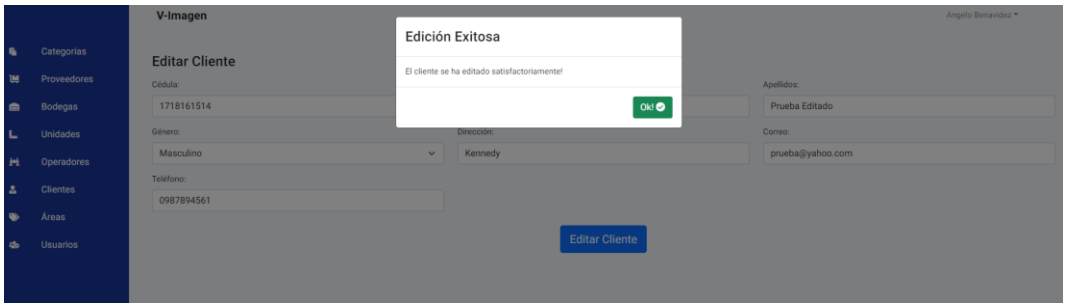


5.1.4.35 **Frontend:** Historia de usuario 28

La historia de usuario 28 corresponde a la actualización de clientes, esta acción es realizada por parte del administrador y operador.

Figura 68

Historia de usuario 28, actualización de cliente.



5.1.4.36 **Frontend:** Historia de usuario 29

La historia de usuario 29 corresponde a la consulta de clientes a nivel general y por parámetros, esta acción es realizada por parte del administrador y operador.

Figura 69

Historia de usuario 29, consulta general de clientes.

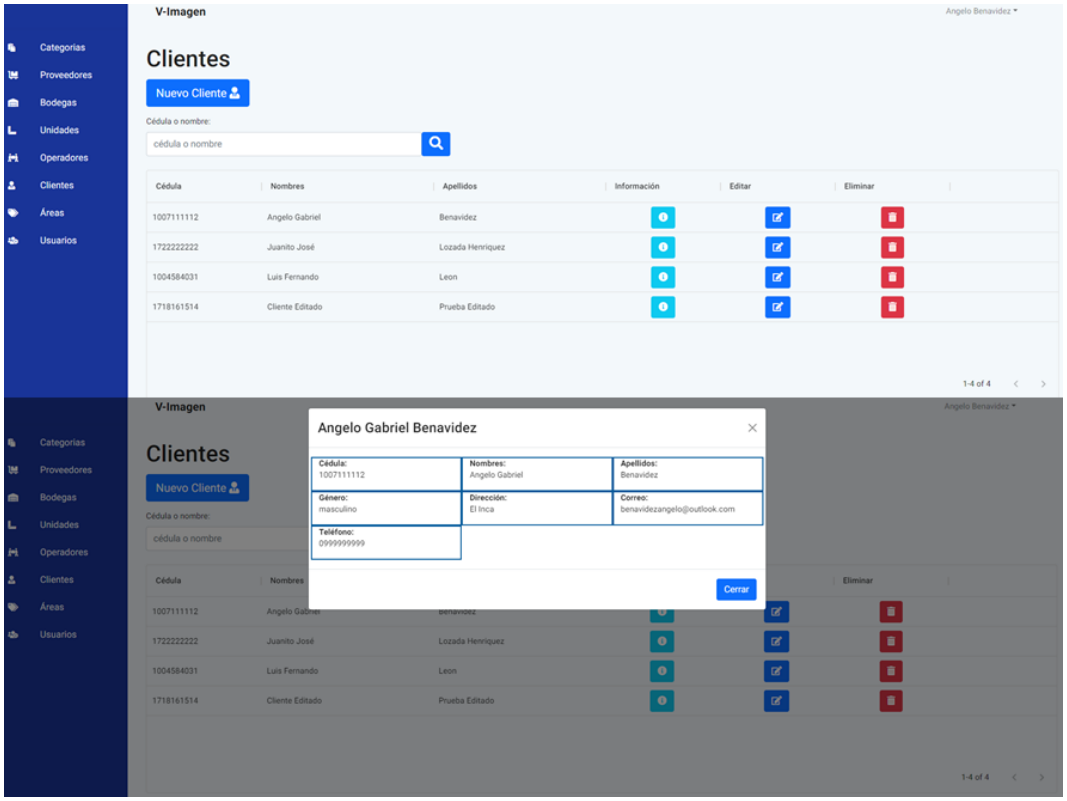
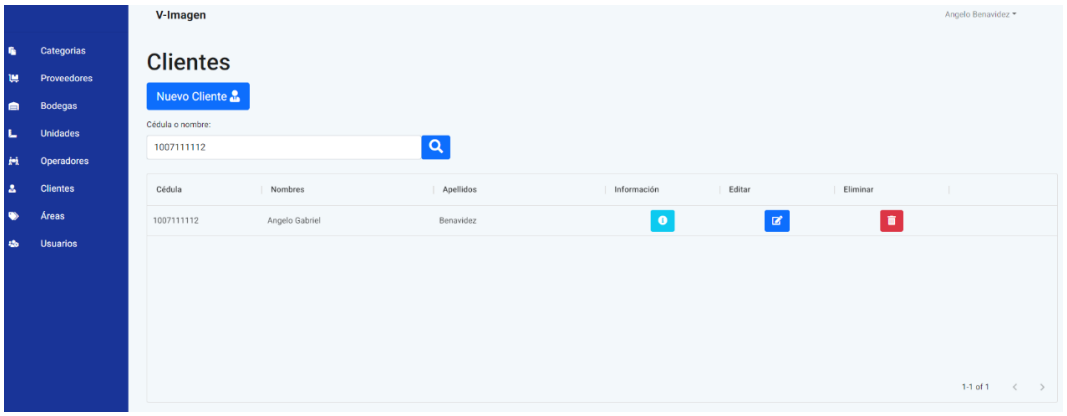


Figura 70

Historia de usuario 29, consulta por parámetro de clientes.

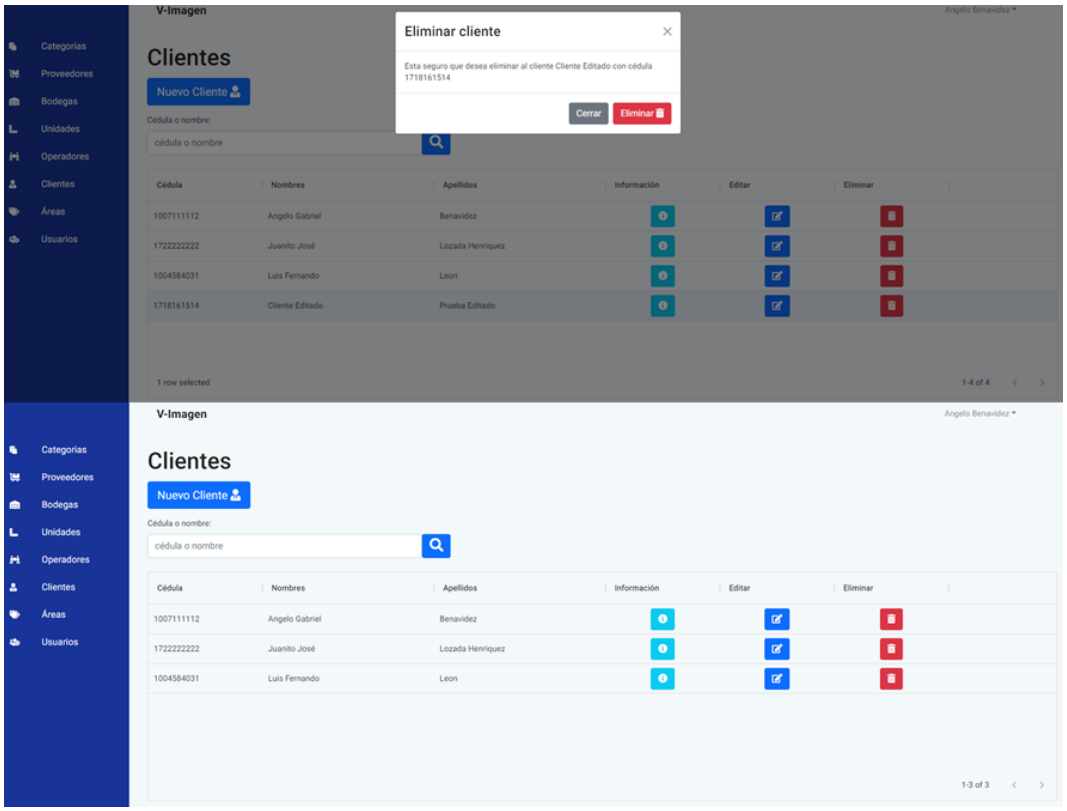


5.1.4.37 **Frontend:** Historia de usuario 30

La historia de usuario 30 corresponde a la eliminación de clientes, esta acción es realizada por parte del administrador.

Figura 71

Historia de usuario 30, eliminación de clientes



5.2 Sprint 2

5.2.1 Planificación del Sprint

En base a la planificación realizada para el sprint 2 se tiene las siguientes historias de usuario descritas en la tabla 3. En la tabla 16 se puede observar un resumen de las historias de usuario que serán construidas en el Sprint 2.

Tabla 16

Resumen de historias de usuario de sprint 2

Número	Nombre de Historia	Rol
1	Registro de Productos	Administrador
2	Actualizar Productos	Administrador

Número	Nombre de Historia	Rol
3	Consulta General y por Parámetros de Productos	Administrador/Operador
4	Eliminar Productos	Administrador
5	Registro de Categorías	Administrador
6	Actualizar Categorías	Administrador
7	Consulta General y por Parámetros de Categorías	Administrador
8	Eliminar Categorías	Administrador
9	Registrar Ordenes de Trabajo	Administrador
10	Actualizar Datos de Ordenes de Trabajo	Administrador/Operador
11	Visualización en Calendario de Ordenes de Trabajo/Consulta General	Administrador/Operador
12	Eliminar Ordenes de Trabajo	Administrador

5.2.2 Sprint backlog

El sprint backlog correspondiente al sprint 2 contempla las tablas bases restantes, adicional a la épica de Productos que se convierte en una entidad que ya consume recursos de otras entidades. Por otra parte, en el sprint 2 se trabajará el proceso de ordenes de trabajo. Se puede ver los requerimientos a trabajar en la tabla 16.

5.2.3 Duración del Sprint

El sprint 2 en base a la planificación y estimación del proceso de desarrollo tendrá una duración de 36 horas de trabajo.

5.2.4 Desarrollo

Para el desarrollo se empezará construyendo los Endpoints en el Backend y luego las pantallas asociadas a la historia de usuario.

5.2.4.1 Backend: Resumen de épica de Productos

En la tabla 17 se muestran los recursos del backend para las historias de usuarios 31, 32, 33 y 34 correspondiente a la épica de Usuarios

Tabla 17

Historia de usuario 31, 32, 33 y 34: Endpoints

Historia de usuario	Método	URI	Descripción
31	POST	/api/productos	Esté método nos permite registrar un nuevo producto, con sus datos asociados desde el perfil de administrador.
32	PUT	/api/productos	Este método permite actualizar los datos un producto desde el perfil de administrador.
33.1	GET	/api/productos /api/productos/id	El primer método permite obtener una lista general de todos los productos registrados. El segundo método permite obtener un producto por su id.
33.2	GET	/api/buscar/productos/valor	Este método permite buscar un producto por su SKU, nombre, nombre del proveedor y nombre de la categoría desde el perfil de administrador y operador.
34	DELETE	/api/productos/id	Este método permite eliminar un producto de manera lógica de la aplicación desde el perfil de administrador.

5.2.4.2 Frontend: Historia de usuario 31

La historia de usuario 31 corresponde a la creación de productos, esta acción es realizada por parte del administrador.

Figura 72

Historia de usuario 31, validaciones para creación de producto

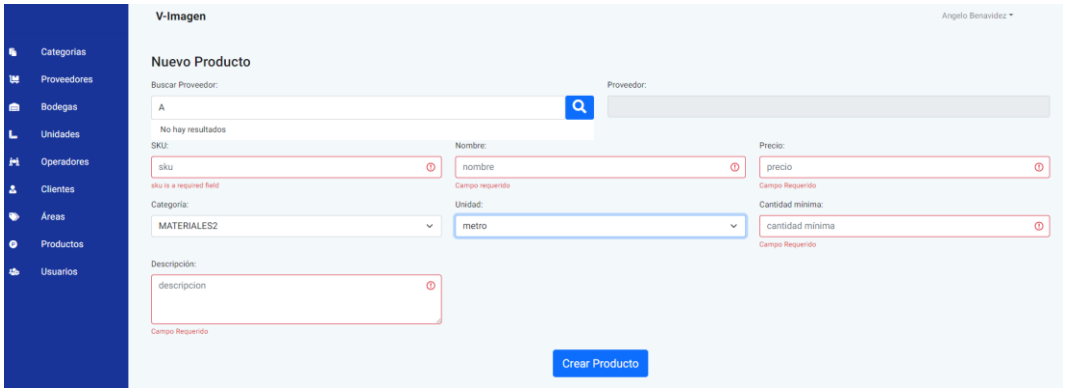
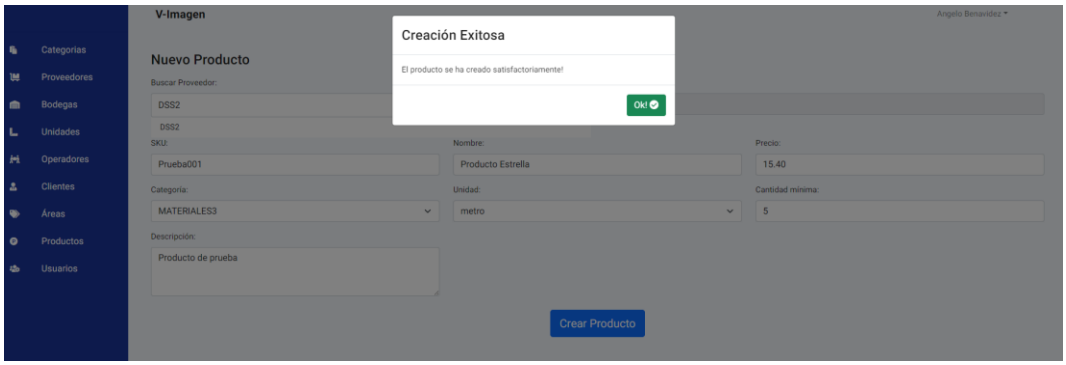


Figura 73

Historia de usuario 31, creación de producto

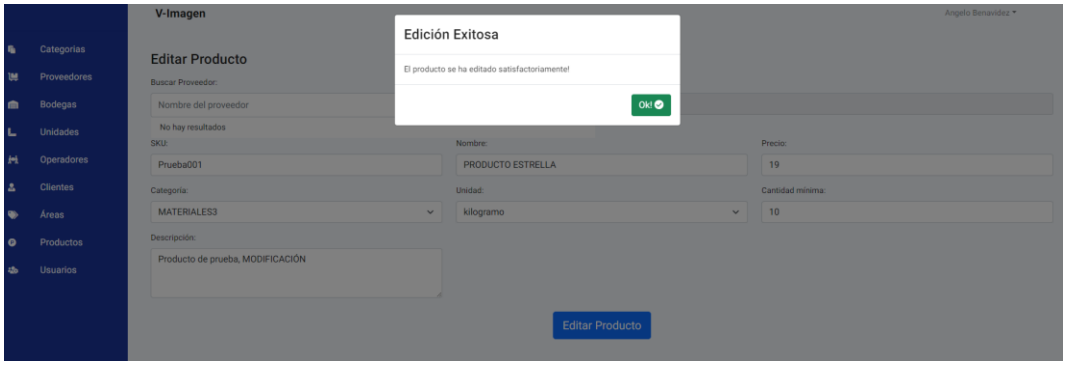


5.2.4.3 Frontend: Historia de usuario 32

La historia de usuario 32 corresponde a la actualización de productos, esta acción es realizada por parte del administrador.

Figura 74

Historia de usuario 32, actualización de producto



5.2.4.4 Frontend: Historia de usuario 33

La historia de usuario 33 corresponde a la consulta de productos a nivel general y por parámetros, esta acción es realizada por parte del administrador y operador.

Figura 75

Historia de usuario 33, consulta general de productos

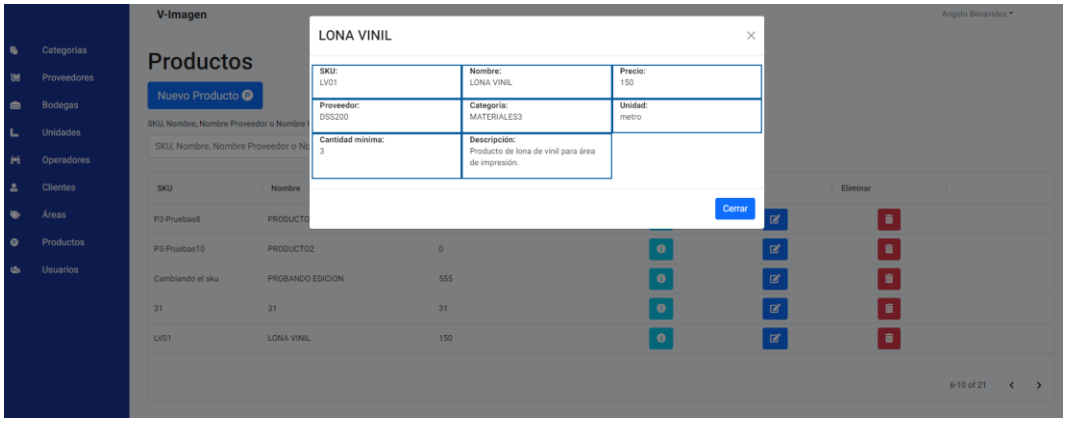
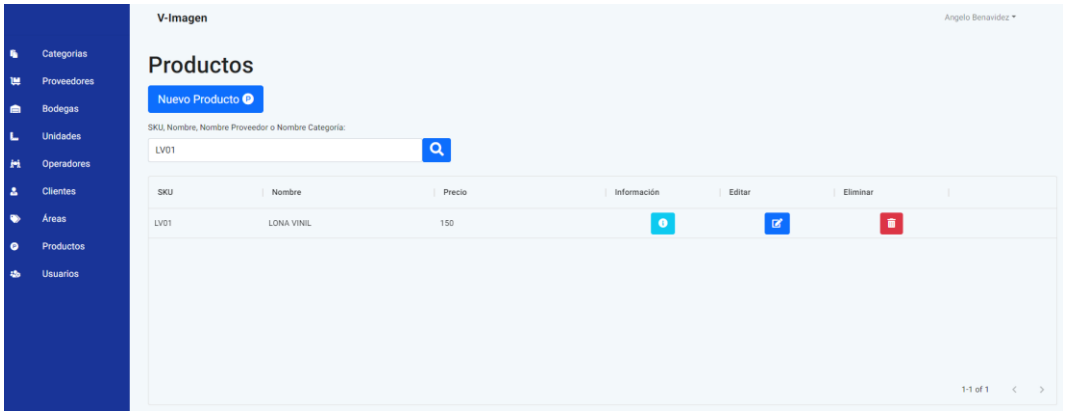


Figura 76

Historia de usuario 33, consulta por parámetro de productos

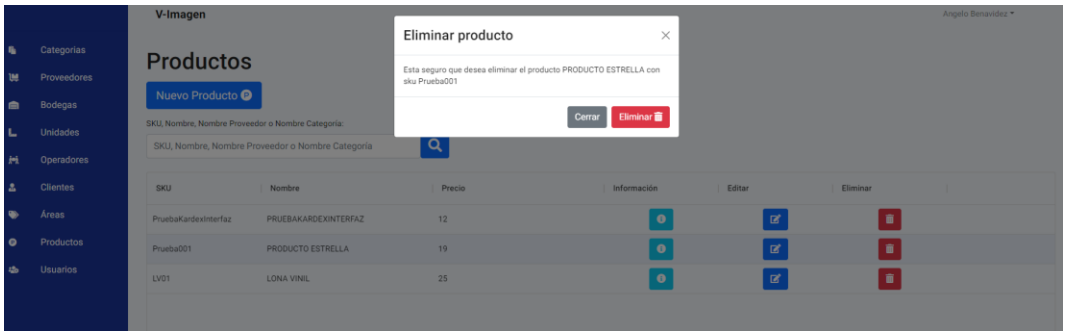


5.2.4.5 Frontend: Historia de usuario 34

La historia de usuario 34 corresponde a la eliminación de un producto, esta acción es realizada por parte del administrador.

Figura 77

Historia de usuario 34, eliminación de producto



5.2.4.6 Backend: Resumen de épica de Categoría

En la tabla 18 se muestran los recursos del backend para las historias de usuarios 35, 36, 37 y 38 correspondiente a la épica de Usuarios.

Tabla 18

Historia de usuario 35, 36, 37 y 38: Endpoints

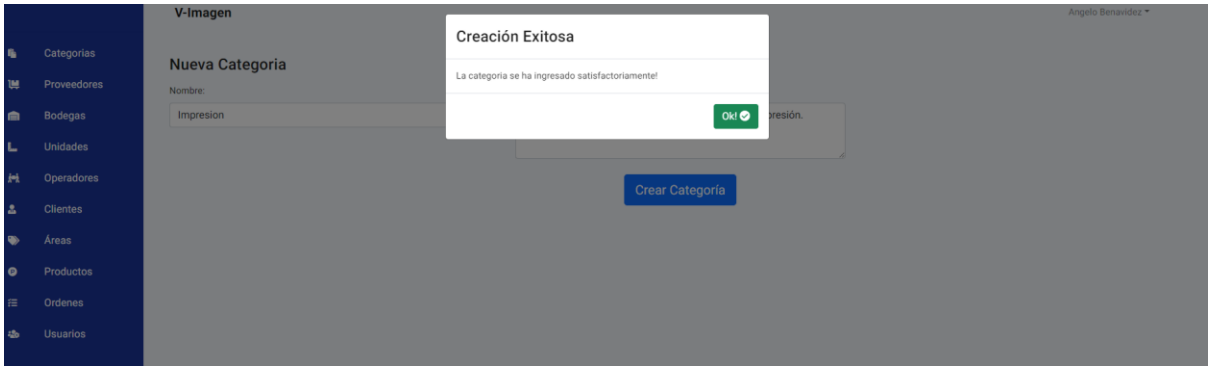
Historia de usuario	Método	URI	Descripción
35	POST	/api/categorias	Esté método nos permite registrar una nueva categoría, con sus datos asociados desde el perfil de administrador.
36	PUT	/api/categorias	Este método permite actualizar los datos una categoría desde el perfil de administrador.
37.1	GET	/api/categorias /api/categorias/id	El primer método permite obtener una lista de todas las categorías registradas. El segundo método permite obtener una categoría por su id.
37.2	GET	/api/buscar/categorias/valor	Este método permite buscar una categoría por su nombre desde el perfil de administrador.
38	DELETE	/api/categorias/id	Este método permite eliminar una categoría de manera lógica de la aplicación desde el perfil de administrador.

5.2.4.7 Frontend: Historia de usuario 35

La historia de usuario 35 corresponde a la creación de categorías, esta acción es realizada por parte del administrador.

Figura 78

Historia de usuario 35, creación de categoría

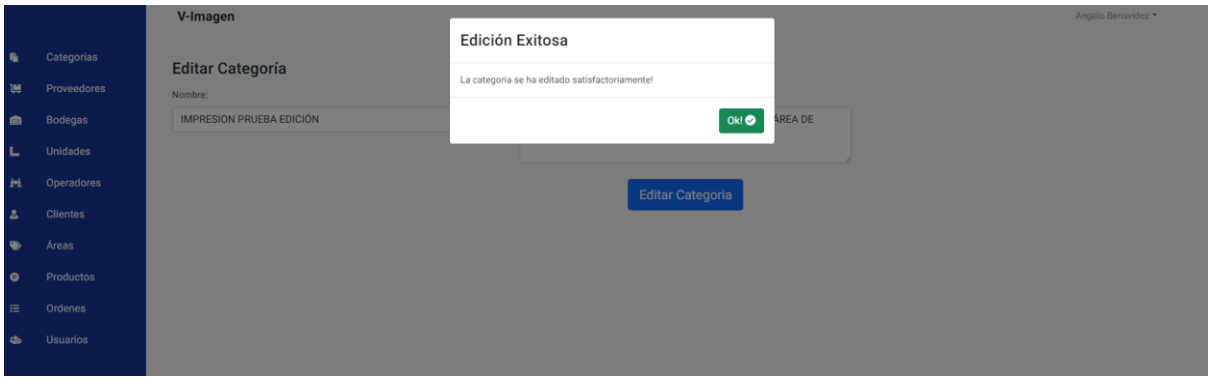


5.2.4.8 Frontend: Historia de usuario 36

La historia de usuario 36 corresponde a la actualización de categorías, esta acción es realizada por parte del administrador.

Figura 79

Historia de usuario 36, actualización de categoría



5.2.4.9 Frontend: Historia de usuario 37

La historia de usuario 37 corresponde a la consulta general y por parámetros de categorías, esta acción es realizada por parte del administrador.

Figura 80

Historia de usuario 37, consulta general de categorías

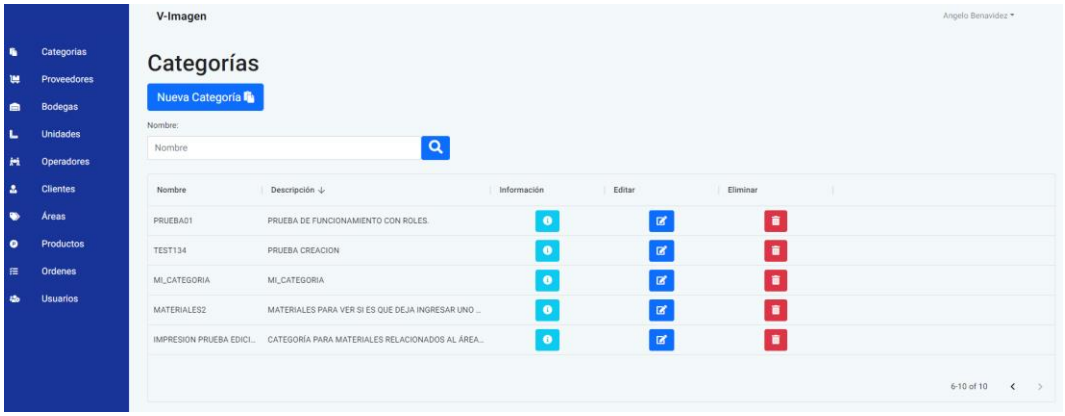
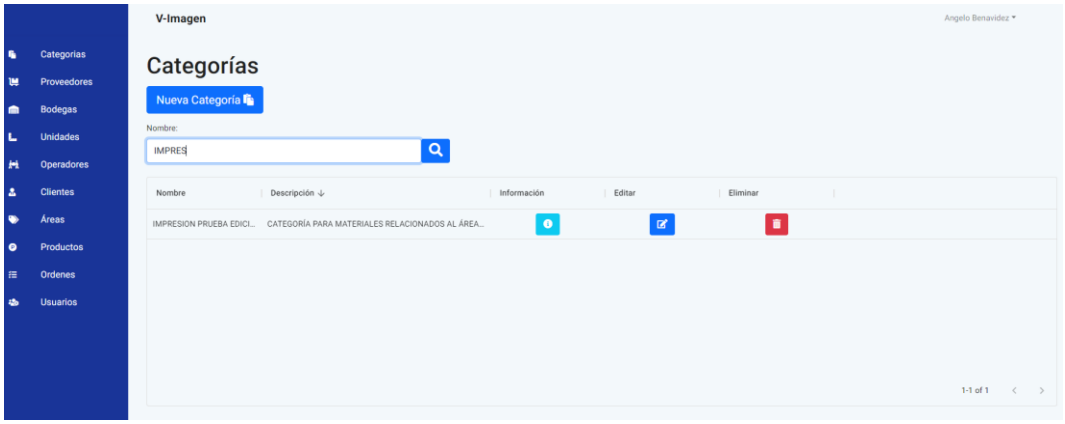


Figura 81

Historia de usuario 37, consulta por parámetros de categorías

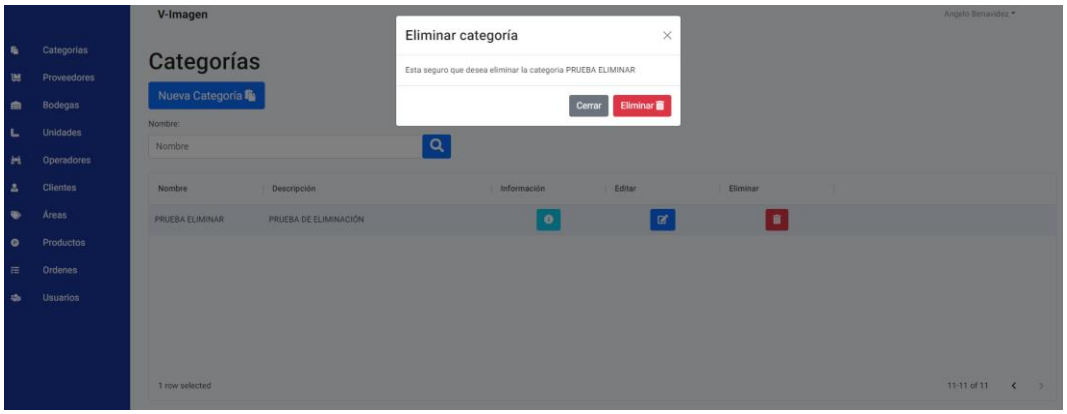


5.2.4.10 **Frontend:** Historia de usuario 38

La historia de usuario 38 corresponde a la eliminación de una categoría, esta acción es realizada por parte del administrador.

Figura 82

Historia de usuario 38, eliminación de categoría



5.2.4.11 **Backend:** Resumen de épica de Ordenes de trabajo

En la tabla 19 se muestran los recursos del backend para las historias de usuarios 39, 40, 41 y 42 correspondiente a la épica de Ordenes de Trabajo.

Tabla 19

Historia de usuario 39, 40, 41 y 42: Endpoints

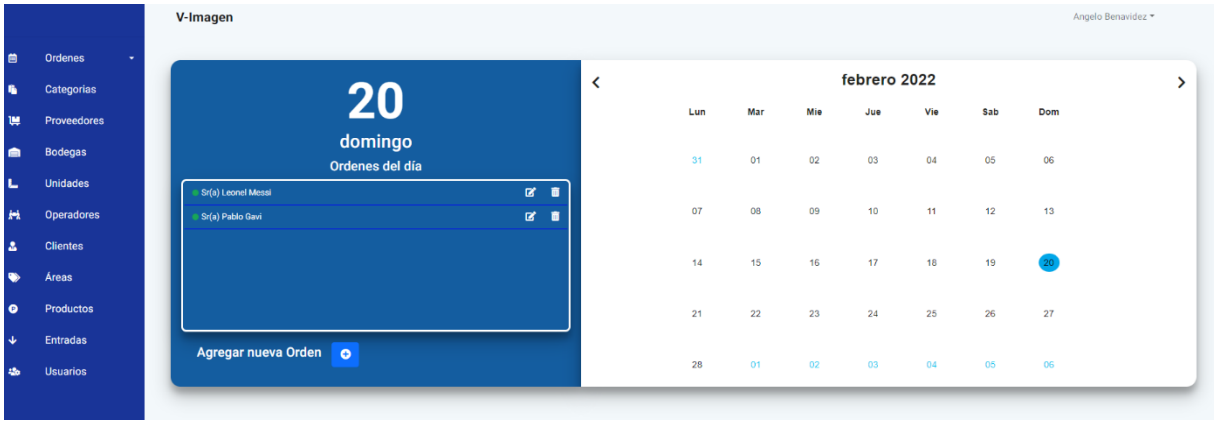
Historia de usuario	Método	URI	Descripción
39	POST	/api/ordenes	Esté método nos permite registrar una nueva orden, con sus datos asociados desde el perfil de administrador.
40	PUT	/api/ordenes	Este método permite actualizar los datos una orden desde el perfil de administrador y operador.
41.1	GET	/api/ordenes/id /api/ordenes/fecha	El primer método permite obtener una orden por su id. El segundo método permite obtener ordenes por fecha determinada.
41.2	GET	/api/buscar/ordenes/valor	Este método permite buscar una orden por el nombre del operador, nombre del cliente desde el perfil de administrador.
42	DELETE	/api/ordenes/id	Este método permite eliminar una orden de manera lógica de la aplicación desde el perfil de administrador.

5.2.4.12 **Frontend:** Historia de usuario 39

La historia de usuario 39 corresponde a la creación de una orden de trabajo, esta acción es realizada por parte del administrador.

Figura 83

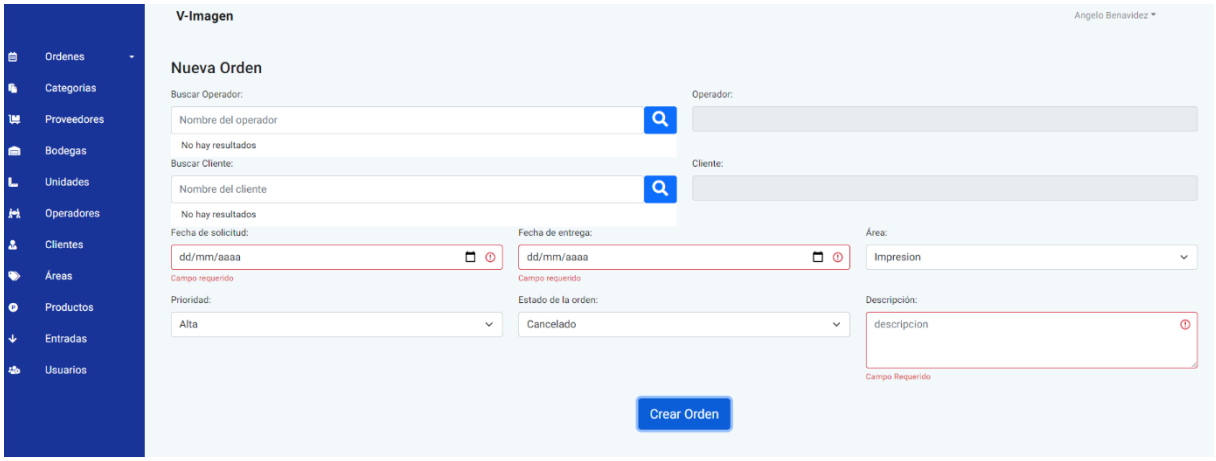
Historia de usuario 39, pantalla principal de creación de una orden



Nota. Pantalla principal para la gestión de las ordenes de trabajo, desde la interfaz se puede crear una nueva orden, editar la orden en función del día. En la sección de órdenes del día se tiene la ordenes con los operadores asignados y el estado de la orden.

Figura 84

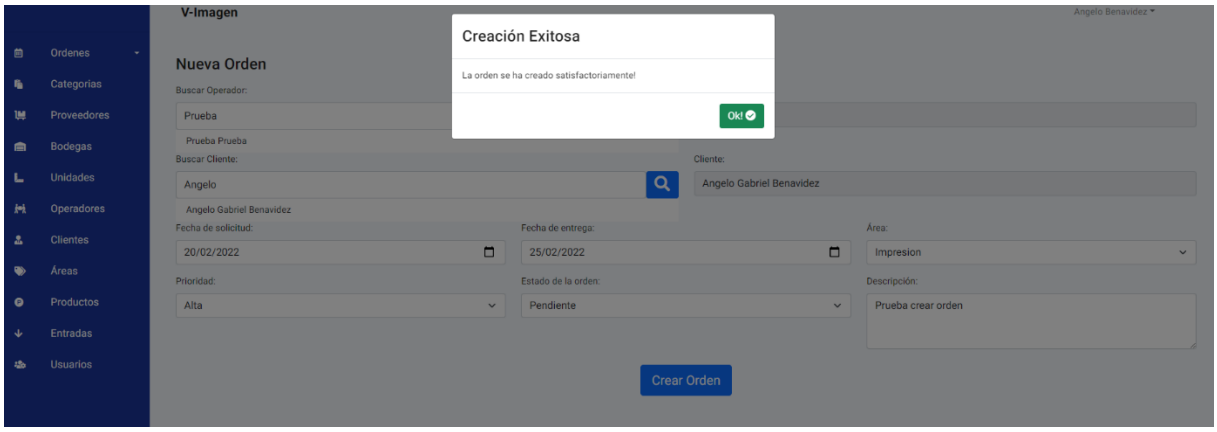
Historia de usuario 39, validaciones para la creación de una orden



Nota. El sistema no permite la creación de una orden de trabajo si los campos requeridos no están llenados correctamente.

Figura 85

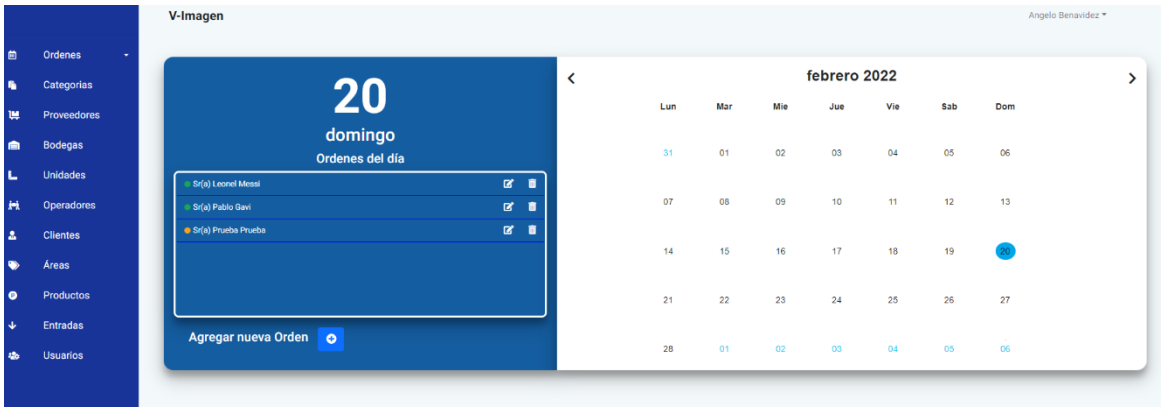
Historia de usuario 39, creación de una orden



Nota. Una vez la información registrada es correcta el sistema permite la creación de la orden de trabajo, esto lo podemos verificar en la pantalla principal de creación de orden de trabajo.

Figura 86

Historia de usuario 39, validación de creación de una de una orden



Nota. Se identifica la orden creada con un color en función del estado de la orden, por defecto se ubica el color naranja que muestra que la orden está pendiente, el color verde identifica una orden completada y el color rojo identifica una orden con estado cancelado.

5.2.4.13 **Frontend:** Historia de usuario 40

La historia de usuario 40 corresponde a la actualización de una orden de trabajo, esta acción es realizada por parte del administrador y operador.

Figura 87

Historia de usuario 40, actualización de una de una orden

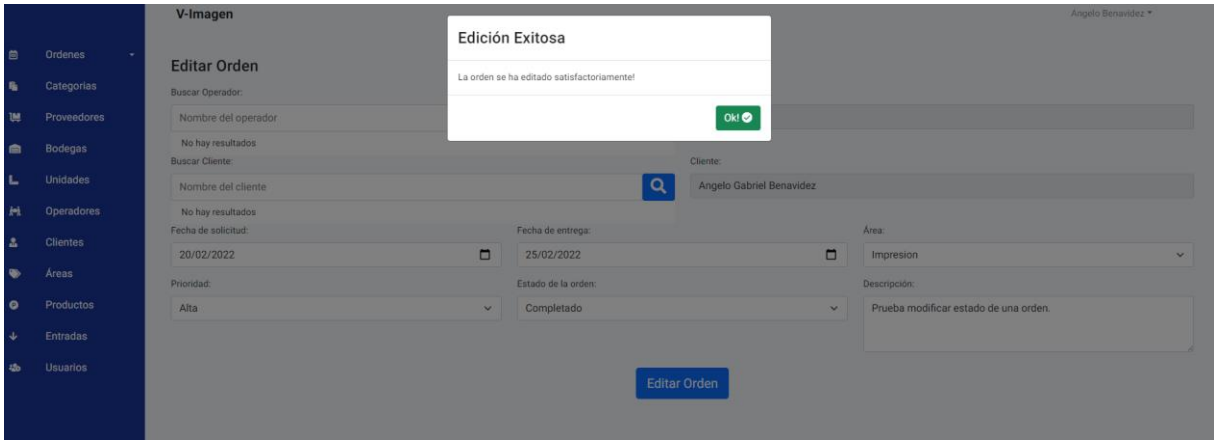
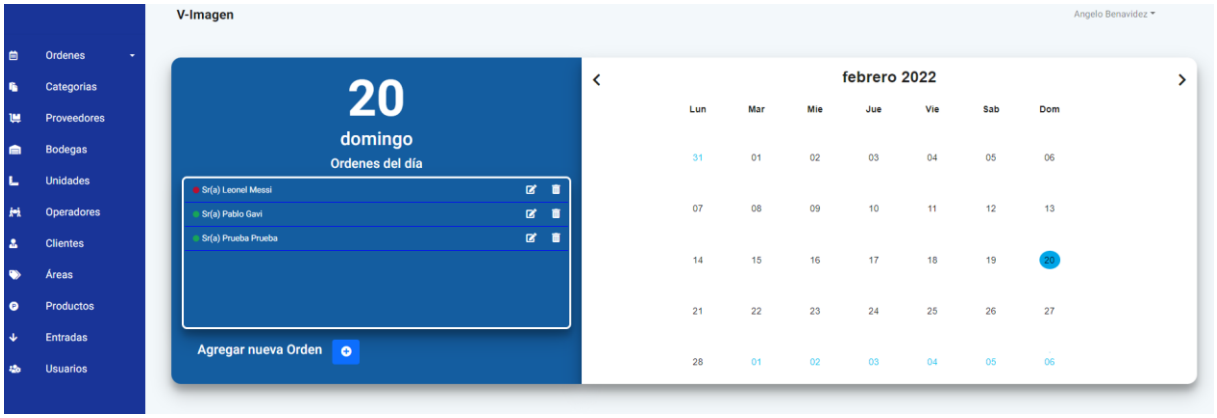


Figura 88

Historia de usuario 40, verificación de actualización de una de una orden



Nota. La bandera que identifica el estado de la orden ha cambiado a verde, denotando él estado actualizado.

5.2.4.14 **Frontend:** Historia de usuario 41

La historia de usuario 41 corresponde a la consulta general y por parámetros de ordenes de trabajo, esta acción es realizada por parte del administrador.

Figura 89

Historia de usuario 41, consulta general de ordenes

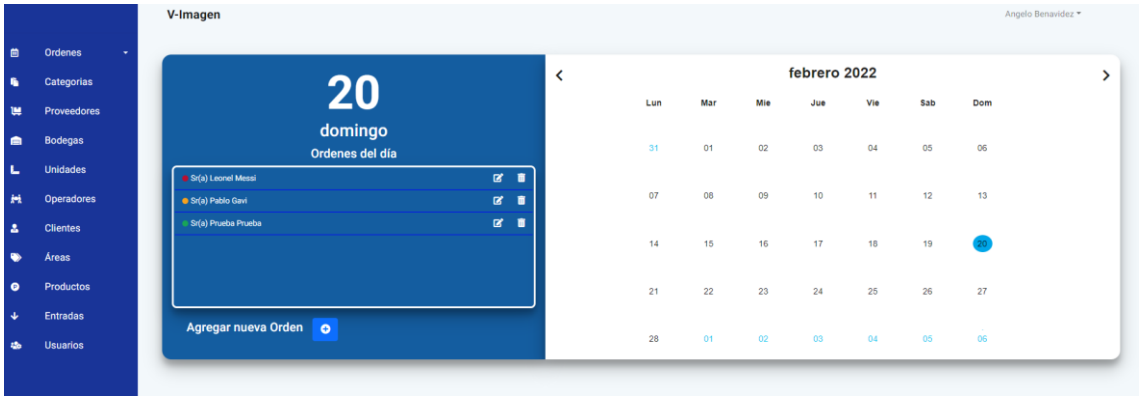
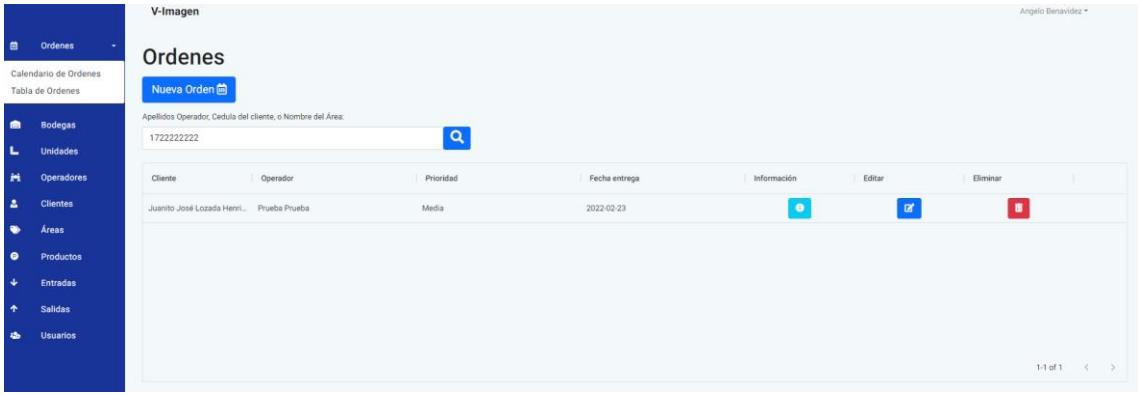


Figura 90

Historia de usuario 41, consulta por parámetros de ordenes



5.2.4.15 **Frontend:** Historia de usuario 42

La historia de usuario 42 corresponde a la eliminación de ordenes de trabajo, esta acción es realizada por parte del administrador.

Figura 91

Historia de usuario 42, modal para la eliminación de una orden

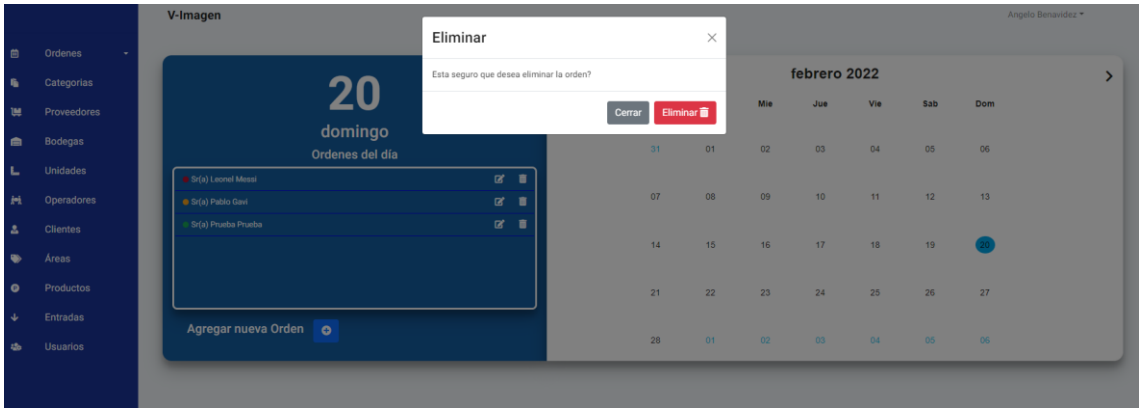
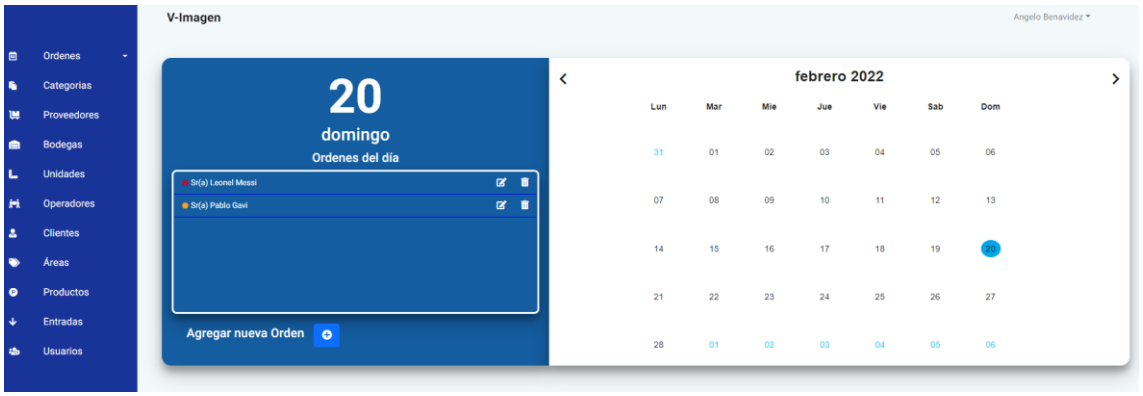


Figura 92

Historia de usuario 42, verificación de eliminación de la orden de trabajo



5.3 Sprint 3

5.3.1 Planificación del Sprint

En base a la planificación realizada para el sprint 3 se tiene las siguientes historias de usuario descritas en la tabla 3. En la tabla 20 se puede observar un resumen de las historias de usuario que serán construidas en el Sprint 3.

Tabla 20

Resumen de historias de usuario de sprint 3

Número	Nombre de Historia	Rol
1	Registro de Entradas	Administrador
2	Actualizar Entradas	Administrador
3	Consulta General y por Parámetros de Entradas	Administrador/Operador
4	Eliminar Entradas	Administrador
5	Registro de Salidas	Administrador
6	Actualizar Salidas	Administrador
7	Consulta General y por Parámetros de Salidas	Administrador
8	Eliminar Salidas	Administrador
9	Registro de Kardex	Administrador
10	Cambiar contraseña desde el perfil	Administrador/Operador
11	Registrar Kardex automáticamente	NA
12	Actualizar Kardex automáticamente	NA
13	Construir dashboard de estadísticas generales	Administrador

5.3.2 Sprint backlog

El sprint backlog correspondiente al sprint 3 contempla los procesos relacionados a el control de inventarios, la actualización de contraseñas desde el perfil del usuario, además del envío de correos previo falta de existencias mínimas de los productos, se puede observar los requerimientos a desarrollar en este sprint en la tabla 20.

5.3.3 Duración del Sprint

El sprint 3 en base a la planificación y estimación del proceso de desarrollo tendrá una duración de 70 horas de trabajo.

5.3.4 Desarrollo

Para el desarrollo se empezará construyendo los Endpoints en el Backend y luego las pantallas asociadas a la historia de usuario.

5.3.4.1 Backend: Resumen de épica de Entradas

En la tabla 21 se muestran los recursos del backend para las historias de usuarios 43, 44, 45 y 46 correspondiente a la épica de Entradas.

Tabla 21

Historia de usuario 43, 44, 45 y 46: Endpoints

Historia de usuario	Método	URI	Descripción
43	POST	/api/entradas	Esté método nos permite registrar una nueva entrada, con sus datos asociados desde el perfil de administrador.
44	PUT	/api/entradas	Este método permite actualizar los datos de una entrada desde el perfil de administrador.
45.1	GET	/api/entradas /api/entradas/id	El primer método permite obtener una lista de todas las entradas registradas. El segundo método permite obtener una entrada por su id.
45.2	GET	/api/buscar/entradas/valor	Este método permite buscar una entrada por su concepto, nombre del producto o nombre de la bodega desde el perfil de administrador.
46	DELETE	/api/entradas/id	Este método permite eliminar una entrada de manera lógica de la aplicación desde el perfil de administrador.

5.3.4.2 Frontend: Historia de usuario 43

La historia de usuario 43 corresponde a la creación de una entrada, esta acción es realizada por parte del administrador.

Figura 93

Historia de usuario 43, validaciones para creación de una entrada.

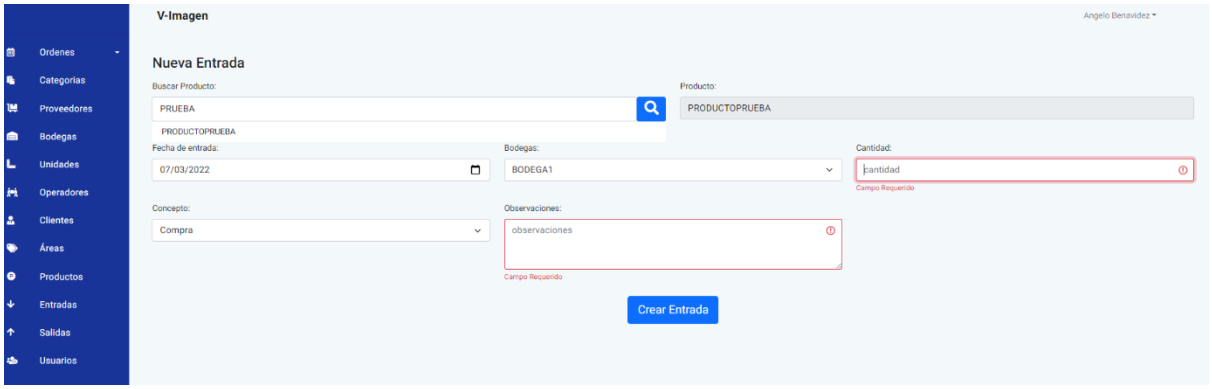
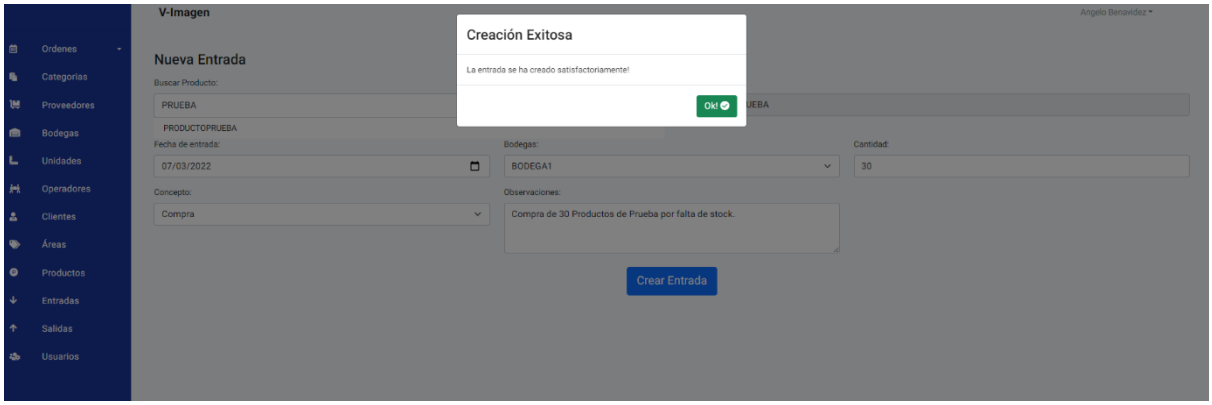


Figura 94

Historia de usuario 43, creación de una entrada.

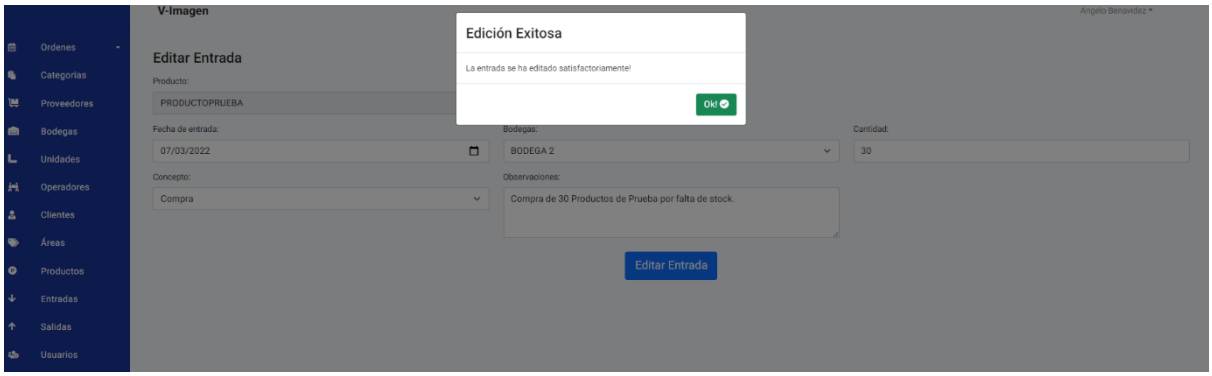


5.3.4.3 Frontend: Historia de usuario 44

La historia de usuario 44 corresponde a la actualización de una entrada, esta acción es realizada por parte del administrador.

Figura 95

Historia de usuario 44, modificación de una entrada.



5.3.4.4 **Frontend:** Historia de usuario 45

La historia de usuario 45 corresponde a la consulta general y por parámetros de entradas, esta acción es realizada por parte del administrador.

Figura 96

Historia de usuario 45, consulta general de entradas.

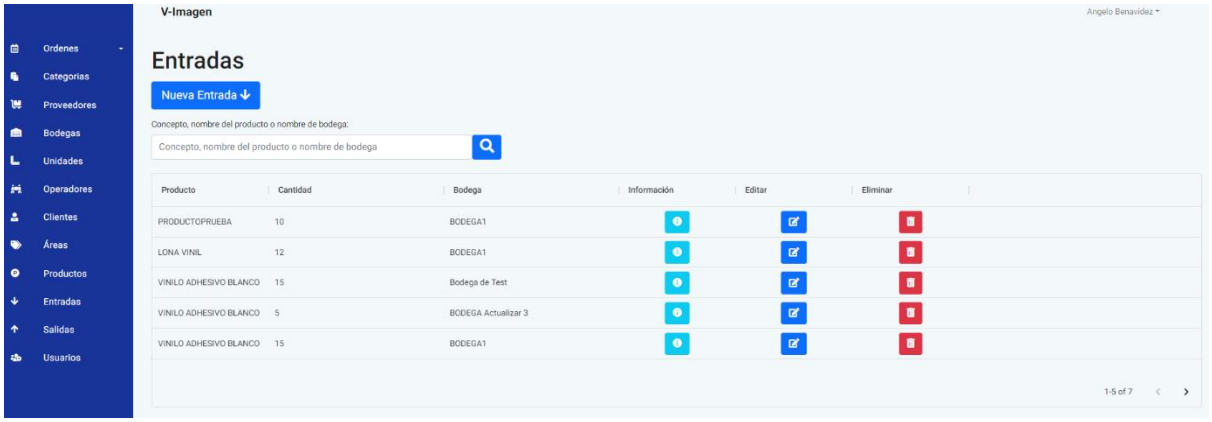
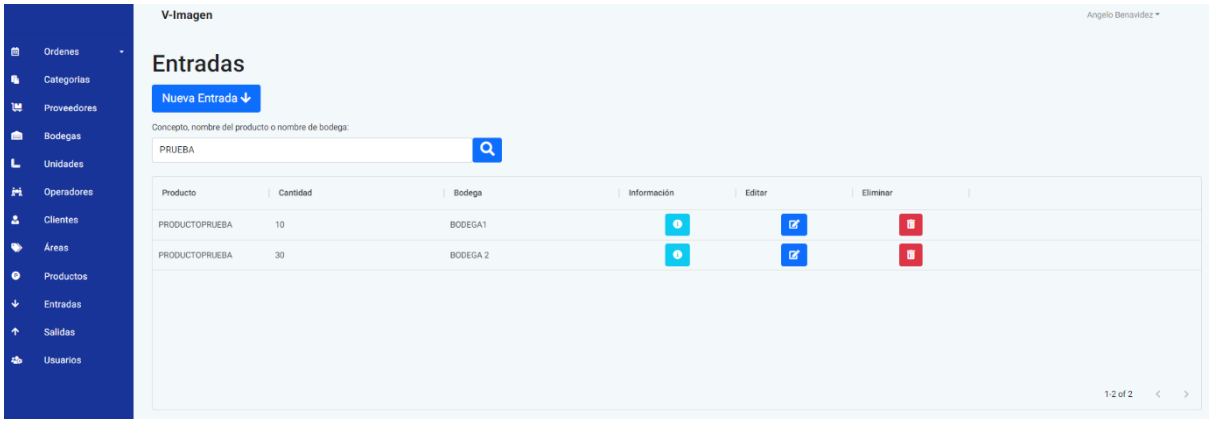


Figura 97

Historia de usuario 45, consulta por parámetros de entradas.

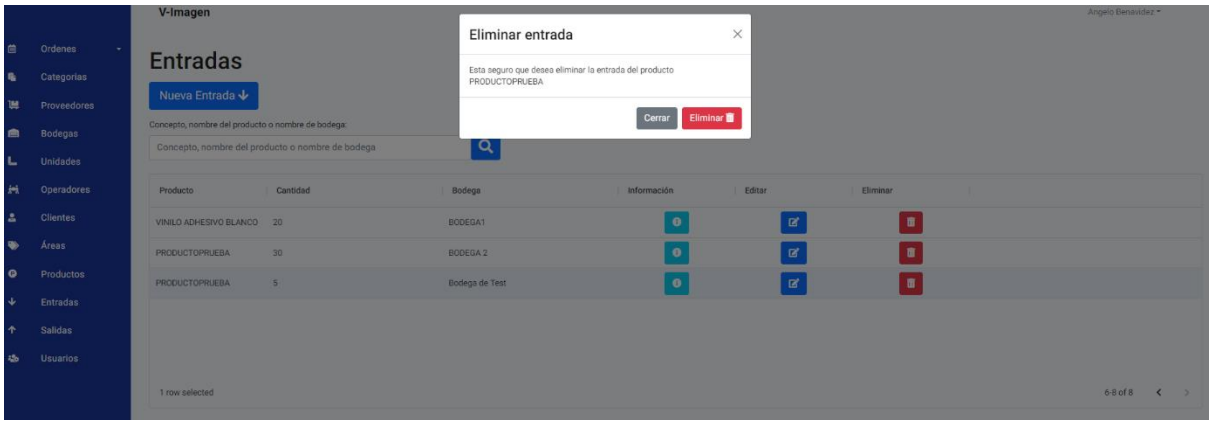


5.3.4.5 Frontend: Historia de usuario 46

La historia de usuario 46 corresponde a la eliminación de entradas, esta acción es realizada por parte del administrador.

Figura 98

Historia de usuario 46, eliminación de entradas.



5.3.4.6 Backend: Resumen de épica de Salidas

En la tabla 22 se muestran los recursos del backend para las historias de usuarios 47, 48, 49 y 50 correspondiente a la épica de Salidas.

Tabla 22

Historia de usuario 47, 48, 49 y 50: Endpoints

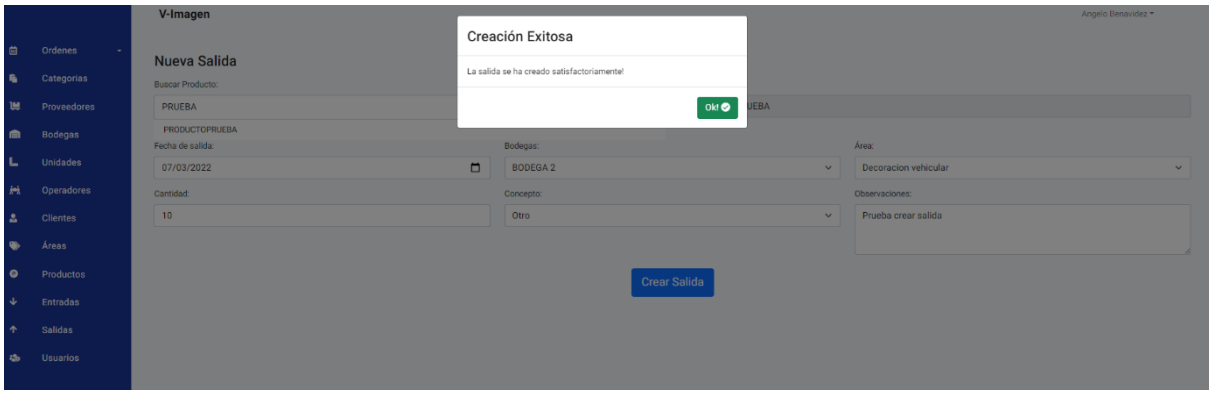
Historia de usuario	Método	URI	Descripción
47	POST	/api/salidas	Esté método nos permite registrar una nueva salida, con sus datos asociados desde el perfil de administrador.
48	PUT	/api/salidas	Este método permite actualizar los datos de una salida desde el perfil de administrador.
49.1	GET	/api/salidas/ /api/salidas/id	El primer método permite obtener una lista de todas las salidas registradas. El segundo método permite obtener una salida por su id.
49.2	GET	/api/buscar/salidas/valor	Este método permite buscar una entrada por su concepto, nombre del producto o nombre del área desde el perfil de administrador.
50	DELETE	/api/salidas/id	Este método permite eliminar una entrada de manera lógica de la aplicación desde el perfil de administrador.

5.3.4.7 Frontend: Historia de usuario 47

La historia de usuario 47 corresponde a la creación de una salida, esta acción es realizada por parte del administrador.

Figura 99

Historia de usuario 47, creación de salida.

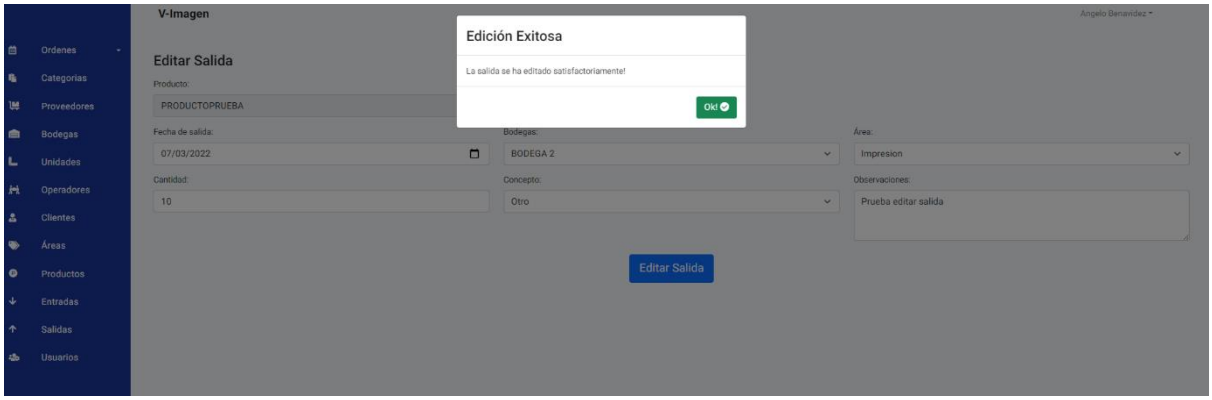


5.3.4.8 Frontend: Historia de usuario 48

La historia de usuario 48 corresponde a la actualización de una salida, esta acción es realizada por parte del administrador.

Figura 100

Historia de usuario 48, modificación de entradas.



5.3.4.9 Frontend: Historia de usuario 49

La historia de usuario 49 corresponde a la consulta general y por parámetros de salidas, esta acción es realizada por parte del administrador.

Figura 101

Historia de usuario 49, consulta general de salidas.

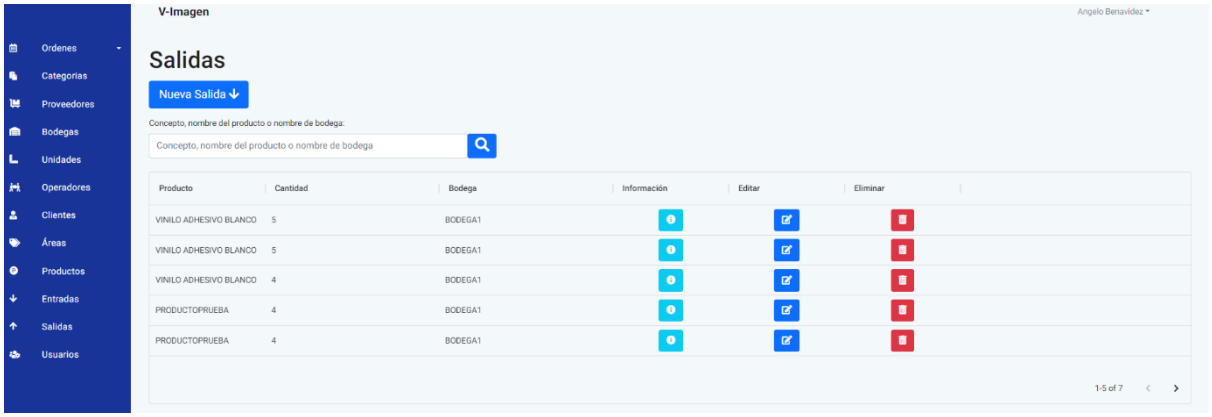
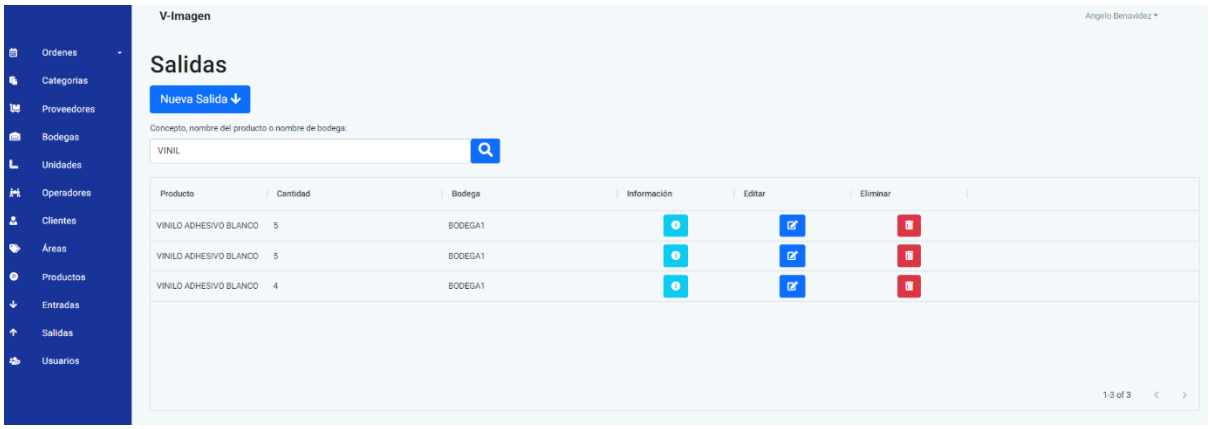


Figura 102

Historia de usuario 49, consulta por parámetro de salidas.

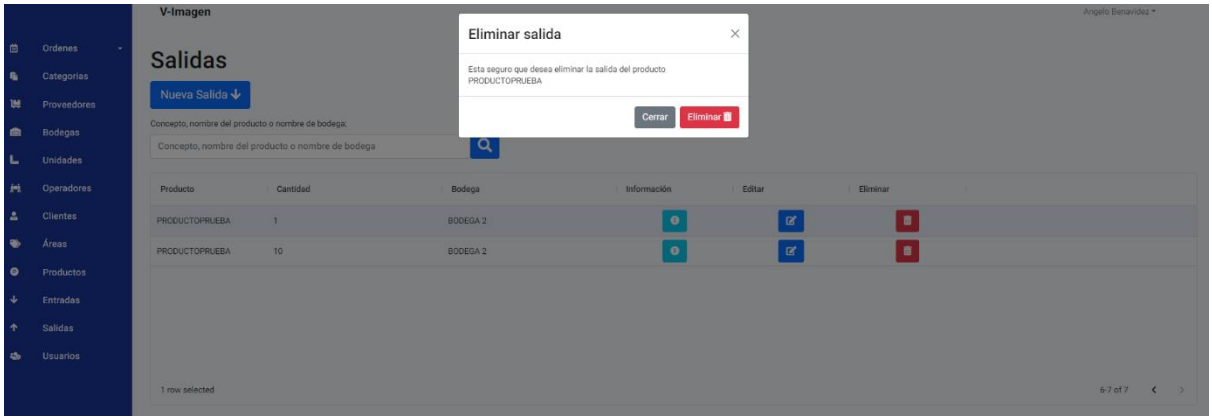


5.3.4.10 **Frontend:** Historia de usuario 50

La historia de usuario 50 corresponde a la eliminación de salidas, esta acción es realizada por parte del administrador.

Figura 103

Historia de usuario 50, eliminación de salidas.



5.3.4.11 **Backend:** Actualizar contraseña

En la tabla 23 se muestran los recursos del backend para la historia de usuario 51 correspondiente a actualizar contraseña.

Tabla 23

Historia de usuario 51: Endpoints

Historia de usuario	Método	URI	Descripción
51	PATCH	/api/usuarios/cp/id	Esté método nos permite cambiar la contraseña desde la pestaña cambiar contraseña, las condiciones son que se deba ingresar la contraseña actual, la nueva contraseña y la confirmación de la nueva contraseña, esta acción la pueden ejecutar administradores como operadores.

5.3.4.12 **Frontend:** Historia de usuario 51

La historia de usuario 51 corresponde al cambio de contraseña desde la pestaña cambiar contraseña, esta acción es realizada por parte del administrador y operador.

Figura 104

Historia de usuario 51, validaciones previo cambiar contraseña.

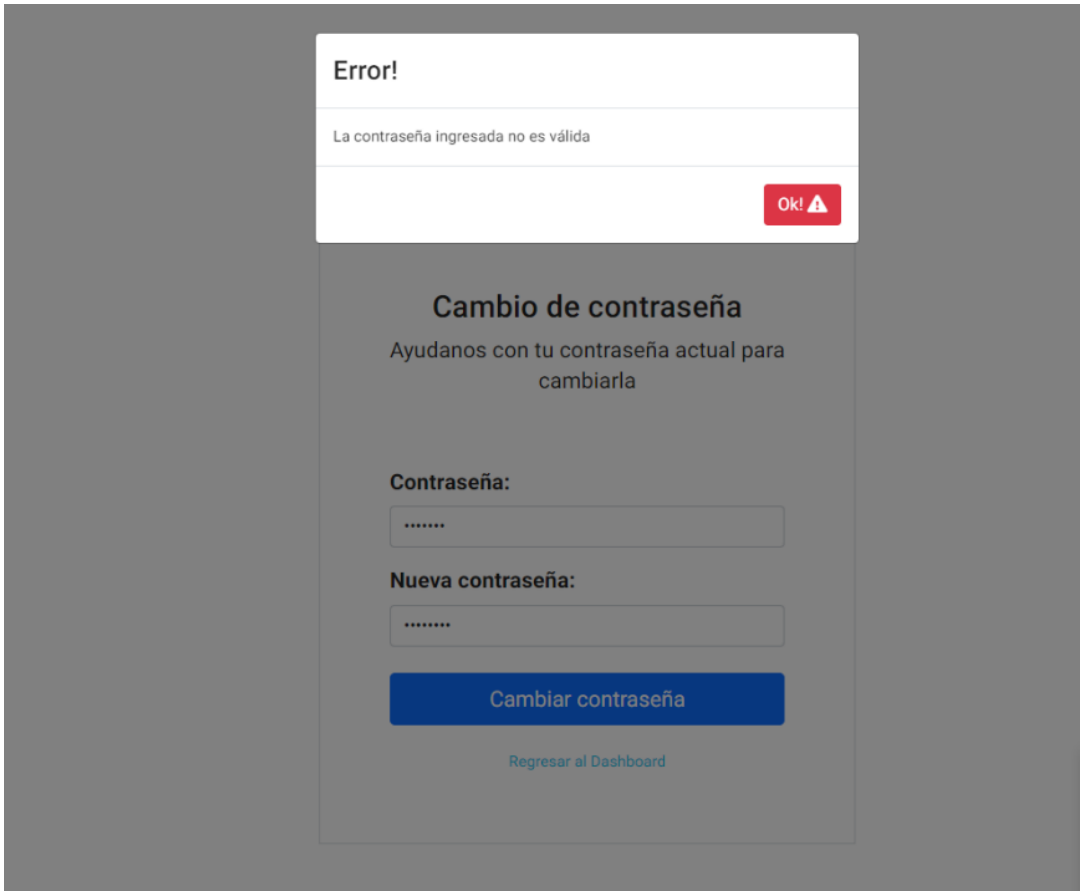
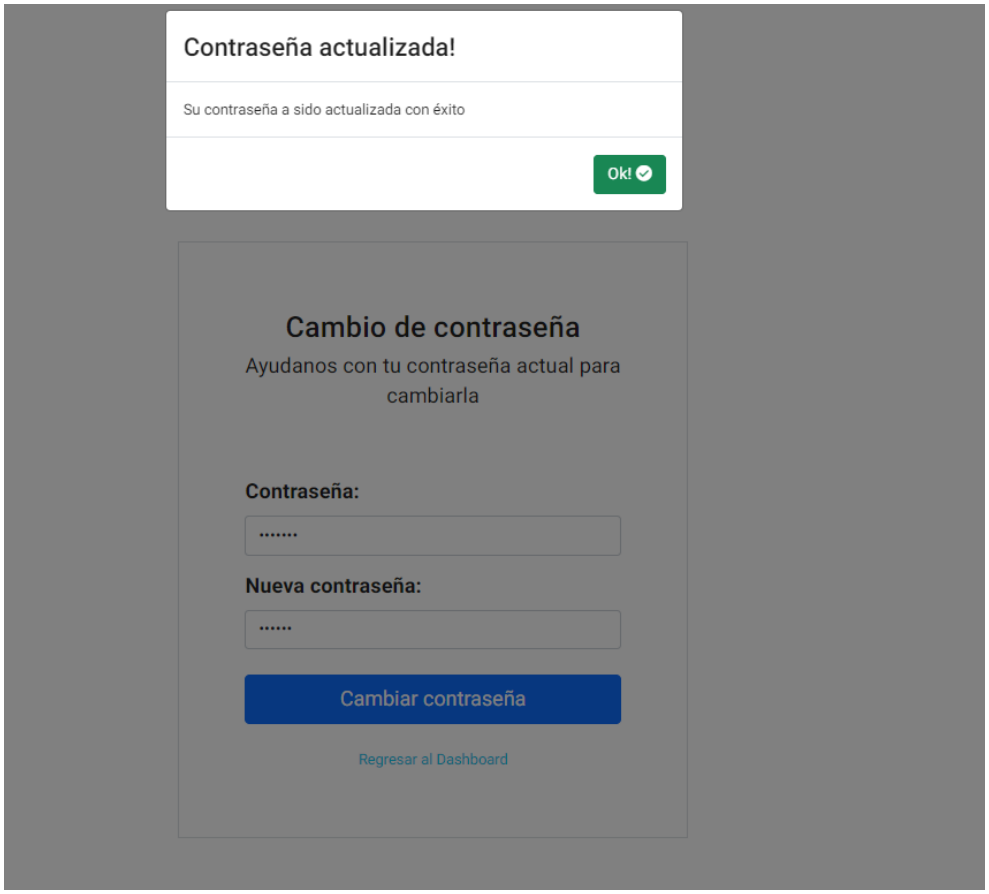


Figura 105

Historia de usuario 51, cambio de contraseña



5.3.4.13 **Backend:** Insertar Kardex

En la tabla 24 se muestran los recursos del backend para la historia de usuario 52 correspondiente a actualizar contraseña.

Tabla 24

Historia de usuario 52: Endpoints

Historia de usuario	Método	URI	Descripción
52	POST	NA	Esté método permite insertar un registro en el Kardex, una vez se ingrese un producto. El método no contiene una URI, al ser un proceso automático no requiere una acción adicional por parte del usuario.

5.3.4.14 **Frontend:** Historia de usuario 52

La historia de usuario 52 corresponde al registro de Kardex de un producto, el proceso inicia cuando se crea un producto, de forma simultánea el Kardex debe insertar el registro asociado a ese producto con balance de 0 entradas, 0 salidas y, por ende, 0 stock.

Figura 106

Historia de usuario 52, productos actuales.

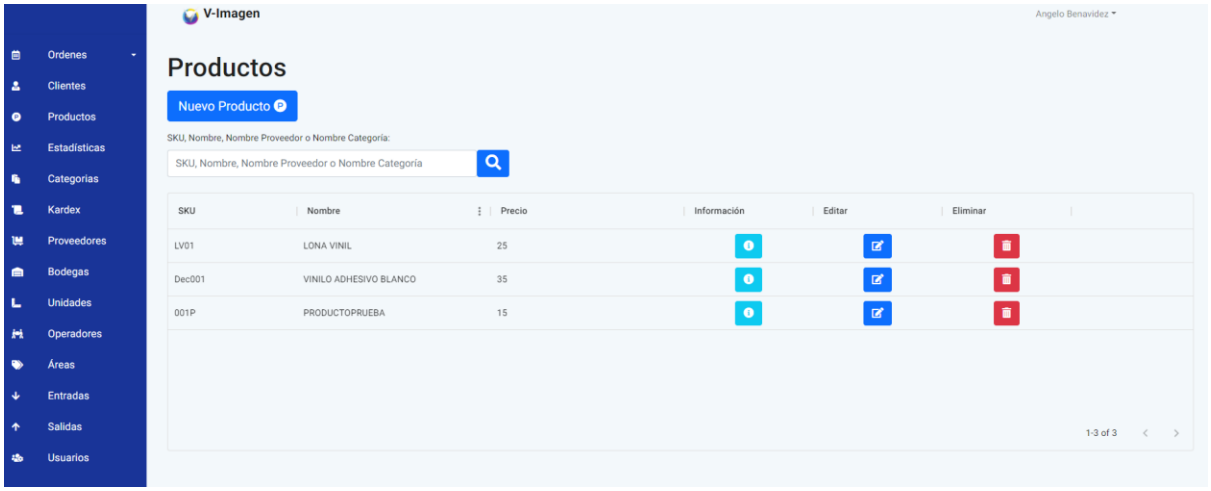


Figura 107

Historia de usuario 52, Kardex actual.

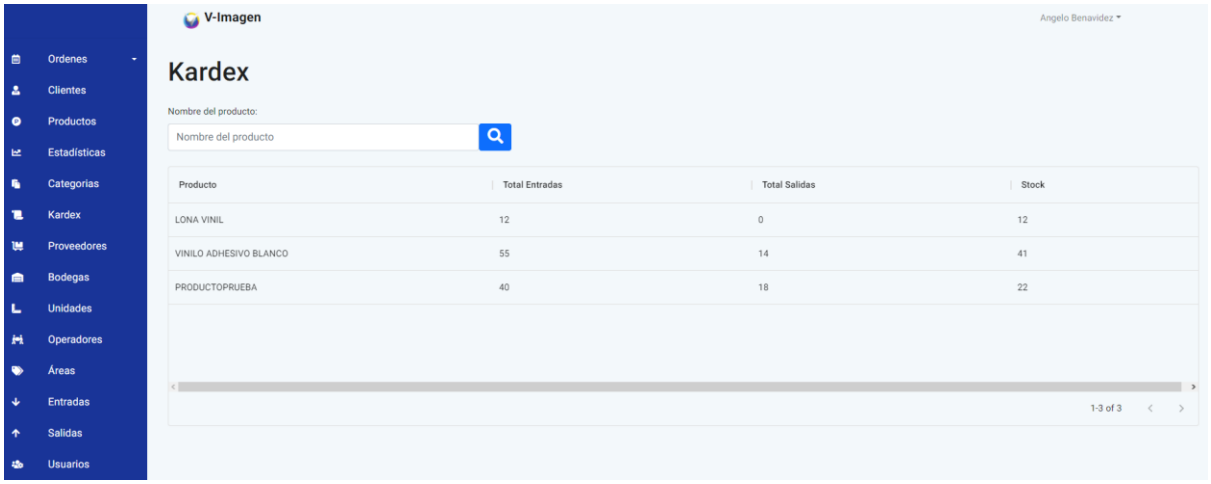


Figura 108

Historia de usuario 52, insertar producto.

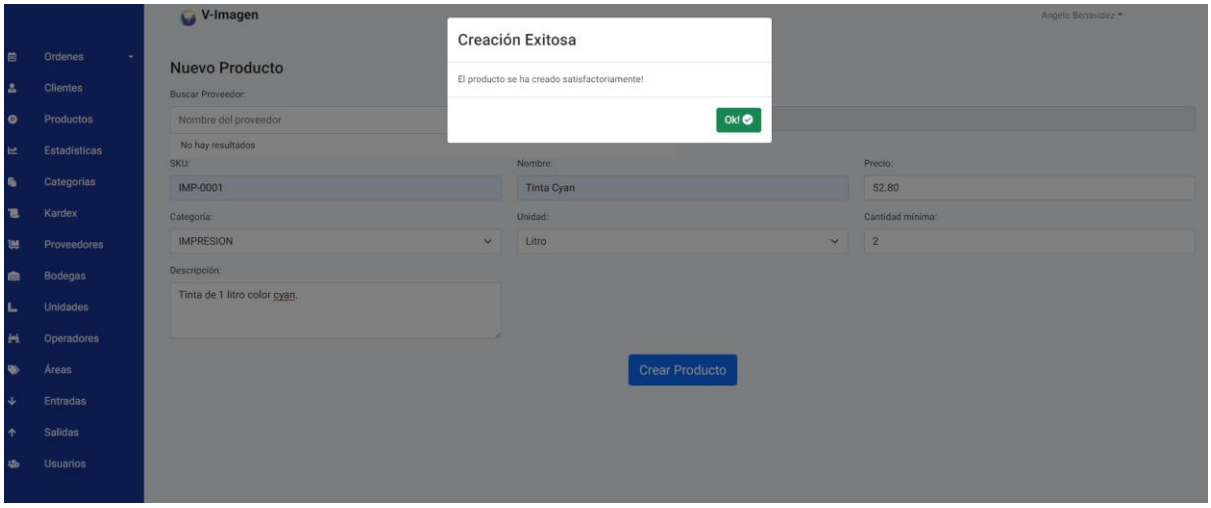
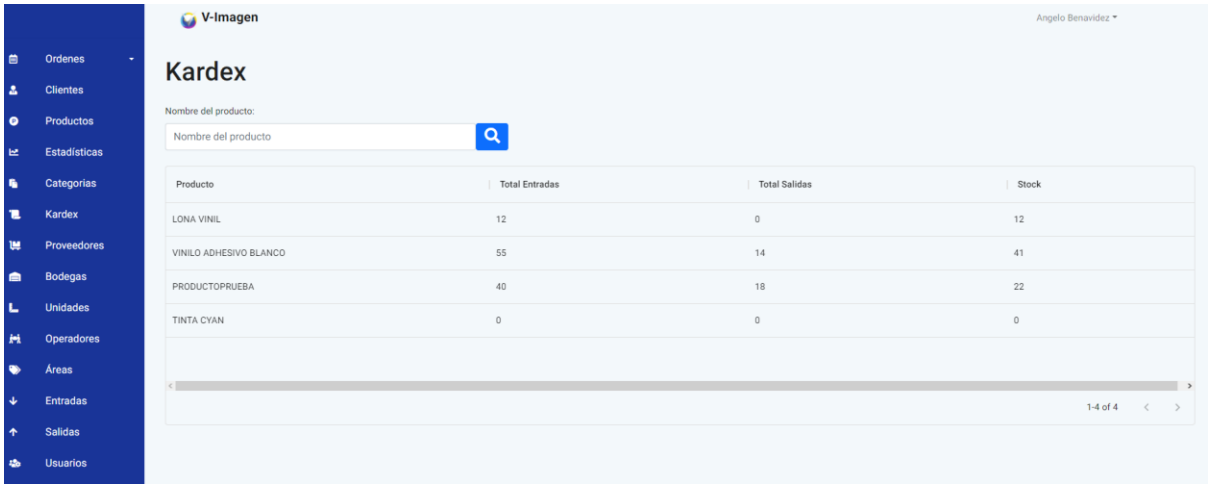


Figura 109

Historia de usuario 52, Kardex actualizado.



5.3.4.15 **Backend:** Actualizar Kardex por Entrada o Salida

En la tabla 25 se muestran los recursos del backend para la historia de usuario 53 correspondiente a actualizar el Kardex al realizar el proceso de entrada o salida de un producto, el Kardex debe calcular el nuevo stock en función de lo que se requiera.

Tabla 25

Historia de usuario 53: Endpoints

Historia de usuario	Método	URI	Descripción
53	PUT	NA	Esté método permite actualizar el Kardex, una vez que se produce una entrada el Kardex actualiza los valores, de igual manera al realizar una salida.

5.3.4.16 **Frontend:** Historia de usuario 53

La historia de usuario 53 corresponde a la actualización de Kardex de un producto, para este caso puede recibir una actualización de entradas como salidas de un producto. La validación para este caso es que no se puede realizar una salida sin previamente existir una entrada de un producto.

Figura 110

Historia de usuario 53, validación previa.

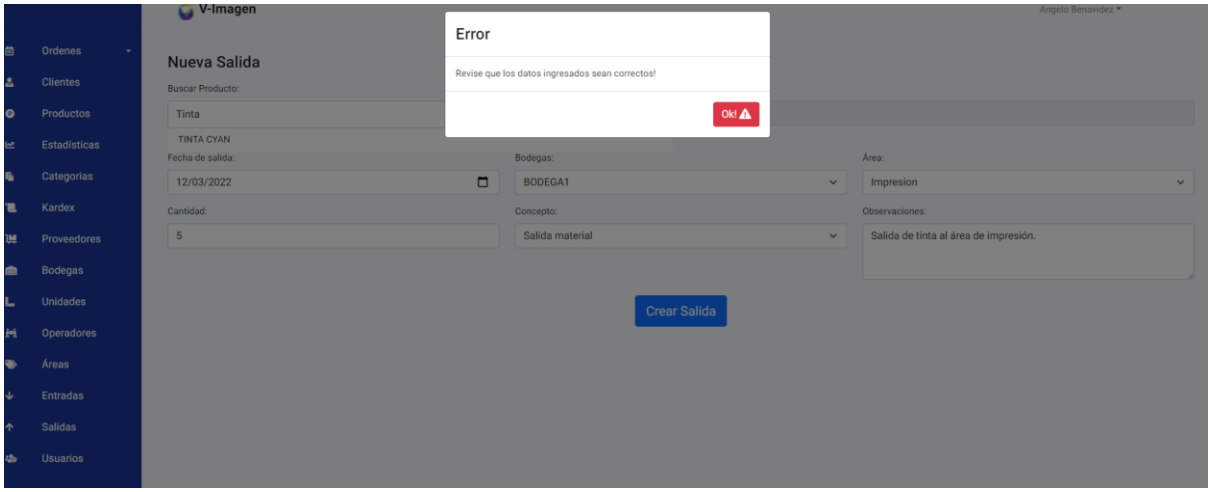


Figura 111

Historia de usuario 53, registrar entrada.

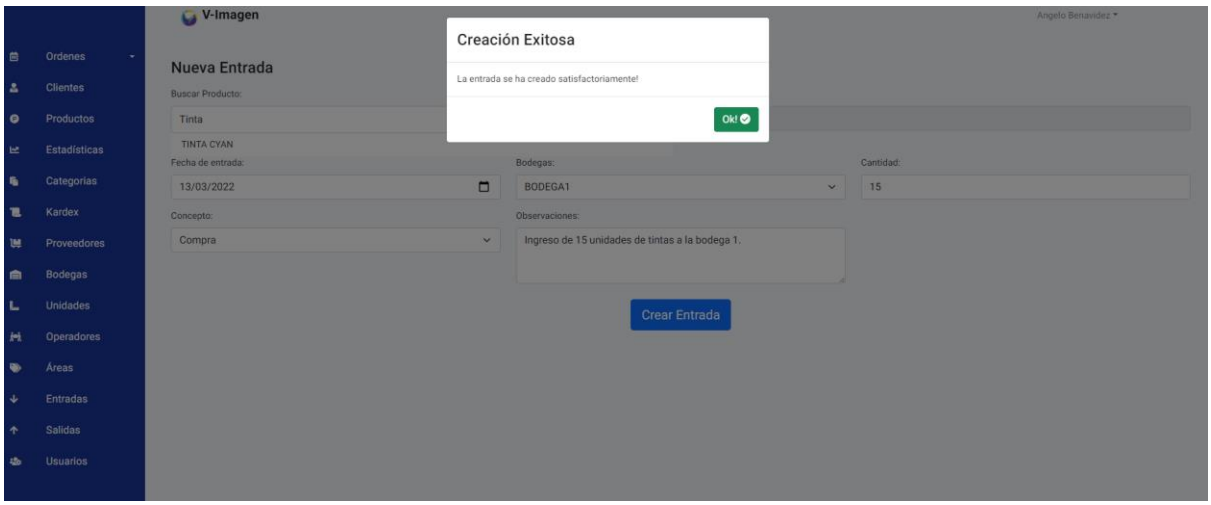


Figura 112

Historia de usuario 53, Kardex actualizado por entrada.

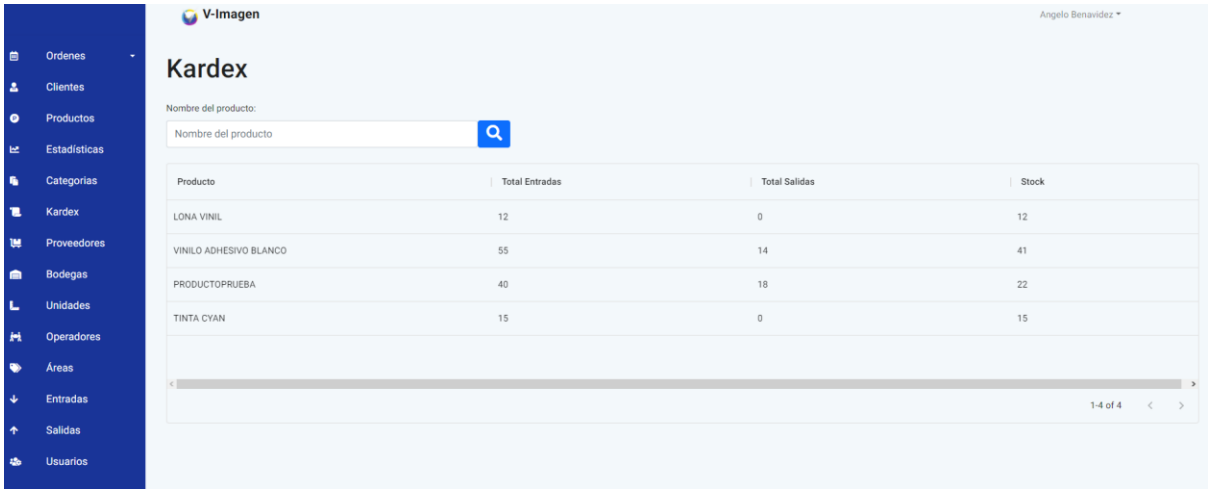
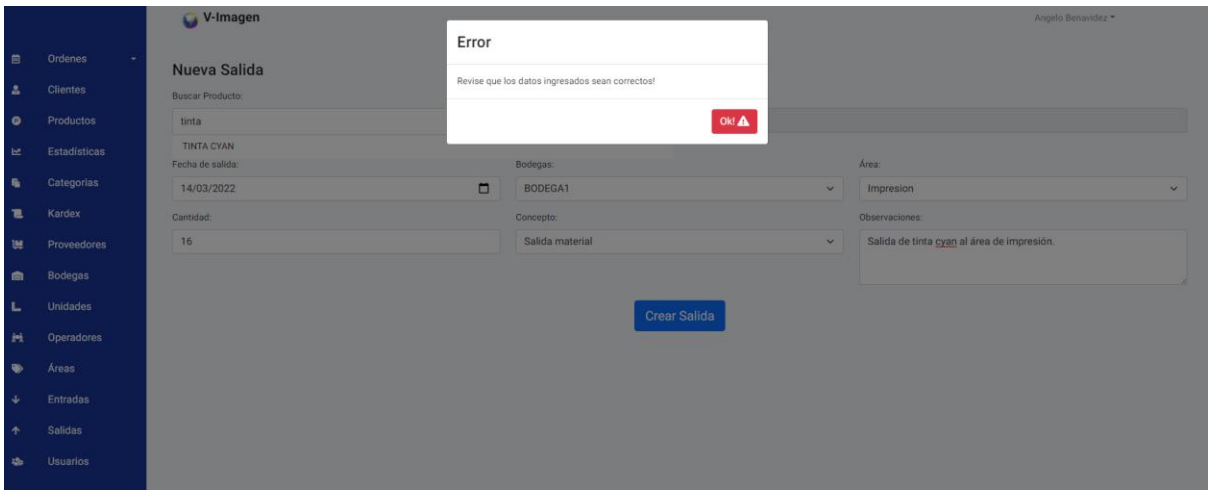


Figura 113

Historia de usuario 53, validación registro de salida.



Nota. El usuario no podrá realizar una salida de un material mayor a la suma del total de entradas para ese producto

Figura 114

Historia de usuario 53, registro de salida.

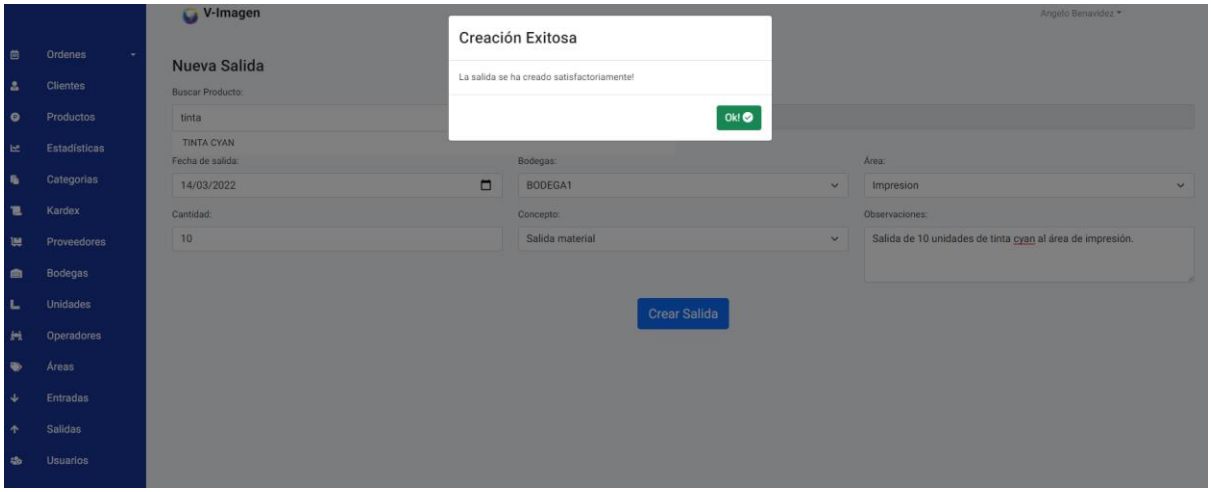
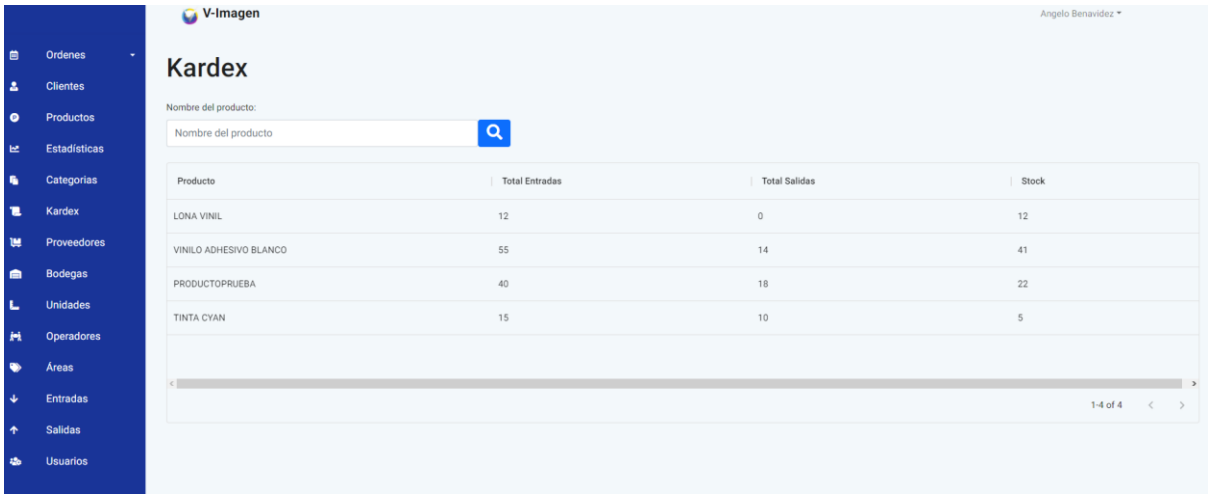


Figura 115

Historia de usuario 53, Kardex actualizado por salida.



5.3.4.17 **Backend:** Dashboard

En la tabla 26 se muestran los recursos del backend para la historia de usuario 54 correspondiente a las estadísticas generales, tiene la particularidad de actualizarse cada treinta segundos y presentar los nuevos datos al administrador.

Tabla 26

Historia de usuario 54: Endpoints

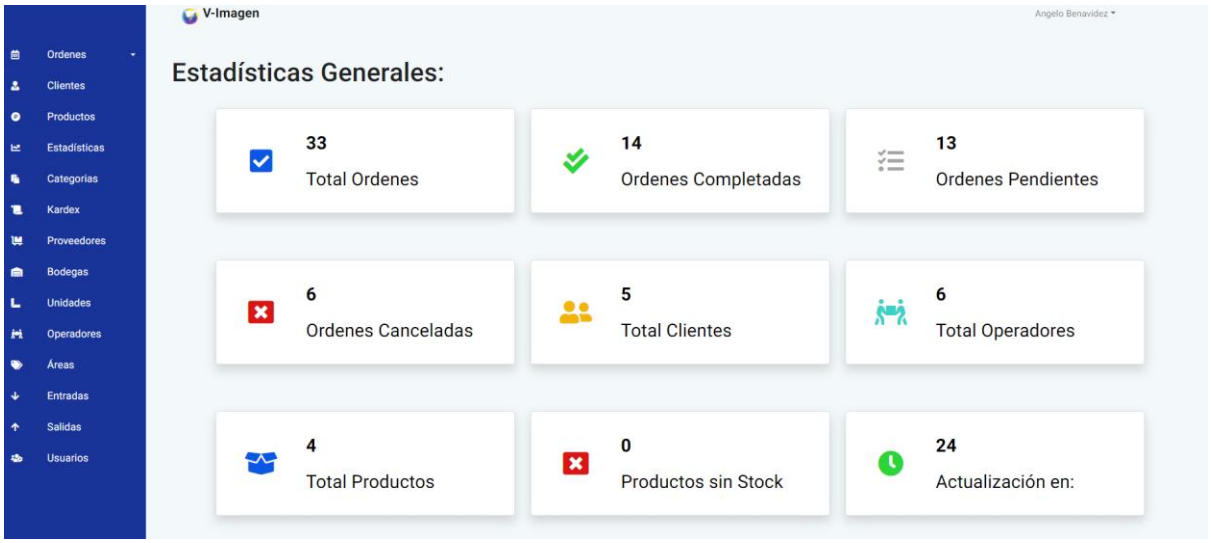
Historia de usuario	Método	URI	Descripción
54	GET	/api/estadisticas	Esté método obtener estadísticas de distintas colecciones del sistema, se actualiza cada treinta segundos, permite al administrador tener referencia de los principales procesos de la empresa.

5.3.4.18 **Frontend:** Historia de usuario 54

La historia de usuario 54 corresponde a la presentación de estadísticas de las principales colecciones del sistema involucrando los principales procesos. El administrador podrá observar un panel con los datos numéricos y su correspondiente colección.

Figura 116

Historia de usuario 54, estadísticas.



5.4 Pruebas

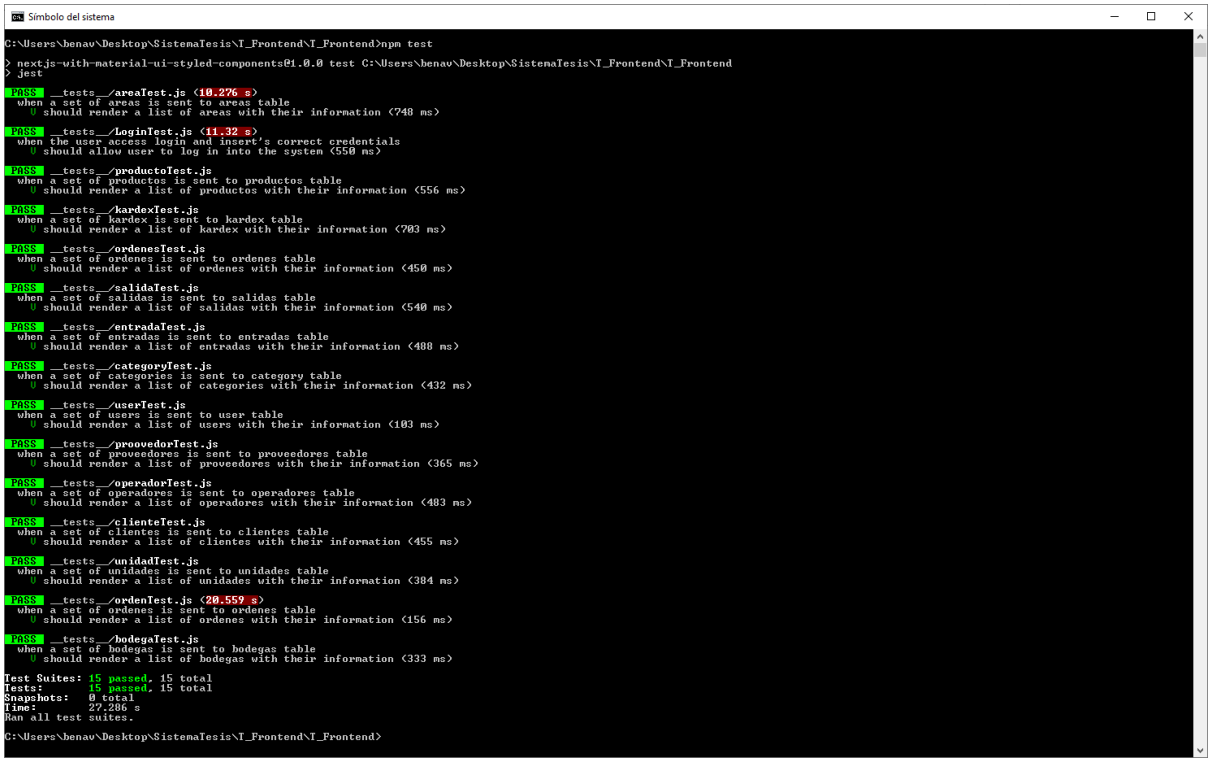
Una vez finalizado el desarrollo de los módulos se han realizado las pruebas necesarias a fin de garantizar la calidad del producto desarrollo, para las pruebas unitarias automatizadas se ha usado JEST, el cual es un framework muy popular usado en la industria del desarrollo web para hacer pruebas unitarias, una de sus principales características es la velocidad con que

se ejecutan las pruebas, es una librería potente que a su vez integra utilidades y recursos de otras librerías para realizar simulaciones sobre los sistemas.

En la figura 117 se aprecia el resultado de las pruebas ejecutadas bajo los estándares de JEST, se inicia ejecutando las pruebas que toman más tiempo de forma paralela, en este caso los módulos de áreas, ingreso al sistema y ordenes son los que más tiempo demoran en ejecutarse, resaltados en rojo en la figura, los demás módulos son ejecutados una vez terminan de pasar las pruebas de los que requieren de mayor tiempo, en total se requiere de 27.29 segundos para ejecutar las pruebas que corresponden a todos los módulos del sistema, garantizando que las funcionalidades estén correctas. Al final de la figura se aprecia un resumen donde detalla que de los 15 módulos que han sido sometidos a pruebas, han pasado 15 las con éxito.

Figura 117

Pruebas del sistema



5.5 Despliegue

Una vez se ha culminado el desarrollo de cada una de las historias de usuarios que conforman una épica o modulo del sistema, y las prueban unitarias y de aceptación han sido pasadas con éxito, se procedió a realizar el despliegue del sistema web en un servidor en la nube alojado en Vercel, para acceder al sistema web se debe ingresar al enlace: https://vimagen.vercel.app

6 CAPITULO VI: CONCLUSIONES Y RECOMENDACIONES

A continuación, se presentan las conclusiones y recomendaciones obtenidas una vez ha finalizado el desarrollo del sistema web.

6.1 CONCLUSIONES

- Se desarrolló el sistema web previo análisis y levantamiento de requerimientos para el plan de disertación “Análisis, Desarrollo e Implementación de un Sistema Web para el Control de Inventarios y Gestión de Ordenes de Trabajo, Caso de Estudio: empresa de soluciones gráficas V-Imagen de la Ciudad de San Lorenzo”, con el objetivo de automatizar procesos que a su vez permiten un control más eficiente del inventario y las ordenes de trabajo que se ejecutan diariamente.
- La aplicación de la metodología Scrum permite una participación constante de las partes involucradas, siendo esto un factor fundamental en el desarrollo de un proyecto, una vez se conocen las necesidades de los usuarios se procede a levantar los requerimientos mediante un proceso formal dando lugar a un artefacto importante dentro de Scrum como lo es el Product Backlog, donde se encuentran las funcionalidades a desarrollar.
- Elaborar un diseño de Base de Datos a partir del levantamiento de requisitos permite tener una idea clara de lo que se quiere construir, evitando redundancia de información, y ayudando a que los procesos sean más eficientes de ejecutar y dar seguimiento.
- Contar con una herramienta de control de versiones a nivel de desarrollo como Git es fundamental para mantener una historia de los cambios que se realizan en el proyecto, permitiendo regresar en el tiempo para casos fortuitos, así como

mantener estándares modernos del desarrollo de software que involucren la colaboración con demás profesionales. Además, el uso de estándares de programación permite que la legibilidad del código sea mayor, esto incluye nombrado de carpetas, archivos, abriendo a la posibilidad de que un nuevo desarrollador pueda continuar el proyecto con facilidad.

- La elección de las herramientas para el desarrollo influye en la construcción del sistema web, debido a ello se debe elegir una pila de desarrollo adecuada para el tipo de sistema que se está construyendo, para el presente la elección de la variación de la pila MERN (Mongo, Express, React, Node.JS) por MEEN (Mongo, Express, Next, Node.JS) ha sido adecuada, incluyendo a NEXT.JS que a través de su renderizado por servidor permite que la visualización de las páginas sea más rápida.
- La escalabilidad del sistema está garantizada, debido a la arquitectura con la que ha sido construido utilizando arquitectura en capas, en caso de requerir se puede agregar una nueva capa que consuma los servicios REST y pueda ser usada para construir una aplicación móvil.

6.2 RECOMENDACIONES

- Utilizar la metodología Scrum para desarrollos que involucren una comunicación permanente con cliente, así como incrementos constantes del producto, obteniendo retroalimentación inmediata que permita hacer cambios a medida que avanza la construcción del sistema.
- El sistema web realiza un borrado lógico para asegurar que datos importantes no se pierdan, por lo que es recomendable realizar periódicamente una tarea de depuración, o mantener respaldos mensuales de la base de datos.

- Realizar periódicamente mantenimiento al sistema web, para reemplazar funciones depreciadas u implementar mejoras que surjan con las nuevas versiones de los lenguajes de programación.
- Para implementación de nuevas funcionalidades se recomienda mantener los estándares de programación y metodología. Esto se refiere añadir nuevas historias de usuario por el lado de aplicación de Scrum, así como del lado de desarrollo involucra el contar con un ambiente de desarrollo donde se permita realizar pruebas con datos ajenos a los que se tiene en producción, permitiendo así trabajar en paralelo y posterior acoplar los nuevos cambios a producción sin verse afectado en ningún momento el sistema.
- Se recomienda ampliar las funcionalidades del sistema, implementando un módulo de facturación que este asociado directamente con las ordenes de trabajo generadas, así como la implementación de nuevas métricas de evaluación del desempeño de los colaboradores de la empresa.

Bibliografía

Amazon. (2017). *Bases de datos SQL | AWS*. AWS. https://aws.amazon.com/es/relational-database/

Amazon. (2019). *MongoDB en AWS: Quick Start*. Https://Aws.Amazon.Com. https://aws.amazon.com/es/quickstart/architecture/mongodb-atlas/

Amazon. (2020). *Bases de datos no relacionales | Bases de datos de gráficos | AWS*. AWS. https://aws.amazon.com/es/nosql/

Amazon. (2021). *¿Qué son los microservicios? | AWS*. AWS. https://aws.amazon.com/es/microservices/

Bahit, E. (2012). *Scrum & Extreme Programming Para Programadores*. 162. http://www.cursosdeprogramacionadistancia.com/static/pdf/material-sin-personalizar-agile.pdf

Blancarte Iturralde, O. J. (2020). *Introducción a la arquitectura de software – Un enfoque práctico* (O. J. Blancarte Iturralde (ed.); Primera Ed, Issue 20 20).

Cadavid, A. N., Fernández Martínez, J. D., & Morales Vélez, J. (2013). Revisión de metodologías ágiles para el desarrolldo de software A review of agile methoologies for software development. *Universidad Icesi*, *11 No. 2*, 30–39. dialnet.unirioja.es/descarga/articulo/4752083.pdf

Celis, R. (2018). *Heroku: qué es, cómo funciona y para qué sirve - Platzi*. Platzi.Com/Blog. https://platzi.com/blog/que-es-heroku/

Cohn, M. (2010). *Succeeding with Agile, Software development using Scrum*.

Durán, Y. (2012). Administración del inventario: elemento clave para la optimización de las

utilidades en las empresas. *Visión Gerencial*, *1*, 55–78. https://www.redalyc.org/pdf/4655/465545892008.pdf

Ehrhardt, M., & Brigham, E. (2007). Finanzas Corporativas. In *Cengage Learning* (Vol. 2).

Express.js. (2021). *Express - Infraestructura de aplicaciones web Node.js*. Express. https://expressjs.com/es/

Facebook Inc. (2021). *Jest · Delightful JavaScript Testing*. Jest Documentation. https://jestjs.io/es-ES/

Gutiérrez, J. J. (2010). Qué es un framework web ? *Universidad de Sevilla*, 1–4. http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

Herrera, E., & Valencia, L. E. (2007). Del manifiesto ágil sus valores y principios. *Scientia Et Technica*, *13*(34), 381–386. http://www.redalyc.org/html/849/84934064/%0Ahttp://www.redalyc.org/resumen.oa?id=84934064%5Cnhttp://www.redalyc.org/articulo.oa?id=84934064%0Ahttp://www.redalyc.org/articulo.oa?id=84934064

IBM. (2019a). *¿Qué es SOAP? - Documentación de IBM*. Https://Www.Ibm.Com/Docs/Es. https://www.ibm.com/docs/es/integration-bus/10.0?topic=ssmkhh-10-0-0-com-ibm-etools-mft-doc-ac55770--htm

IBM. (2019b). *Definición de casos de uso - Documentación de IBM*. Https://Www.Ibm.Com/Docs/Es/. https://www.ibm.com/docs/es/elm/6.0.3?topic=requirements-defining-use-cases

JWT. (2017). JSON Web Token Introduction - jwt.io. In *Jtw.Io*. https://jwt.io/introduction

Karnik, N. (2018). *Introduction to Mongoose for MongoDB*. FreeCodeCamp.

https://www.freecodecamp.org/news/introduction-to-mongoose-for-mongodb-d2a7aa593c57/

Lerma Blasco, R., Murcia Andrés, J. A., & Musfad Talón, E. (2013). *Aplicaciones web Aplicaciones Web Aplicaciones web Materiales del proyecto*. www.mhe.es/cf/informatica

Luján Mora, S. (2002). Programación de aplicaciones web: historia, principios básicos y clientes web. In *Programación de aplicaciones web: historia, principios básicos y clientes web* (Editorial, p. 48). https://rua.ua.es/dspace/bitstream/10045/16995/1/sergio_lujan-programacion_de_aplicaciones_web.pdf

Martínez Orencio, A. (2017). *La información en la organización, su gestión y auditoría*. GestioPolis. https://www.gestiopolis.com/la-informacion-en-la-organizacion-su-gestion-y-auditoria/

Meana Coalla, P. P. (2017). *Gestión de inventarios*. https://books.google.es/books?hl=es&lr=&id=Ml5IDgAAQBAJ&oi=fnd&pg=PP1&dq=tipos+de+inventarios&ots=6wsarqCoG1&sig=AOC7B8jlmwo3iuevlzNJnK2RUJQ#v=onepage&q=tipos de inventarios&f=false

Menzinsky, A., López, G., Palacio, J., Sobrino, M. Á., Álvarez, R., & Rivas, V. (2020). Historias de Usuario. In *de Scrum Manager, Madrid, Creative Commons* (3rd ed.). https://scrummanager.net/files/scrum_manager_historias_usuario.pdf

Microsoft. (2016). *Un paseo por C# - Guía de C# | Microsoft Docs*. 09/08/2016. https://docs.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/

Microsoft. (2021). *¿Qué es Git? - Azure DevOps | Microsoft Docs*. Https://Docs.Microsoft.Com. https://docs.microsoft.com/es-es/devops/develop/git/what-is-git

MongoDB. (2019). *¿Qué es MongoDB? | MongoDB*. ¿Qué Es MongoDB? | MongoDB. https://www.mongodb.com/es/what-is-mongodb

MongoDB. (2021). *Explicación sobre las bases de datos NoSQL | MongoDB*. Mongodb.Com. https://www.mongodb.com/es/nosql-explained

Nodemailer. (2020). *Nodemailer :: Nodemailer*. Https://Nodemailer.Com/About/. https://nodemailer.com/about/

odoo. (2020). *El Sistema de Manejo de Inventario en Código Abierto #1 | Odoo*. https://www.odoo.com/es_ES/app/inventory

OpenJS. (2021). *Node.js*. Nodejs.Org. https://nodejs.org/es/about/

Oracle. (2019). ¿Qué es una base de datos? | Oracle México. In *Oracle* (p. 1). https://www.oracle.com/mx/database/what-is-database/

Ortega Marqués Ana, Padilla Domínguez Sandy, Torres Durán Johana, & Ruz Gómez Alexander. (2017, December 1). Nivel de importancia del control interno de los inventarios dentro del marco conceptual de una empresa | Liderazgo Estratégico. *Liderazgo Estratégico, Revista de Divulgación de La Facultad de Administración y Negocios de La Universidad Simón Bolívar*. http://revistas.unisimon.edu.co/index.php/liderazgo/article/view/3261

Pérez, J. E. (2009). *Introduccion a JavaScript*. www.librosweb.es

Red Hat. (2019a). *¿Qué es la arquitectura basada en eventos? |*. Red Hat. https://www.redhat.com/es/topics/integration/what-is-event-driven-architecture

Red Hat. (2019b). *¿Qué es una API?* Documentación Proporcionada Por Red Hat Sobre El Concepto de API (Application Programming Interface).

https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces

Red Hat. (2019c). *Diferencias entre REST y SOAP*. Red Hat. https://www.redhat.com/en/topics/integration/whats-the-difference-between-soap-rest

Red Hat. (2020). *¿Qué es una API de REST?* ¿Qué Es Una API de REST? https://www.redhat.com/es/topics/api/what-is-a-rest-api

Red Hat. (2021). *¿Qué es la arquitectura orientada a los servicios (SOA)?* Red Hat. https://www.redhat.com/es/topics/cloud-native-apps/what-is-service-oriented-architecture

Salazar, J. C., Tovar, Á., Linares, J. C., Lozano, A., & Valbuena, L. (2018). Scrum versus XP : similitudes y diferencias Scrum vs XP : Similarities and Differences. *Tia*, *6*(2), 29–37. https://revistas.udistrital.edu.co/index.php/tia/article/view/10496/14690

Sitio oficial de Vue.js. (2014). *Introducción - Vue.js*. Vuejs.Org. https://es.vuejs.org/v2/guide/

Taylor, D., & Cabrera, M. (2019). *Los 5 mejores software de gestión de inventarios gratis y de código abierto*. Capterra. https://www.capterra.es/blog/566/software-gestion-inventarios-gratis-y-codigo-abierto

Ucha, F. (2013). *Definición de Orden de trabajo » Concepto en Definición ABC*. Definicionabc.Com/. https://www.definicionabc.com/general/orden-de-trabajo.php

W3C. (2010). *Web of Services - {W3C}*. https://www.w3.org/standards/webofservices/

ANEXOS

EMPRESA DE SOLUCIONES GRÁFICAS **V-IMAGEN**

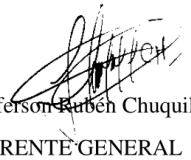
Aceptación del Sistema Web para el Control de Inventarios y Gestión de Ordenes de Trabajo

San Lorenzo, 20 de marzo de 2022

Yo Jefferson Chuquilla con C.I. 0802122408, en calidad de Gerente General de la Empresa de soluciones gráficas V-Imagen, informo que el sistema web desarrollado para el trabajo de disertación denominado "ANÁLISIS, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA EL CONTROL DE INVENTARIOS Y GESTIÓN DE ORDENES DE TRABAJO, CASO DE ESTUDIO: EMPRESA DE SOLUCIONES GRÁFICAS V-IMAGEN DE LA CIUDAD DE SAN LORENZO", consta con los siguientes módulos, cuyas funcionalidades han sido comprobadas y aprobadas:

- Módulo de autenticación y autorización
- Módulo de gestión de usuarios
- Módulo de operadores
- Módulo de categorías
- Módulo de unidades
- Módulo de bodegas
- Módulo de proveedores
- Módulo de áreas
- Módulo de productos
- Módulo de ordenes de trabajo.
- Módulo de entradas
- Módulo de salidas
- Módulo de Kardex
- Tablero de estadísticas

En favor de lo expuesto, doy por concluido el proyecto y acepto el sistema web desarrollado por el Sr. Angelo Benavidez con C.I. 1004584031



Jefferson Rubén Chuquilla Guaita
GERENTE GENERAL

Av. Camilo Ponce Enrique y calle Simón Plata Torres Contacto: 0967092896
San Lorenzo – Esmeraldas - Ecuador