



**Diseño y despliegue de un sistema web de administración orientado a la optimización  
operacional de negocios retail para la ferretería Electrocat ubicada en la ciudad de Santo  
Domingo.**

Castillo Cujilan, Eddy Fernando y Pérez Garófalo, Christian Enrique

Departamento de Ciencias de la Computación


Carrera de Ingeniería en Tecnologías de la Información

Trabajo de integración curricular, previo a la obtención del título de Ingeniería en Tecnologías  
de la Información

Ing. Chica Moncayo, Luis Manuel

22 de febrero del 2023

## Reporte de Verificación de Contenido


CERTIFICADO DE ANÁLISIS  
magister

### Castillo\_Eddy\_Perez\_Christian\_UIC20251

3%  
Similitudes
 

< 1% Texto entre comillas

< 1% similitudes entre comillas

0% Idioma no reconocido

**Nombre del documento:** Castillo\_Eddy\_Perez\_Christian\_UIC20251.pdf

**ID del documento:** 2825fe77aa62a139b5014c903cb505c79034f342

**Tamaño del documento original:** 2,93 Mo

**Depositante:** Christian Alfredo Coronel Guerrero

**Fecha de depósito:** 23/2/2023


**Tipo de carga:** interface

**Fecha de fin de análisis:** 23/2/2023

**Número de palabras:** 27.137

**Número de caracteres:** 207.852

Ubicación de las similitudes en el documento:



**Fuentes principales detectadas**

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	<b>CASTILLO_ARIEL_MORRERA_EDISSON_UIC202251.pdf   CASTILLO_ARIEL_MO...</b> El documento proviene de mi biblioteca de referencias 1 fuente similar	1%		Palabras idénticas: 1% (412 palabras)
2	<b>www.primoinformatica.com   Requerimientos no funcionales: Ejemplos - La Oficina d...</b> http://www.primoinformatica.com/2015/05/requerimientos-no-funcionales-ejemplos.html 2 fuentes similares	< 1%		Palabras idénticas: < 1% (115 palabras)
3	<b>scrumguides.org   scrumguides.org/doc/scrumguides/v5/Scrum-Guide-ES.pdf</b> https://scrumguides.org/doc/scrumguides/v5/Scrum-Guide-ES.pdf 4 fuentes similares	< 1%		Palabras idénticas: < 1% (75 palabras)
4	<b>localhost   Desarrollar una aplicación para la planificación de proyectos empleando ...</b> http://localhost:8080/om/labbits/wares/3317150659/UT-UIC3G-PRG-ING-C15-277.pdf.es 2 fuentes similares	< 1%		Palabras idénticas: < 1% (66 palabras)
5	<b>UIC_Artiaga_Venegas.docx   UIC_Artiaga_Venegas_A216NAF</b> El documento proviene de mi grupo	< 1%		Palabras idénticas: < 1% (58 palabras)

**Fuentes con similitudes fortuitas**

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	<b>dialnet.unirioja.es</b> http://dialnet.unirioja.es/servlet/articulo?codigo=8549651.pdf#--base=Según Díaz, Giga & Pardo (2021), se ...	< 1%		Palabras idénticas: < 1% (31 palabras)
2	<b>www.ils.org</b> http://www.ils.org/CDs/2023/CD2022/Summerpapers/CAP1_30C.pdf	< 1%		Palabras idénticas: < 1% (31 palabras)
3	<b>repository.udistrital.edu.co   Modelado Bajo una Arquitectura de Microservicios de ...</b> http://repository.udistrital.edu.co/handle/13454754	< 1%		Palabras idénticas: < 1% (24 palabras)
4	<b>universidadvirtualcnci.mx</b> https://universidadvirtualcnci.mx/wp-content/uploads/2022/09/La-era-de-las-competencias-digitales-d...	< 1%		Palabras idénticas: < 1% (10 palabras)
5	<b>Tesina_Final_Sanchez_Romeo_16-02-2023-Anti plagio.docx   Tesina_Final_San...</b> El documento proviene de mi grupo	< 1%		Palabras idénticas: < 1% (14 palabras)


**Fuentes ignoradas** Estas fuentes han sido retiradas del cálculo del porcentaje de similitud por el propietario del documento.

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	<b>Castillo_Eddy_Perez_Christian_UIC202251.pdf   Castillo_Eddy_Perez_Christia...</b> El documento proviene de mi biblioteca de referencias	63%		Palabras idénticas: 63% (19.266 palabras)
2	<b>Alcivar_Génesis_Romero_Génesis_UIC202251.pdf   Alcivar_Génesis_Romero_...</b> El documento proviene de mi biblioteca de referencias	< 1%		Palabras idénticas: < 1% (116 palabras)
3	<b>Caiza_Klar_Grande_Xavier_UIC202251.pdf   Caiza_Klar_Grande_Xavier_UIC20...</b> El documento proviene de mi biblioteca de referencias	< 1%		Palabras idénticas: < 1% (51 palabras)
4	<b>Cervantes_Jessica_Rivas_Mariam_UIC202251.pdf   Cervantes_Jessica_Rivas_...</b> El documento proviene de mi biblioteca de referencias	< 1%		Palabras idénticas: < 1% (43 palabras)

**Fuentes mencionadas (sin similitudes detectadas)** Estas fuentes han sido citadas en el documento sin encontrar similitudes.

- <https://nodejs.org/es/>
- <https://angular.io/>
- <https://plainenglish.io/blog/angular-vs-react-vs-vue-js-which-is-the-best-choice-for-2022>
- <https://www.java.com/>

Firma



El texto es autenticado por:  
**LUIS MANUEL CHICA MONCAYO**

**Ing. Chica Moncayo, Luis Manuel**

**Director**



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

CERTIFICACIÓN

Certifico que el trabajo de integración curricular: "Diseño y despliegue de un sistema web de administración orientado a la optimización operacional de negocios retail para la ferretería Electrocat ubicada en la ciudad de Santo Domingo" fue realizado por los señores Castillo Cujilan, Eddy Fernando y Pérez Garófalo, Christian Enrique el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Santo Domingo de los Tsáchilas, 22 de febrero del 2023

Firma:



Ing. Chica Moncayo, Luis Manuel

C. C: 1717455610




DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

**RESPONSABILIDAD DE AUTORÍA**

Nosotros, **Castillo Cujilan, Eddy Fernando y Pérez Garófalo, Christian Enrique**, con cédulas de ciudadanía n° 2300172547 y 2350576589, declaramos que el contenido, ideas y criterios del trabajo de integración curricular: "**Diseño y despliegue de un sistema web de administración orientado a la optimización operacional de negocios retail para la ferretería Electrocat ubicada en la ciudad de Santo Domingo**" es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

**Santo Domingo de los Tsáchilas, 22 de febrero del 2023**

Firmas:

  
.....  
**Castillo Cujilan, Eddy Fernando**  
C.C.: 2300172547

  
.....  
**Pérez Garófalo, Christian Enrique**  
C.C.: 2350576589




DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN


### AUTORIZACIÓN DE PUBLICACIÓN

Nosotros **Castillo Cujilan, Eddy Fernando y Pérez Garófalo, Christian Enrique**, con cédulas de ciudadanía n° 2300172547 y 2350576589, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: : **"Diseño y despliegue de un sistema web de administración orientado a la optimización operacional de negocios retail para la ferretería Electrocat ubicada en la ciudad de Santo Domingo"** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi/nuestra responsabilidad.

**Santo Domingo de los Tsáchilas, 22 de febrero del 2023**

Firmas:

  
.....  
**Castillo Cujilan, Eddy Fernando**  
C.C.: 2300172547

  
.....  
**Pérez Garófalo, Christian Enrique**  
C.C.: 2350576589

### **Dedicatoria**

Dedico este proyecto a mi familia. Principalmente, a mis padres, por el apoyo incondicional en este largo proceso, con su paciencia, motivación, consejos para afrontar cada obstáculo y con a mejorar tanto como persona y profesional. A mis amigos más cercanos que han compartido diferentes momentos conmigo en todo este extenso camino y han animado a seguir adelante.

Eddy Fernando Castillo Cujilan

### **Dedicatoria**

Dedico este trabajo a mi familia, que siempre me ha apoyado a lo largo de mi carrera, y les agradezco su cariño, paciencia y apoyo incondicional. A mis amigos que han compartido conmigo esta etapa de mi vida y han sido parte esencial de mi crecimiento personal y académico. Gracias por su amistad, por sus risas y por sus consejos que me han ayudado a superar obstáculos y a disfrutar de mis éxitos.

Christian Enrique Pérez Garofalo

## **Agradecimiento**

Agradezco primeramente a Dios por su bendición, salud y vida en todo mi camino.

A mi familia, que me han animado a perseguir mis sueños y creer en mí durante todo este proceso.

A todos los docentes que formaron parte de mi preparación académica. Principalmente, a mi tutor de proyecto el Ing. Luis Chica, por su orientación, paciencia y apoyo en el desarrollo del presente trabajo.

A mis compañeros y amigos más cercanos por el apoyo en los diferentes momentos, sobre todo en los más difíciles, compartir horas de estudio y enseñanzas de manera mutua. De manera especial a mi compañero de tesis por el apoyo en la realización de este proyecto.

Por último, a todas aquellas personas en general que formaron parte de este largo camino y proceso de formación.

Eddy Fernando Castillo Cujilan



## **Agradecimiento**

Quisiera expresar mi sincera gratitud a todos los que me han ayudado a completar este trabajo. A mi tutor Ing. Luis Chica por su inestimable orientación, paciencia y sabiduría en todo momento. Gracias por compartir sus conocimientos conmigo, por motivarme y por creer en mí.

Gracias a mi familia por su apoyo incondicional, sus ánimos y su paciencia. Gracias por escucharme, por comprender mi ausencia en los momentos importantes y por estar siempre a mi lado.

Agradezco a mis amigos su amistad, compañerismo y alegría que me ayudaron a superar los momentos difíciles y a disfrutar de los buenos momentos.

Por último, doy las gracias a todas las personas que contribuyeron de un modo u otro a la realización de esta tesis, y a todas aquellas que no menciono en esta lista, pero que participaron en mi proceso de investigación.

¡Gracias a todos!

Christian Enrique Pérez Garófalo

## Índice de Contenido

Carátula .....	1
Reporte de Verificación de Contenido .....	2
Certificación .....	3
Responsabilidad de Autoría .....	4
Autorización de Publicación .....	5
Dedicatoria.....	6
Agradecimiento .....	8
Resumen .....	23
Abstract.....	24
Capítulo I: Descripción del Proyecto .....	25
Introducción.....	25
Antecedentes .....	27
Justificación.....	28
Alcance .....	29
Objetivos .....	30
Objetivo General.....	30
Objetivos Específicos.....	30
Capítulo II: Marco Teórico .....	31
Retail.....	31
Definición .....	31
Funcionamiento Operacional de los Negocios Retail .....	31

	11
Factores Operativos.....	32
Descripción de la Metodología .....	32
Metodología de trabajo ágil.....	32
Scrum .....	33
Definición. ....	33
Scrum Team.....	33
Artefactos de Scrum. ....	34
Descripción de las Herramientas de Desarrollo .....	35
Tecnologías Web.....	35
Node js. ....	35
Angular.....	35
Microservicios.....	37
Middlewares. ....	38
Java. ....	39
Spring Boot. ....	39
Jira Software. ....	39
Amazon Web Services. ....	39
Linode. ....	40
Herramientas de Calidad de Código .....	41
SonarQube.....	41
Apache JMeter. ....	42

	12
Herramientas de Contenedorización .....	42
Docker.....	42
Kubernetes.....	43
Servicios RESTful.....	43
Base de datos.....	44
Relacionales.....	45
PostgreSql.....	46
Capítulo III: Metodología.....	49
Análisis y requerimientos.....	49
Requerimientos Funcionales.....	49
Requerimientos No Funcionales .....	50
Requerimientos del sistema.....	51
Definición y Roles de proyecto .....	52
Planificación del product backlog.....	53
Definición de los Sprints .....	54
Planificación de los Sprints.....	57
Objetivo de sprint 1 .....	57
Sprint 1 .....	57
Objetivo de sprint 2 .....	60
Sprint 2 .....	60
Objetivo de sprint 3.....	61

Sprint 3 .....	62
Objetivo de sprint 4 .....	65
Sprint 4 .....	66
Objetivo de sprint 5 .....	68
Sprint 5 .....	68
Criterios de Aceptación .....	70
Sprint 1 .....	70
Sprint 2 .....	75
Sprint 3 .....	77
Sprint 4 .....	82
Sprint 5 .....	83
Capitulo IV: Resultados y Discusión.....	86
Revisión y Retrospectiva .....	86
Revisión Sprint 1 .....	86
Retrospectiva Sprint 1 .....	86
Revisión Sprint 2.....	87
Retrospectiva Sprint 2.....	88
Revisión Sprint 3.....	88
Retrospectiva Sprint 3.....	89
Revisión Sprint 4.....	89
Retrospectiva Sprint 4.....	90

	14
Revisión Sprint 5.....	91
Retrospectiva Sprint 5.....	91
Arquitectura del Sistema .....	92
Dependencias de desarrollo .....	94
Base de datos.....	96
Backend.....	96
Implementación.....	107
Resultado del Sprint 1.....	107
Modulo para la Gestión y Autenticación de Usuarios. ....	107
Burndown Chart Sprint 1. ....	112
Resultado del Sprint 2.....	113
Módulo para la gestión de clientes. ....	113
Burndown Chart Sprint 2. ....	115
Resultado del Sprint 3.....	116
Módulo para la Gestión de Inventario. ....	116
Módulo para la gestión de proveedores y pedidos.....	122
Burndown Chart Sprint 3. ....	125
Resultado del Sprint 4.....	126
Módulo para la Gestión de Ventas.....	126
Burndown Chart Sprint 4 .....	130
Resultado del Sprint 5.....	130

Módulo para la Gestión de Facturación.....	130
Burndown Chart Sprint 5 .....	133
Pruebas de Calidad de Código.....	134
Análisis de Código del Microservicio de Autenticación .....	134
Análisis de Código del Microservicio de Clientes .....	135
Análisis de Código del Microservicio de Productos .....	137
Análisis de Código del Microservicio de Ventas.....	139
Análisis de Código del Microservicio de Facturación.....	141
Pruebas de carga .....	143
Escenario.....	144
Listar Empleados.....	144
Listar ventas .....	146
Listar Clientes.....	149
Listar productos.....	151
Pruebas de Funcionalidad .....	153
Análisis de Resultados Obtenidos .....	162
Corrección y Mejoras del Sistema.....	162
Trabajos Futuros .....	164
Capítulo V: Conclusiones y Recomendaciones.....	166
Conclusiones.....	166
Recomendaciones.....	167

Bibliografia.....	169
-------------------	-----



## Índice de Figuras

Figura 1 Relación de precios entre Linode, AWS y Azure. ....	40
Figura 2 Arquitectura del sistema web .....	93
Figura 3 Interfaz gráfica del login del sistema .....	108
Figura 4 Interfaz gráfica para la recuperación de contraseña.....	108
Figura 5 Interfaz gráfica para crear nueva contraseña .....	109
Figura 6 Correo del código de recuperación .....	110
Figura 7 Interfaz gráfica para el registro de deudas. ....	111
Figura 8 Interfaz gráfica para listar las deudas pendientes.....	111
Figura 9 Interfaz gráfica para listar las deudas canceladas.....	112
Figura 10 Burndown Chart Sprint 1.....	112
Figura 11 Interfaz gráfica para registrar clientes. ....	113
Figura 12 Interfaz gráfica para actualizar información del cliente. ....	114
Figura 13 Interfaz gráfica para eliminar el cliente.....	114
Figura 14 Interfaz gráfica para listar los clientes. ....	115
Figura 15 Burndown Chart Sprint 2.....	116
Figura 16 Interfaz gráfica para listar los productos.....	117
Figura 17 Interfaz gráfica para actualizar la información de los productos. ....	117
Figura 18 Interfaz gráfica para eliminar el registro del producto.....	118
Figura 19 Interfaz gráfica para listar los registros de productos .....	118
Figura 20 Interfaz gráfica para registrar la información de bodega.....	119
Figura 21 Interfaz gráfica para actualizar la información de bodega.....	119
Figura 22 Interfaz gráfica para listar la información de bodega. ....	120
Figura 23 Interfaz gráfica para añadir stock. ....	120
Figura 24 Interfaz gráfica para actualizar información del inventario.....	121
Figura 25 Interfaz gráfica para actualizar información del inventario.....	121

Figura 26	Interfaz gráfica para registrar proveedor. ....	122
Figura 27	Interfaz gráfica para actualizar la información del proveedor.....	123
Figura 28	Interfaz gráfica para listar los proveedores.....	123
Figura 29	Interfaz gráfica para registro de pedidos. ....	124
Figura 30	Interfaz gráfica para actualizar la información de pedidos.....	124
Figura 31	Interfaz gráfica para listar la información de pedidos ordenados.....	125
Figura 32	Interfaz gráfica para listar la información de pedidos entregados.....	125
Figura 33	Burndown Chart Sprint 3.....	126
Figura 34	Interfaz gráfica para realizar la venta. ....	127
Figura 35	Interfaz gráfica para actualizar la venta.....	127
Figura 36	Interfaz gráfica para listar las ventas.....	128
Figura 37	Interfaz gráfica para reporte de caja.....	128
Figura 38	Interfaz gráfica para reporte de ingresos o egresos.....	129
Figura 39	Interfaz gráfica para reporte de productos.....	129
Figura 40	Burndown Chart Sprint 4.....	130
Figura 41	Interfaz gráfica de facturas generadas.....	131
Figura 42	Interfaz gráfica de facturas enviadas.....	131
Figura 43	Interfaz gráfica de facturas autorizadas.....	132
Figura 44	Interfaz gráfica de facturas anuladas.....	132
Figura 45	Burndown Chart Sprint 5.....	133
Figura 46	Calidad de código microservicio de autenticación.....	134
Figura 47	Correcciones de calidad de código de microservicio Autenticación.....	135
Figura 48	Calidad de código microservicio de clientes.....	136
Figura 49	Correcciones de calidad de código de microservicio Clientes.....	137
Figura 50	Calidad de código microservicio de productos.....	138
Figura 51	Correcciones de calidad de código de microservicio Productos.....	139

Figura 52	Calidad de código microservicio de Ventas.....	140
Figura 53	Correcciones de calidad de código de microservicio Ventas.....	141
Figura 54	Calidad de código microservicio de Facturación.....	142
Figura 55	Correcciones de calidad de código de microservicio Ventas.....	143
Figura 56	Recursos asignados a contenedor.....	144
Figura 57	Configuración API listar empleados.....	145
Figura 58	Petición con un usuario.....	146
Figura 59	Petición con 10 usuarios.....	146
Figura 60	Petición con 100 usuarios.....	146
Figura 61	Configuración API listar ventas.....	147
Figura 62	Petición con un usuario.....	147
Figura 63	Petición con 10 usuarios.....	148
Figura 64	Petición con 25 usuarios.....	148
Figura 65	Petición de prueba con 75 usuarios.....	149
Figura 66	Petición de prueba con 100 usuarios.....	149
Figura 67	Configuración de API listar clientes.....	150
Figura 68	Petición de prueba con un usuario.....	150
Figura 69	Petición de prueba con 10 usuarios.....	151
Figura 70	Petición de prueba con 40 usuarios.....	151
Figura 71	Configuración de API listar productos.....	152
Figura 72	Petición de prueba con un usuario.....	152
Figura 73	Petición de prueba con 10 usuarios.....	153
Figura 74	Petición de prueba con 60 usuarios.....	153
Figura 75	Prueba de Funcionamiento Login del Sistema.....	154
Figura 76	Prueba de Recuperación de Contraseña.....	154
Figura 77	Prueba de Funcionamiento Registro Empleados.....	155

Figura 78 Prueba de Funcionamiento Control de Deudas .....	156
Figura 79 Prueba de Funcionamiento de Clientes.....	157
Figura 80 Prueba de Funcionamiento de Registro Productos .....	158
Figura 81 Prueba de Funcionamiento de Registro Proveedores .....	158
Figura 82 Prueba de Funcionamiento de Registro de Inventario.....	159
Figura 83 Prueba de Funcionamiento de Realizar una Venta .....	160
Figura 84 Prueba de Funcionamiento de Realizar una Factura.....	161
Figura 85 Prueba de Funcionamiento de Realizar una Nota de Crédito.....	161
Figura 86 Acciones realizadas con éxito y errores de proceso.....	162
Figura 87 Corrección de prueba de registro de producto .....	163
Figura 88 Corrección de prueba de realizar una venta.....	164

## Índice de Tablas

Tabla 1	Angular Vs React Vs Vue.....	36
Tabla 2	Arquitecturas monolíticas vs. Microservicios .....	37
Tabla 3	Tabla comparativa de Linode vs AWS vs Azure y Google Cloud.....	41
Tabla 4	Tabla comparativa de MySQL y PostgreSQL. ....	46
Tabla 5	Requerimientos Funcionales .....	49
Tabla 6	Requerimientos No Funcionales.....	50
Tabla 7	Requerimientos del Sistema.....	52
Tabla 8	Definición de Roles .....	52
Tabla 9	Product Backlog .....	53
Tabla 10	Definición Sprint nº. 1 .....	54
Tabla 11	Definición Sprint nº. 2.....	55
Tabla 12	Definición Sprint nº. 3.....	55
Tabla 13	Definición Sprint nº. 4.....	56
Tabla 14	Definición Sprint nº. 5.....	56
Tabla 15	Sprint Goal 1 .....	57
Tabla 16	Planificación Detallada del Sprint no. 1 .....	58
Tabla 17	Sprint Goal 2 .....	60
Tabla 18	Planificación Detallada del Sprint no. 2 .....	61
Tabla 19	Sprint Goal 3 .....	62
Tabla 20	Planificación Detallada del Sprint no. 3 .....	62
Tabla 21	Sprint Goal 4 .....	66
Tabla 22	Planificación Detallada del Sprint no. 4 .....	66
Tabla 23	Sprint Goal 5 .....	68
Tabla 24	Planificación Detallada del Sprint no. 5 .....	69

Tabla 25	Criterios de aceptación del sprint no. 1.....	70
Tabla 26	Criterios de Aceptación del Sprint no. 2.....	75
Tabla 27	Criterios de Aceptación del Sprint no. 3.....	77
Tabla 28	Criterios de Aceptación del Sprint no. 4.....	82
Tabla 29	Criterios de Aceptación del Sprint no. 5.....	84
Tabla 30	Revisión del sprint 1. ....	86
Tabla 31	Revisión del sprint 1. ....	87
Tabla 32	Revisión del sprint 2. ....	87
Tabla 33	Retrospectiva y Retroalimentación del Sprint no. 2. ....	88
Tabla 34	Revisión del Sprint 3. ....	88
Tabla 35	Retrospectiva y Retroalimentación del Sprint no. 3. ....	89
Tabla 36	Revisión del Sprint 4. ....	89
Tabla 37	Retrospectiva y Retroalimentación del Sprint no. 4. ....	90
Tabla 38	Revisión del Sprint 5. ....	91
Tabla 39	Retrospectiva y Retroalimentación del Sprint no. 5. ....	91
Tabla 40	Servicios RESTful de Microservicio Autenticación .....	96
Tabla 41	Servicios RESTful del Microservicio Clientes .....	99
Tabla 42	Servicios RESTful del Microservicio Productos .....	100
Tabla 43	Servicios RESTful del Microservicio Ventas .....	102
Tabla 44	RESTful del Microservicio Facturación .....	106

## Resumen

Este trabajo muestra el desarrollo de un sistema web de administración orientado a la optimización operacional de negocios retail para la ferretería Electrocat ubicada en la ciudad de Santo Domingo; se utilizó la metodología SCRUM para estructurar adecuadamente las tareas y actividades a realizar, las cuales se dividieron en un total de cinco sprints; para el desarrollo del front-end se optó por el framework Angular y para el back-end se empleó Java con el framework Spring Boot y JavaScript como lenguaje de programación utilizando Node Js basado en una arquitectura de microservicios, adicionalmente se empleó la utilización de PostgreSQL como el gestor de base de datos relacional. El sistema consta de varios módulos funcionales: ventas, inventario, empresa, compras, recibos, informes y usuarios. Cada módulo está conectado a un microservicio desplegado en servidores del proveedor de servicios en la nube Linode y gestionado por Kubernetes para automatizar el despliegue, escalado y administración de los mismos. El resultado final es un sistema web escalable desarrollado sobre la base de una arquitectura de microservicios que puede adaptarse fácilmente a cualquier negocio retail. Proporciona todas las funciones básicas necesarias para una gestión adecuada, desde el control de inventarios y el control de ventas hasta los métodos de facturación electrónica y la visualización de informes. El rendimiento del sistema se ha comprobado mediante pruebas funcionales de los principales módulos del sistema, pruebas de carga a los microservicios y pruebas de calidad del código basado en las métricas del software SonarQube.

**Palabras clave:** microservicios, retail, facturación, inventario, administración.

## Abstract

This work shows the development of a web management system oriented to the operational optimization of retail business for Electrocat hardware store located in the city of Santo Domingo; the SCRUM methodology was used to properly structure the tasks and activities to be performed, which were divided into a total of five sprints; For the front-end development, the Angular framework was chosen and for the back-end, Java was used with the Spring Boot framework and JavaScript as programming language using Node Js based on a microservices architecture; additionally, PostgreSQL was used as the relational database manager. The system consists of several functional modules: sales, inventory, company, purchases, receipts, reports and users. Each module is connected to a microservice deployed on servers from cloud service provider Linode and managed by Kubernetes to automate deployment, scaling and administration. The end result is a scalable web system developed on the basis of a microservices architecture that can be easily adapted to any retail business. It provides all the basic functions needed for proper management, from inventory control and sales control to electronic invoicing methods and report viewing. System performance has been tested through functional testing of the main system modules, load testing of the microservices and code quality testing based on SonarQube software metrics.

**Keywords:** microservices, retail, invoicing, inventory, administration.



## Capítulo I: Descripción del Proyecto

### Introducción

El presente documento abarca el desarrollo de un sistema para la administración operacional de los negocios retail mediante una plataforma web para la ferretería Electrocat ubicada en la provincia de Santo Domingo de Tsáchilas, utilizando una metodología ágil de desarrollo de software.

En la actualidad, el avance tecnológico ha obligado de cierta manera a las PYMES (pequeñas y medianas empresas) a implementar sistemas para la administración del negocio como también la atención al cliente. Para ello se hace uso de las TIC (Tecnologías de la información y comunicación) para la automatización de procesos, ya sea como un control de inventario, las ventas realizadas, registro de proveedores, clientes, controles tributarios (Facturación) entre otros. Para entrar en contexto sobre la transformación digital o digitalización en una empresa se conoce la misma como, “proceso de transformación de su organización, proceso productivo, estrategia competitiva y modelo de negocio vinculado a la adopción de tecnologías digitales” (Dini , Gligo, & Patiño, 2021, pág. 7)

La Ferretería Electrocat actualmente no cuenta con un sistema de administración que le permita automatizar las diferentes operaciones mencionadas anteriormente. Los controles operacionales del local se realizan de forma manual en software básicos como hojas de cálculo y en algunas ocasiones en registros físicos como libretas, cuadernos y hojas, lo cual no brinda una transparencia y eficacia en los procesos administrativos del negocio.

Dada la problemática planteada con anterioridad, se propone como una solución la implementación de un sistema web enfocado en la arquitectura de microservicios, de los cuales se dividirán basándose en la funcionalidad que realizará cada uno. La implementación de la solución tecnológica se encontrará dividido en dos apartados, el primero enfocado en el front-end o interfaces gráficas con las cuales el usuario interactúa; y el segundo apartado que se trata del back-end o también conocido como el lado del servidor, en donde se implementaran

un total de cinco microservicios. Se encuentran divididos de la siguiente manera:

Autenticación, Clientes, Productos, Ventas y Facturación.

Adicionalmente cabe mencionar que para el apartado de la seguridad se tomó como referencia las recomendaciones estipuladas en el informe “OWASP Top 10”, el cual es un documento de concienciación estándar para desarrolladores y seguridad de aplicaciones web. Representa un amplio consenso sobre los riesgos de seguridad más críticos para las aplicaciones web. Como guía la versión utilizada es la del año 2021, en la que se describen los 10 principales riesgos de seguridad de las aplicaciones web (Owasp, 2021).

El proyecto está conformado por cinco capítulos los cuales se visualizan y describen de la siguiente forma:

El primer capítulo, se enfoca en la descripción del proyecto, el cual se basa en la introducción, antecedentes, justificación, alcance y los objetivos correspondientes del proyecto tanto general como específicos.

El segundo capítulo, se basa en el marco teórico, en el cual se describen proyectos similares realizados, la descripción de la metodología a utilizar y la descripción de las herramientas seleccionadas para el desarrollo del proyecto.

Dentro del tercer capítulo, se basa la metodología implementada, para ello se seleccionó la metodología SCRUM, identificando los roles que conlleva la misma, el análisis de los requerimientos, la planificación con los sprints realizados en el desarrollo del proyecto.

Para el cuarto capítulo, se enfoca en el caso de estudio, el cual abarca la aplicación del sistema, el análisis de los resultados obtenidos y los trabajos futuros.

Para el quinto y último capítulo, se basa en las conclusiones y recomendaciones que se obtuvieron en el desarrollo del sistema, como también las deducciones de las herramientas y metodología utilizada.

## Antecedentes

Hoy en día cada vez son más las PYMES que invierten e incorporan herramientas tecnológicas para el funcionamiento de sus negocios. La adopción de las tecnologías en las pymes en Ecuador fue acelerada debido a la pandemia del Covid-19, ya que el trabajo presencial estaba restringido y tuvieron que buscar soluciones para las operaciones de sus negocios, como lo pueden ser sistemas y aplicaciones web, donde su información se almacena en la nube. Según el diario digital (Primicias, 2022) “en la actualidad nueve de cada 10 pymes en Ecuador está invirtiendo en su desarrollo tecnológico”.

Existen una gran variedad de sistemas web que se enfocan en optimizar las actividades comerciales de los negocios y cada uno de ellos se encuentran desarrollados utilizando diferentes tecnologías y arquitecturas según la problemática que deseen resolver. Una de las arquitecturas de desarrollo son los microservicios, en los cuales se busca independizar las funcionalidades del aplicativo y mejorar la escalabilidad del mismo.

En el trabajo de investigación realizado por Diego Alfonso Contreras, de la Universidad Distrital Francisco José Caldas de Colombia, a través de su trabajo “Modelado bajo una arquitectura de microservicios de un prototipo de software, para el comercio de tiendas de barrio”. El cual indica que gracias al auge que los microservicios han tenido en los últimos años de la misma forma han aumentado las soluciones y herramientas para su implementación. En base a ello en su trabajo pretende facilitar los servicios ofertados por parte de las tiendas de barrio a sus clientes, con un enfoque en microservicios. Señala como resultados que tener la independencia de cada uno de los servicios permite hacer su mantenimiento y desarrollo mucho más simple, pues el enfoque está puesto en esta tarea y en ese contexto en específico (Barrios D. , 2015).

De igual manera en el trabajo de titulación realizado por Juan Alberto Canqui Ticona de la Universidad Mayor de San Andrés, propone como una solución utilizar la arquitectura de microservicios para la facturación electrónica como una alternativa a la arquitectura de software

tradicional o monolítica. El objetivo principal de la investigación era desarrollar los microservicios necesarios para la facturación y hacer una comparación de la eficiencia en relación con una arquitectura monolítica. La conclusión fue que el uso de microservicios es mucho mejor en términos de tiempo y coste, permitiendo una mejor escalabilidad y un mayor soporte a las peticiones a las API 's correspondientes (Canqui, 2021).

En la investigación titulada “Aplicación web con arquitectura de microservicios y el incremento de la eficiencia en el comercio electrónico de una empresa peruana de calzado”, utilizando medidas estadísticas como la estadística inferencial a través de la prueba paramétrica t de Student para datos emparejados y la prueba no paramétrica Mann Whitney para datos no emparejados, con un nivel de significación del 5%. Los resultados obtenidos muestran que una aplicación web con arquitectura de microservicios puede aumentar su eficiencia, reduciendo los tiempos de procesamiento en una media del 85% (Ortiz, y otros, 2022).

En el trabajo de titulación de Luis Miguel Arango Chiza, de la universidad politécnica Salesiana, el trabajo denominado “Sistema de facturación electrónica para la web de la distribuidora Mateito”. En el cual presenta como resultados de la misma, la integración del módulo de facturación electrónica al sistema cumpliendo con las especificaciones del cliente para la generación de facturas electrónicas. Así mismo señala que trabajar con documentos electrónicos no solo implica la reducción de tiempo entre un proceso, sino que también mantiene la información segura y respaldada por el Servicio de Rentas Internas (SRI) (Anrango & Romero, 2019).

### **Justificación**

Actualmente la ferretería Electrocat realiza su actividad mercantil sin la utilización de un sistema que lleve el control de todas sus transacciones, y llevan estos registros de forma manual en documentos físicos o en softwares que no se encuentran especializados para dicha actividad, provocando varios problemas como los puede ser una planificación deficiente en las

actividades mercantiles, pérdida de información debido a la poca integridad de los datos que son registrados en documentos físicos, control impreciso de la entrada y salida de productos, limitación en las estadísticas de ventas, entre otros.

Cabe mencionar que también en el año 2022 debido a la reforma tributaria aprobada por la Ley de Desarrollo Económico del Estado Ecuatoriano, la facturación electrónica debe ser obligatoria a partir del 29 de noviembre del 2022 para todos quienes tienen un Registro Único de Contribuyentes (RUC) y tengan un ingreso bruto anual superior a USD 20.000. Solo los negocios populares pertenecientes al sistema Régimen Simplificado para Emprendedores y Negocios populares (RIMPE) aún tienen la posibilidad de emitir comprobantes de notas de venta de manera física. Pero, si en alguna transacción se requiera la emisión de una factura, ésta deberá ser de forma electrónica (Primicias, 2022).

Es por ello que se planteó la implementación de una aplicación web de administración y control empresarial, que logre cumplir con todos los requerimientos que solicita la ferretería Electrocat, como el control de ventas, facturación electrónica, registro de entrada y salida de mercadería, control de inventario, registro de proveedores, registro de clientes, control de deudas y contar con un sistema de roles de usuarios.

### **Alcance**

El sistema web de administración está enfocado principalmente a los negocios retail, para la automatización de los diferentes procesos del negocio. Basándose en la arquitectura de microservicios, desarrollar un sistema altamente escalable en el cumplimiento de las diferentes necesidades del negocio.

El sistema para la ferretería Electrocat contará con el manejo de tres roles, como administrador, gerente y empleado. Lo cual deberá permitir el control de los movimientos e información del negocio, con ello el usuario administrador podrá ingresar la información del negocio (datos generales, actividad económica, entre otros), para posteriormente realizar la administración de los productos, proveedores, clientes, compras y ventas.

De igual forma el sistema implementará facturación electrónica para la emisión de comprobantes de venta, cumpliendo los requisitos reglamentarios para la autorización por parte del Servicio de Rentas Internas (SRI). Una de las ideas principales es lograr una vinculación directa del gerente y el negocio, llevando un control de los movimientos del mismo. Se destaca también la visualización de los reportes de venta y varios procesos administrativos del sistema.

## **Objetivos**

### ***Objetivo General***

Desarrollar un sistema web de administración orientado a la optimización operacional de negocios retail para la ferretería Electrocat ubicada en la ciudad de Santo Domingo.

### ***Objetivos Específicos***

- Analizar los requerimientos funcionales y no funcionales necesarios para el desarrollo del sistema web, en función de las principales actividades que realizan los negocios retail.
- Efectuar el desarrollo del frontend utilizando frameworks que permiten la creación de aplicaciones web del lado del cliente.
- Realizar el desarrollo del backend utilizando la arquitectura de microservicios.
- Elaborar las pruebas necesarias del correcto funcionamiento del sistema web.
- Desplegar la solución tecnológica en un ambiente en la nube.

## **Capítulo II: Marco Teórico**

En el presente capítulo se presentan temas importantes para el desarrollo y comprensión del proyecto, es decir, se describen los conceptos técnicos que son tratados a lo largo de la documentación para un mejor entendimiento del mismo. Así mismo, se muestra la metodología de trabajo y las herramientas que se han utilizado, de esta manera justificando el uso de cada una de ellas en el desarrollo del proyecto.

### **Retail**

#### ***Definición***

Es un término inglés usado para definir la venta de productos al por menor o al detalle de persona a persona. Este es un sector de venta al detalle o minorista, cualquier tipo de producto que se venden por unidades al consumidor final, es por definición un producto minorista. Hoy por hoy los ejemplos más comunes de retail son supermercados, hipermercados, cadena de farmacias, tiendas de marca, cadena de tiendas de conveniencia y restaurantes, ferretería, hasta sucursales bancarias (Guerrero, 2012).

#### ***Funcionamiento Operacional de los Negocios Retail***

El comercio minorista o retail es el último eslabón de la cadena de distribución de productos y servicios. Su cargo es operar como intermediario entre los productores o mayoristas y los consumidores. En los últimos años, el desarrollo y crecimiento económico de esta actividad comercial la ha convertido en un elemento fundamental de cualquier economía, entre otras cosas por su gran impacto en los niveles de empleo y por ser un nuevo centro de poder por la concentración de compras que crea. Las grandes cadenas de distribución progresan agrupando cada vez más volumen de ventas. Los que se centran especialmente en el cliente ganan terreno y lo recuperan siempre adaptando sus estrategias a las necesidades cambiantes del mercado.

### ***Factores Operativos***

El funcionamiento de este tipo de negocio es similar al resto de empresas, la diferencia principal se basa en el modelo de gestión a utilizarse, ya que el éxito y funcionamiento de este tipo de negocios depende en gran medida de la aplicación de tres factores fundamentales según (Guerrero, 2012):

- **Gestión por Categorías (Category Management).** - Metodología de trabajo que trata con el proveedor y el minorista. Su objetivo se basa en definir categorías de productos que satisfagan las necesidades de ramos específicos y gestionar cada una de las líneas de productos, como unidades de negocios y centros de beneficio individuales.
- **Merchandising.** - Permite definir diferencias y ventajas frente a la competencia, siendo la forma en la que se organizan físicamente los productos, por lo que afecta directamente el comportamiento de compra de los clientes.
- **Marketing.** - Identifica la conducta de los clientes para comprender el proceso de elección entre varios puntos de venta, los de elección y de compra. Mediante estrategias, técnicas y prácticas atribuir una importancia mayor para un determinado público objetivo.

### **Descripción de la Metodología**

La metodología de trabajo en un proyecto es parte fundamental del mismo, ya que se basa en definir los pasos a seguir para la gestión del proyecto. En este caso la metodología seleccionada para llevar el proceso de trabajo es la Scrum, este marco de trabajo para el desarrollo ágil de software representa un punto de partida para la gestión del mismo, para ello primeramente es fundamental conocer sobre las metodologías ágiles.

### ***Metodología de trabajo ágil***

Buscan distribuir de forma indeleble sistemas de software en funcionamiento diseñados con iteraciones rápidas. Pretenden ofrecer un compromiso eficaz entre técnicas,



proporcionando los procesos únicamente necesarios para conseguir una recompensa razonable. Las metodologías ágiles son flexibles más que predictivas, ya que se acogen al cambio continuo y pretenden ser procesos que se aplican y progresan con el propio cambio, así también facilitar en poco tiempo pequeñas piezas de software en funcionamiento para elevar la satisfacción del cliente (Urteaga, 2015).

Al presente hay muchas metodologías ágiles en uso, pero para el desarrollo del presente proyecto es preciso hacer uso de esta metodología. Por lo tanto, es indispensable hablar de Scrum previamente a la implementación del mismo. A continuación, se procede a describir brevemente la metodología Scrum:

### **Scrum**

**Definición.** La metodología Scrum para el desarrollo ágil de software representa un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complicados, dentro del cual se consiguen utilizar varias técnicas y procesos. Scrum indica la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que se pueda mejorar (Schwaber & Sutherland, 2020).

**Scrum Team.** La unidad fundamental de Scrum es un grupo de personas, denominado Scrum Team, este equipo Scrum consta de diferentes roles que se asignan dentro del desarrollo del proyecto. Product Owner (propietario del producto), Scrum Master (especialista en Scrum) y Developers (desarrolladores), los Equipos Scrum son autoorganizados y multifuncionales de esta forma eligen la mejor forma de llevar a cabo su trabajo (Schwaber & Sutherland, 2020).

- **Product Owner:** El Dueño de Producto es el responsable de maximizar el valor del producto y del trabajo del equipo de desarrollo. La forma en que lleva esto podría variar considerablemente entre distintas organizaciones, equipos Scrum e individuos. El Product Owner, así mismo es el responsable de la gestión efectiva del Product Backlog,

en el cual se determinan los elementos de la lista del producto para cumplir los objetivos (Schwaber & Sutherland, 2020).

- **Scrum Master:** Es el responsable de establecer Scrum como se define en la Guía de Scrum. para ende ayuda a todos a comprender la teoría y la práctica de Scrum, tanto dentro del Scrum Team como de la organización. El Scrum Master es un líder que está al servicio del Equipo y brinda a las personas externas al equipo Scrum a entender qué interacciones con el mismo pueden ser de ayuda y cuáles no. El Scrum Master asiste a todos a cambiar estas interacciones para alcanzar el valor creado por el equipo de trabajo (Schwaber & Sutherland, 2020).
- **Developers:** El equipo de desarrollo se compromete a entregar un incremento de producto que se considere como “Finalizado”, según la distribución del trabajo en el desarrollo del proyecto, que potencialmente se pueda situar en producción, al final de cada Sprint. Los Equipos de Desarrollo son ordenados y empoderados por la organización para establecer y gestionar su propio trabajo. La reunión resultante optimiza la eficiencia y efectividad del equipo de desarrollo (Schwaber & Sutherland, 2020).

**Artefactos de Scrum.** Para la implementación del marco de trabajo Scrum, es fundamental conocer los siguientes artefactos (Schwaber & Sutherland, 2020):

- **Sprint:** Eventos que por lo general tienen una duración de un mes o menos, los cuales se realizan de forma secuencial permitiendo garantizar la inspección y adaptación del progreso hacia el objetivo planteado. Su objetivo y alcance se basa en el Product Backlog.
- **Product Backlog:** Se trata de una lista ordenada con los requerimientos respectivos para mejorar el producto. Esta fuente de trabajo es realizada por el Scrum Team,

por otra parte, el Product Owner se encarga de dar un orden de prioridad a las tareas que serán ejecutadas.

- **Sprint Backlog:** Es una lista estructurada de tareas identificadas por el Scrum Team que se realiza durante el tiempo establecido por cada sprints. Se establecen los tiempos y se asignan las responsabilidades entre el equipo, con ello brindar un incremento visible y susceptible al producto final.
- **Scrum Daily:** Es un evento o reunión para presentar los resultados de la realización de cada sprint planteado. En estas reuniones se puede realizar ajustes o adaptar el sprint Backlog según sea necesario, los developers pueden hacer uso de la estructura y técnicas que deseen, haciendo que su Daily Scrum se centre en el progreso del objetivo planteado.

Su finalidad es mejorar la comunicación, agilizar la elección de decisiones y en base a ello solventar dudas o cambios en las tareas realizadas y evitar reuniones posteriores.

## **Descripción de las Herramientas de Desarrollo**

### ***Tecnologías Web***

**Node Js.** Según Puciarelli (2020), define Node Js como “un entorno de ejecución multiplataforma para el lenguaje de programación JavaScript” (pág. 2). Plataforma del lado del servidor que permite crear fácilmente aplicaciones de red rápida y escalables, de la misma forma fundamental para aplicaciones de tiempo real y uso recurrente de datos.

**Angular.** Según Castro (2018), define Angular como “es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página” (pág. 10). Las diferentes bibliotecas integradas presentan una amplia variedad de funciones como, el enrutamiento, la gestión de formularios, la comunicación cliente-servidor y también basarse en el modelo vista controlador (MVC).

Se debe puntualizar que la elección de Angular sobre otras tecnologías para el desarrollo del frontend se dio debido a la familiarización que se tiene con el mismo, ya que en el transcurso de la formación académica se utilizó Angular como principal marco de trabajo en el desarrollo de aplicaciones web, a continuación, en la tabla 1 se indica una comparativa entre Angular, React y Vue.

**Tabla 1**

*Angular Vs React Vs Vue*

<b>Categoría</b>	<b>Angular</b>	<b>React</b>	<b>Vue</b>
Tipo	Un Framework	Una rica librería para construir la interfaz de usuario	Una librería
Enfoque de desarrollo	Basado en TypeScript	Todo en JavaScript	Basado en JavaScript y HTML
Mantenido y apoyado por	Google	Facebook	Antiguos empleados de Google
Ideal para	Angular es un enfoque ideal para crear aplicaciones a gran escala y con muchas funciones o aplicaciones de nivel empresarial	Adecuado para el desarrollo moderno de la web y las aplicaciones nativas para iOS y Android	Ideal para el desarrollo web y las aplicaciones de una sola página
Lenguaje de preferencia	Recomienda el uso de TypeScript	Recomienda el uso de JSX - JavaScript XML	HTML templates y JavaScript
Modelo	Basado en MVC (arquitectura Modelo-Vista-Controlador)	Basado en Virtual DOM (Document Object Model)	Basado en Virtual DOM (Document Object Model)

*Nota.* Comparación de tecnologías de desarrollo Frontend. Obtenido de

<https://plainenglish.io/blog/angular-vs-react-vs-vue-js-which-is-the-best-choice-for-2022>.

En base a la comparativa anterior se puede observar que Angular utiliza como lenguaje principal TypeScript, el cual puede ser considerado como una mejora de JavaScript al tener un tipado fuerte, ya que se puede prevenir de mejor forma el manejo de errores por los diferentes tipos de datos, adicionalmente mencionar que es ideal para soluciones de gran escalado o aplicaciones de nivel empresarial, por lo que Angular se presenta como la mejor opción a elegir entre las tecnologías cotejadas.

**Microservicios.** Son una arquitectura de desarrollo de aplicaciones software, que permiten dividir una aplicación grande en partes independientes más pequeñas y que cada una de ellas tenga su propio rol o compromiso asignado. Una aplicación que consiste en microservicios puede llamar a muchos microservicios internos para redactar su respuesta con la finalidad de entregar una única solicitud de usuario (Barrios D. A., 2018).

En una arquitectura de microservicios se pueden hallar varias ventajas a comparación de una arquitectura tradicional o monolítica, esta comparación se la puede observar en la tabla 2.

**Tabla 2**

*Arquitecturas monolíticas vs. Microservicios*

<b>Categoría</b>	<b>Arquitectura Monolítica</b>	<b>Arquitectura de microservicios</b>
Código	Una base de código única para toda la aplicación.	Múltiples bases de código. Cada microservicio tiene su propia base de código.
Comprensibilidad	A menudo confuso y difícil de mantener	Mayor facilidad de lectura y mucho más fácil de mantener

Despliegue	Implementaciones complejas con ventanas de mantenimiento y paradas programadas.	Despliegue sencillo ya que cada microservicio se puede implementar de forma individual, con un tiempo de inactividad mínima, si no es cero
Lenguaje	Típicamente totalmente desarrollado en un lenguaje de programación.	Cada microservicio puede desarrollarse en un lenguaje de programación diferente
Escalamiento	Requiere escalar la aplicación entera, aunque los cuellos de botella estén localizados.	Cada microservicio puede ser Escalado de forma independiente

---

*Nota.* La arquitectura de microservicios se comporta de mejor manera ante sistemas de alta concurrencia y gran escalabilidad. Obtenido de “Microservices from Theory to Practice IBM Corporation”, por Daya, S., Van Duy, N., Eati, K., Ferreira, C., Glozic D., Gucer, V., Vennam R, 2016.

En base a la información de la tabla anterior se puede observar que en la arquitectura de microservicios existe la flexibilidad de poder utilizar diferentes lenguajes de programación dependiendo del escenario que se presente, igualmente es una mejor solución para aplicaciones de gran escalabilidad, ya que cada microservicio puede desplegarse de forma independiente y realizar los cambios de forma específica sin necesidad de volver a desplegar toda la aplicación.

**Middleware.** Software de conectividad que consiste en un conjunto de servicios que permiten interactuar a múltiples procesos que se ejecutan en distintas máquinas a través de una red. Ocultan la heterogeneidad, abstraen la complejidad subyacente y proveen de un modelo de programación conveniente para los desarrolladores de aplicaciones. Brinda una

interfaz de programación de aplicaciones (API) estándar para la gestión de la entrada y la salida de los datos requeridos desde el componente (García. F, 2005).

**Java.** Es un lenguaje de programación orientado a objetos. Ha evolucionado desde sus inicios hasta promover una gran parte del mundo digital actual, ya que es una plataforma fiable en la que se crean varios servicios y aplicaciones. Como lo menciona Rojas & Silva (2022), “Java no solo es un mensaje de programación, sino también una plataforma que permite el desarrollo y la ejecución de programas en diferentes sistemas y equipos” (pág. 21)

**Spring Boot.** Es un framework desarrollado para trabajo con Java basado en el Modelo Vista Controlador, como también un entorno de desarrollo de código abierto y gratuito. Este proporciona la creación de aplicaciones establecidas en Spring independientes y de grado de producción que puede "simplemente ejecutar". Las diferentes librerías que brinda hacen fácil el desarrollo y despliegue de los servicios REST (Haro & Guarda, 2018).

**Jira Software.** es un sistema de seguimiento de problemas desarrollado por Atlassian Corporation a partir de 2002, permite la gestión de trabajo para equipos de software que deben organizar y hacer un seguimiento del trabajo. Forma parte de una gama de productos diseñados para ayudar a equipos de todo tipo a gestionar el trabajo. Se ha convertido en una potente herramienta de gestión de trabajo para todo tipo de casos de uso, desde la gestión de requisitos y casos de prueba hasta el desarrollo de software ágil (Fisher, 2013).

**Amazon Web Services.** Es un proveedor de servicios en la nube, ofrece más de 200 servicios integrales de centros de datos a nivel global. Brinda recursos como almacenamiento, recursos de computación, bases de datos entre otros, en la particularidad de cloud computing. Presenta algunas ventajas para sus clientes como seguridad, bases de datos, costos bajos, accesibilidad y entre otros aspectos que se adaptan según a las diferentes necesidades planteadas (Pozo, 2021).

**Linode.** Esta es solución de alojamiento en la nube para ayudar a los desarrolladores de software a gestionar su infraestructura de computación en la nube y sus servidores privados virtuales en una plataforma unificada, soluciones en la nube fáciles de usar y servidores privados virtuales (VPS) (Linode, 2022).

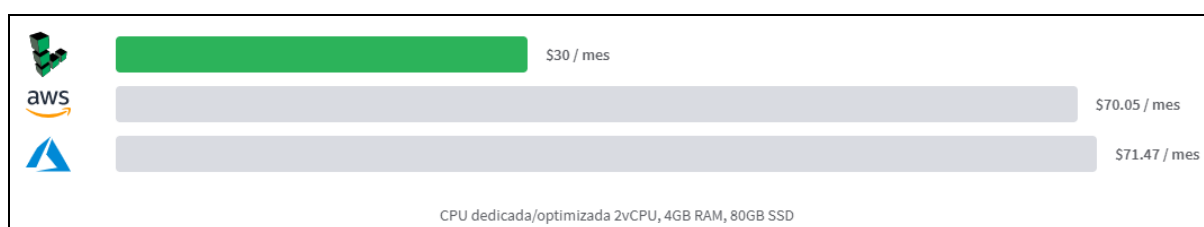
Entre las principales opciones que presenta Linode son:

- Alojamiento compartido
- Alojamiento dedicado
- Alojamiento VPS

La elección de Linode como proveedor de servicios de computación en la nube se realiza en base a una serie de características que ofrece a comparación de otros proveedores como lo pueden ser AWS, Azure, Google Cloud, entre otros. Como primera ventaja que presenta Linode es una mejor relación calidad-precio como se puede observar en la Figura 1.

### Figura 1

*Relación de precios entre Linode, AWS y Azure.*



*Nota.* Comparación de precios de diversos proveedores utilizando la misma infraestructura.

Tomada de *La mejor relación calidad-precio* [Fotografía], Linode, <https://www.linode.com/es/>.

Otra ventaja que Linode ofrece es la Asistencia humana 100% gratuita, además de brindar herramientas de seguridad como los demás proveedores de servicios de computación en la nube, como se observa en la comparativa de la tabla 3.



**Tabla 3**

*Tabla comparativa de Linode vs AWS vs Azure y Google Cloud.*

<b>Características</b>	<b>Linode</b>	<b>AWS</b>	<b>Azure</b>	<b>Google Cloud</b>
GPUs en la nube	Si	Si	Si	Si
Cortafuegos en la nube gratuito	Si	Si	Si	Si
Protección DDoS gratuita	Si	Si	Si	Si
Servidores en la nube de metal desnudo	Si	Si	Si	Si
Kubernetes gestionados	Si	Si	Si	Si
SSD local	Si	Si	Si	Si
VMs de alta memoria	Si	Si	Si	Si
Mercado de aplicaciones	Si	Si	Si	Si
Asistencia humana 100% gratuita	Si	No	No	No
Biblioteca de más de 1.000 documentos	Si	Si	Si	Si
Sencillo API, CLI, y una interfaz en la nube fácil de usar	Si	No	No	No
Precios planos predecibles	Si	No	No	No
Transferencia gratuita de paquetes	Si	No	No	No
El mismo precio en todos los centros de datos	Si	No	No	No

Nota. Métricas de comparación propuestas por el sitio web de Linode. Obtenido de <https://www.linode.com/es/compare/>.

### ***Herramientas de Calidad de Código***

**SonarQube.** Es una plataforma open source para el análisis y evaluación continua de la calidad del código por medio de diferentes herramientas de análisis estático de código fuente. Brinda algunos parámetros que ayuden a mejorar la calidad del código, permitiendo de cierta forma a los desarrolladores detectar los errores y corregir los mismos (Sentry, 2021).

La calidad de software forma parte fundamental en lo que se refiere al desarrollo, tener a disposición diferentes herramientas que brinden esa ayuda de evaluar y analizar el código es algo esencial para garantizar la calidad del código fuente y la adopción de buenas prácticas.

**Apache JMeter.** Es un software de código abierto para cargar, probar el comportamiento funcional y pruebas de rendimiento. Además, permite simular las cargas de usuario para probar los sistemas web en la utilización de usuarios reales. Inicialmente fue creado para probar aplicaciones web, pero en la actualidad se ha extendido a varias funciones de prueba (Memon & Tahseen, 2018).

### ***Herramientas de Contenedorización***

**Docker.** Según Muñoz (2017), define Docker “un proyecto open source que permite crear aplicaciones en contenedores de software que son ligeros, portátiles y autosuficientes” (pág. 16). Esta tecnología proporciona una capa adicional de abstracción y automatización en lo que se refiere a la virtualización, uno de los factores fundamentales de Docker es separar la infraestructura de las aplicaciones de forma que se pueda brindar el software de manera rápida.

Una de las principales diferencias de Docker con los modelos comunes de virtualización precisamente es la utilización de contenedores en vez de las máquinas virtuales. Para conocer de mejor forma el funcionamiento de Docker es fundamental saber de qué se compone, las cuales son:

- Contenedores
- Imágenes
- Docker Daemon
- Docker Client CLI
- Docker Hub

Según indica Muñoz (2017), “los contenedores son un paquete de elementos que te permite crear un entorno donde correr aplicaciones independientemente del sistema operativo” (pág. 15). Por otro lado, una imagen Docker es un paquete independiente y ejecutable que contiene lo fundamental para ejecutar una aplicación como código, herramientas y librerías.

Docker sigue una serie de pasos para ejecutar un contenedor, primeramente, el Docker Client CLI (Interfaz de línea de comando) conecta con el Docker Daemon, el cual se encarga de descargar la imagen del repositorio correspondiente, denominado Docker Hub y este despliega un contenedor a partir de esa imagen. Según Silva (2021) señala que “Durante la ejecución, una imagen se convierte en un contenedor que se ejecuta sobre la máquina Docker (Docker Engine), la cual se encuentra disponible en entornos Linux y Windows” (pág. 6).

**Kubernetes.** Es un orquestador de código abierto para administrar cargas de trabajo y servicios, este proporciona la automatización y la configuración declarativa. Esta tecnología implementa sistemas escalables y confiables en contenedores por medio de API dirigidas a aplicaciones (Ramos & Recalde, 2022).

Kubernetes presenta varias características y se lo puede implementar como:

- Una plataforma de contenedores
- Una plataforma de microservicios
- Una plataforma portable de nube

Esta tecnología orquesta infraestructura de cómputo, redes y almacenamiento. También brinda una simplicidad en las Plataformas como Servicio (PaaS) con la infraestructura como servicio (IaaS) que permite la portabilidad entre proveedores de infraestructura.

**Servicios RESTful.** Para definir los servicios RESTful, se debe iniciar conociendo su raíz, es decir, los servicios web, con el tiempo surgieron estándares comenzando por Simple Object Access Protocol (SOAP) y Web Services Description Language (WSDL) que formó de cierta forma confusión en los desarrolladores y con ello facilitando el surgimiento de los servicios RESTful, siendo más fáciles en su implementación (Adamczyk & Smith, 2011).

Los servicios REST, que el término Representational State Transfer (REST) fue dicho por Roy Fielding para identificar un estilo de arquitectura dirigida al diseño de software basado en la red. Posteriormente el término se expandió para identificar un estilo de conformación de servicios web, en base a los principios establecidos por Fielding, lo cual se ha denominado RESTful (Adamczyk & Smith, 2011).

En la práctica se espera que los servicios web RESTful sigan el modelo de operaciones CRUD, donde se definen los métodos para crear, leer, actualizar y eliminar un recurso en el servidor. De la misma manera apoyado por medio del protocolo Hypertext Transfer Protocol (HTTP) y un Identificador de Recursos Uniforme (URI) (Adamczyk & Smith, 2011).

(Rodríguez, 2008), indica que la implementación concreta de un servicio web REST sigue cuatro principios básicos de diseño como:

- Usar métodos HTTP explícitamente
- No tener estado
- Exponer URI similares a estructuras de directorio
- Para la transferencia de información usar XML y JSON, o ambos.

Los métodos del protocolo HTTP determinan las acciones a realizar en los recursos. Los más conocidos y usados son POST, GET, PUT y DELETE. Según (Rodríguez, 2008) define “en un servicio web RESTful el servidor es el único encargado de generar los resultados a las peticiones realizadas por el cliente para mantener la comunicación en ambas partes”.

### ***Base de datos***

Las bases de datos y sus tecnologías han tenido un gran impacto en el uso de las computadoras, ya que juegan un papel fundamental en las diferentes áreas de funcionamiento. Según Márques (2009) “Una base de datos es un conjunto de datos almacenados en memoria externa que están organizados mediante una estructura de datos” (pág. 2). Cada base de datos

está creada para cumplir con las diferentes necesidades referentes al funcionamiento de empresa u organización.

En relación al concepto de base de datos se encuentra el Sistema Gestor de Bases de Datos (SGDB o por sus siglas en inglés DBMS), este es una colección de programas que permite al usuario la manipulación de una base de datos. Así como lo menciona Marqués (2009) “es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, además de proporcionar un acceso controlado a la misma” (pág. 3). Uno de los puntos principales de SGBD es brindar una manera de recopilar y rescatar la información de una base de datos de forma que sea práctica como eficiente.

En la actualidad hay una gran variedad de base de datos, estas pueden ser diseñadas de acuerdo a su objetivo o propósito, como también la forma de funcionamiento del negocio y en base a eso llevar una adecuada manipulación de los datos. Para realización de este proyecto se empleó una base de datos de tipo relacional ya que se adapta a las diferentes necesidades del negocio, para ello se procede a conocer el funcionamiento de la misma.

**Relacionales.** Las bases de datos relacionales se basan en el concepto matemático de relación que de forma gráfica se representa mediante una tabla. Codd utilizó esta terminología matemática para referirse a este modelo relacional. Así como lo menciona Gracia & Yanes (2013) “es una base de datos en donde todos los datos visibles al usuario están organizados estrictamente como tablas de valores, y en donde todas las operaciones de la base de datos operan sobre estas tablas” (pág. 23). Estas bases de datos se presentan a los usuarios como una colección de relaciones normalizadas de diverso grado, que evolucionan con el tiempo.

Actualmente existen varias bases de datos relacionales, entre las más conocidas y usadas dentro de este modelo están, PostgreSQL, MySQL, MariaDB, Oracle Database y

Microsoft SQL Server. Para ello se conoce el concepto de este tipo de base de datos, ya que este modelo fue seleccionado para la implementación del sistema.

**PostgreSql.** Es un gestor de bases de datos orientadas a objetos compatible con los servicios de OVHcloud y las herramientas más comunes del mercado. Admite varios modelos de datos para crear aplicaciones orientadas a objetos potentes y escalables. Sistema de base de datos relacional de objetos de código abierto con más de 35 años de desarrollo activo que le ha valido una sólida reputación por su confiabilidad, robustez de funciones y rendimiento (Gibert & Pérez, 2012).

A continuación, se hace una breve comparativa entre PostgreSQL y MySQL, observando sus principales características, como se muestra en la tabla 4.

**Tabla 4**

*Tabla comparativa de MySQL, PostgreSQL, Microsoft SQL Server y Oracle Database.*

<b>Criterio</b>	<b>MYSQL</b>	<b>POSTGRESQL</b>	<b>Microsoft SQL Server</b>	<b>Oracle Database</b>
Velocidad	MySQL es mucho más rápido que la mayor parte de su competencia ya que tiene la posibilidad de selección de mecanismos de almacenamiento que ofrecen diferentes velocidades de operación, soporte físico, capacidad,	PostgreSQL parece la mejor opción en el caso de proyectos grandes y complejos con gran volumen de operaciones de datos, que requieren alta velocidad, fiabilidad y escalabilidad tanto en las lecturas como en	Es una base de datos desarrollada por Microsoft y está optimizada para el sistema operativo Windows. En general, tiene un buen rendimiento y es capaz de manejar grandes volúmenes de datos. Sin embargo, puede ser menos eficiente en comparación	Es una base de datos popular para aplicaciones empresariales y proyectos de gran tamaño. Tiene una alta capacidad de manejar grandes volúmenes de datos y una alta disponibilidad. Sin embargo, su complejidad puede resultar en un menor rendimiento

	distribución geográfica, transacciones. Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.	las escrituras de datos.	con otras opciones en consultas complejas.	en comparación con otras opciones en aplicaciones más pequeñas.
Modelo Conceptual	Modelo relacional	Modelo relacional Orientado a Objetos	Modelo relacional	Objeto-relacional
Fiabilidad	MySQL resulta fácil de utilizar y administrar. Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso.	Sus características técnicas la hacen una de las bases de datos más potentes.	Cuenta con una función de alta disponibilidad denominada "Always On", que permite la replicación de bases de datos y la conmutación por error automática.	Replicación de bases de datos, la copia de seguridad y la recuperación basada en registro
Tamaño en registros	MySQL tiene bases de datos de hasta 50 millones de registros.	Ilimitado (Depende del sistema de almacenamiento).	Ilimitado (Depende del sistema de almacenamiento).	Ilimitado (Depende del sistema de almacenamiento).
Respaldo de Bases de Datos	MySQL cuenta con comandos	PostgreSQL cuenta con comandos específicos para la	Cuenta con varias herramientas para el respaldo de	Cuenta con una herramienta de respaldo llamada

específicos para copia de respaldo y bases de datos, "RMAN" (Recovery copia de respaldo también la incluyendo la Manager), que y también restauración de la utilidad de línea de permite crear restaurar la base base de datos si comandos "sqlcmd" copias de de datos si esta esta llegase a y la herramienta de seguridad de las llegase a perderse. administración bases de datos perderse, aunque "SQL Server no siempre sea la Management mejor opción ya Studio" que ha llegado a fallar en pocas ocasiones

Licencia	Base de datos de código abierto y se distribuye bajo la licencia GPL (General Public License) de GNU	Base de datos de código abierto y se distribuye bajo la licencia PostgreSQL, que es similar a la licencia MIT	Base de datos propiedad de Microsoft y se distribuye bajo diferentes licencias según la edición	Base de datos propiedad de Oracle Corporation y se distribuye bajo diferentes licencias según la edición
----------	--	---	---	--

---

*Nota.* MySQL es una muy buena alternativa para páginas o aplicaciones web, pero si se trata de entornos de alta concurrencia una mejor opción es PostgreSQL. Información adicional agregada de las bases de datos SQL Server y Oracle Database. Obtenido de "Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL", por López Herrera, P., 2016.

La elección de PostgreSQL como motor de base de datos sobre otras opciones es debido a que es uno de los motores que mejor garantiza un soporte eficiente para la concurrencia, además de estar dirigido por una comunidad de código abierto (open source), el PGDG (PostgreSQL Global Development Group) y no depende de ningún proveedor, también mencionar que PostgreSQL es el motor que más asegura el cumplimiento de los principios ACID.



## Capítulo III: Metodología

### Análisis y requerimientos

El sistema web a desarrollar se plantea como objetivo desarrollar un sistema web de administración orientado a la optimización operacional de negocios retail para la ferretería Electrocat ubicada en la ciudad de Santo Domingo. Para ello, y según las necesidades identificadas, se procede a establecer la información correspondiente para los requerimientos funcionales, no funcionales, de sistema y exploratorios fundamentales para la implementación y desarrollo del proyecto.

El proceso de elicitación de requerimientos es fundamental para abstraer una comprensión del problema que se quiere resolver con el producto software, en el cual se realizó investigaciones sobre el funcionamiento de los negocios retail y análisis de la estructura administrativa y operacional. Como resultado se obtuvieron los siguientes requerimientos.

### **Requerimientos Funcionales**

En la Tabla 5 se describen todos los requerimientos funcionales necesarios para el desarrollo del sistema web.

**Tabla 5**

#### *Requerimientos Funcionales*

<b>ID</b>	<b>Descripción</b>
RF1	El sistema web debe tener un control de login para usuario - Administrador
RF2	El sistema web debe permitir crear diferentes cuentas de empleados - Administrador
RF3	El sistema web debe permitir recuperar la contraseña de ingreso del usuario – Administrador
RF4	El sistema web debe permitir cambiar la contraseña de ingreso del usuario – Administrador
RF5	El sistema web debe tener un control de login para usuario - Empleado
RF6	El sistema web debe permitir recuperar la contraseña de ingreso del usuario – Empleado

- RF7 El sistema web debe tener un control de inventario - Administrador
- RF8 El sistema web debe realizar el proceso de facturación electrónica - Administrador
- RF9 El sistema web debe tener un control de ventas del negocio - Administrador
- RF10 El sistema web debe permitir un registro de proveedores - Administrador
- RF11 El sistema web debe permitir un registro de clientes - Administrador
- RF12 El sistema web debe permitir llevar un control de deudas - Administrador
- RF13 El sistema web debe tener un control de inventario - Empleado
- RF14 El sistema web debe realizar el proceso de facturación electrónica - Empleado
- RF15 El sistema web debe tener un control de ventas del negocio - Empleado
- RF16 El sistema web debe permitir un registro de proveedores - Empleado
- RF17 El sistema web debe permitir un registro de clientes - Empleado

---

*Nota.* Esta tabla presenta los requerimientos funcionales. Fuente: Elaboración Propia

### **Requerimientos No Funcionales**

En la Tabla 6 Se describen todos los requerimientos no funcionales necesarios para el desarrollo del sistema web.

**Tabla 6**

#### *Requerimientos No Funcionales*

<b>ID</b>	<b>Tipo</b>	<b>Descripción</b>
RNF1	Confiabilidad	El usuario no puede experimentar más de cinco fallas por mes en la aplicación.
RNF2		Solo el usuario administrador tendrá permisos para realizar modificaciones a los registros, como eliminar, actualizar, entre otros
RNF3	Usabilidad	El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 5 horas.
RNF4		El sistema debe proporcionar mensajes de error que sean informativos y orientados al usuario final.
RNF5		El sistema debe poseer interfaces gráficas bien formadas.

RNF6		La aplicación web debe funcionar en Firefox, Chrome, Opera, entre otros.
RNF7	Disponibilidad	La aplicación web debe estar disponible 24/7.
RNF9		El tiempo para iniciar o reiniciar el sistema no podrá ser mayor a 5 minutos
RNF10		El promedio de duración de fallas no podrá ser mayor a 5 horas
RNF11		La probabilidad de falla del Sistema no podrá ser mayor a 0,5.
RNF12	Eficiencia	Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos.
RNF13		Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 5 segundos.
RNF14		La aplicación de recuperar la información del usuario y mostrarla en menos de 3 segundos.
RNF15	Seguridad	Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador
RNF16		La longitud de las contraseñas del sistema debe ser de mínimo 8 caracteres debe incluir símbolos, al menos una mayúscula y al menos un número.
RNF17		El sistema debe desarrollarse aplicando patrones y recomendaciones de programación que incrementen la seguridad de datos.
RNF18		Las contraseñas en la base de datos deberán estar encriptadas

---

Nota. Esta tabla presenta los requerimientos no funcionales. Fuente: Elaboración propia.

### ***Requerimientos del sistema***

En la Tabla 7 se describen todos los requerimientos del sistema necesarios para el desarrollo del sistema web.

**Tabla 7***Requerimientos del Sistema*

<b>ID</b>	<b>Descripción</b>
RS1	El sistema web puede utilizarse sin necesidad de instalar ningún software adicional además de un navegador web.
RS2	El sistema web podrá ser utilizado en los sistemas operativos Windows, Linux y OSX.
RS3	El sistema web debe poder ser utilizado en diferentes navegadores, como: Google Chrome, Microsoft Edge, Mozilla Firefox

Nota. Esta tabla presenta los requerimientos del sistema. Fuente: Elaboración Propia.

**Definición y Roles de proyecto**

Para establecer los roles del proyecto se utilizó la metodología SCRUM. Los roles y participantes para el presente proyecto son presentados en la Tabla 8.

**Tabla 8***Definición de Roles*

	<b>Roles</b>
<b>Product Owner</b>	Antony Mantuano
<b>SCRUM Master</b>	Luis Manuel Chica
<b>Development Team</b>	Christian Pérez Eddy Castillo

Nota. Esta tabla presenta los roles del proyecto. Fuente: Elaboración Propia.

Scrum Master: Es responsable de gestionar los procesos y herramientas del proyecto, trabajando con cada miembro del equipo para dirigir y entrenar al equipo a través del marco Scrum, así como para supervisar continuamente el flujo de trabajo con respecto a los plazos de entrega.

Product Owner: Se centrará en la planificación de las actividades para el desarrollo del producto, en el cual elaborará una lista de características necesarias para el sistema. También

será responsable de la organización y planificación de cada sprint, con el fin de organizar y planificar cada sprint y, en consecuencia, realizar la verificación correspondiente al final de la verificación al final de la ejecución.

Developers: Su principal objetivo es desarrollar la aplicación sobre la base de los diferentes sprints definidos en el marco Scrum. Y entregar un producto funcional al final de cada sprint hasta el final del desarrollo.

### Planificación del product backlog

Una de las ventajas del marco Scrum es que permite planificar el backlog del producto, enumerando y priorizando las tareas dentro de un plazo estimado, y situando el trabajo restante en la lista de tareas del equipo de desarrollo.

Se ha estimado un tiempo de noventa y nueve días para el desarrollo del proyecto, los cuales son previstos por el equipo para llevar las tareas correspondientes, como también revisar y entregar el producto basado en los requerimientos presentados en el marco Scrum. La tabla 9 indica a detalle la planificación del Product Backlog.

**Tabla 9**

#### *Product Backlog*

ID	Descripción	Prioridad
RF1	Login usuario Administrador/Gerente	Alta
RF2	Crear cuentas de empleados – Administrador/Gerente	Alta
RF3	Sistema para recuperar la contraseña de ingreso – Administrador/Gerente	Alta
RF4	Sistema para cambiar la contraseña de ingreso – Administrador/Gerente	Media
RF5	Login usuario Empleado	Alta
RF6	Sistema para recuperar la contraseña de ingreso - Empleado	Media
RF7	Control de inventario – Administrador/Gerente	Alta
RF8	Sistema de facturación – Administrador/Gerente	Alta

RF9	Sistema de control de ventas – Administrador/Gerente	Alta
RF10	Registro de Proveedores – Administrador/Gerente	Media
RF11	Registro de clientes – Administrador/Gerente	Alta
RF12	Control de deudas – Administrador/Gerente	Media
RF13	Control de inventario – Empleado	Media
RF14	Sistema de facturación – Empleado	Alta
RF15	Sistema de control de ventas - Empleado	Alta
RF16	Registro de Proveedores – Empleado	Media
RF17	Registro de clientes – Empleado	Alta

*Nota.* Esta tabla presenta la planificación del producto backlog. Fuente: Elaboración Propia.

### Definición de los Sprints

En el desarrollo ágil, un sprint es un período de tiempo determinado durante el cual completa una actividad específica y se prepara para su revisión. Para ello se define las tareas a realizar con la prioridad y la estimación del tiempo a desarrollar de cada sprint. En la parte de la solución tecnológica de este proyecto se desarrolla con una arquitectura de microservicios, para ello se definieron cinco sprints por cada microservicio, los mismos que tendrán una duración similar y el tiempo máximo no superara las cuatro semanas. Por cada sprint se estimó un tiempo de 6 horas diarias por cada día hábil, estos serán cumplidos por el equipo de desarrollo que se conforma por dos personas. En las tablas 10, 11, 12, 13 y 14 se indican las definiciones de los sprints correspondientes.

#### Tabla 10

*Definición Sprint nº. 1*

Sprint 1			
Duración del Sprint	20		
Días de trabajo	20		
Miembro del equipo	Días hábiles durante el Sprint	Horas hábiles por día	Horas hábiles por Sprint

Christian Pérez	20	6	120
Eddy Castillo	20	6	120
<b>Total</b>			<b>240</b>

*Nota.* Esta tabla describe el Sprint 1 en términos de tiempo y ejecución. Fuente: Elaboración Propia.

**Tabla 11**

*Definición Sprint n.º. 2*

<b>Sprint 2</b>			
Duración del Sprint	14		
Días de trabajo	14		
Miembro del equipo	Días hábiles durante el Sprint	Horas hábiles por día	Horas hábiles por Sprint
Christian Pérez	14	6	84
Eddy Castillo	14	6	84
<b>Total</b>			<b>168</b>

*Nota.* Esta tabla describe el Sprint 2 en términos de tiempo y ejecución. Fuente: Elaboración Propia.

**Tabla 12**

*Definición Sprint n.º. 3*

<b>Sprint 3</b>			
Duración del Sprint	21		
Días de trabajo	21		
Miembro del equipo	Días hábiles durante el Sprint	Horas hábiles por día	Horas hábiles por Sprint
Christian Perez	21	6	126
Eddy Castillo	21	6	126

<b>Total</b>	252
--------------	-----

*Nota.* Esta tabla describe el Sprint 3 en términos de tiempo y ejecución. Fuente: Elaboración Propia.

**Tabla 13**

*Definición Sprint n°. 4*

<b>Sprint 4</b>			
Duración del Sprint	20		
Días de trabajo	20		
<b>Miembro del equipo</b>	<b>Días hábiles durante el Sprint</b>	<b>Horas hábiles por día</b>	<b>Horas hábiles por Sprint</b>
Christian Perez	20	6	120
Eddy Castillo	20	6	120
<b>Total</b>			240

*Nota.* Esta tabla describe el Sprint 4 en términos de tiempo y ejecución. Fuente: Elaboración Propia.

**Tabla 14**

*Definición Sprint n°. 5*

<b>Sprint 5</b>			
Duración del Sprint	15		
Días de trabajo	15		
<b>Miembro del equipo</b>	<b>Días hábiles durante el Sprint</b>	<b>Horas hábiles por día</b>	<b>Horas hábiles por Sprint</b>
Christian Perez	15	6	90
Eddy Castillo	15	6	90
<b>Total</b>			180



*Nota.* Esta tabla describe el Sprint 5 en términos de tiempo y ejecución. Fuente: Elaboración Propia.

### **Planificación de los Sprints**

Previamente establecido el tiempo de trabajo para cada uno de los integrantes del equipo de desarrollo, se procede a plantear las tareas a realizar en cada uno de los sprints presentados en base a los requerimientos y las prioridades correspondientes.

#### ***Objetivo de sprint 1***

Primeramente, se establece el objetivo o meta del primer sprint que se puede lograr mediante la implementación de las diferentes tareas asignadas, en base a esto facilitar la toma de decisiones del equipo de desarrollo. La meta de este sprint se basa en el funcionamiento del primer microservicio de autenticación del sistema web. En la tabla 15 se indica la definición del objetivo del primer sprint.

**Tabla 15**

#### *Sprint Goal 1*

<b>Product Goal - Sprint 1</b>	Funcionamiento del micro servicio de autenticación e implementación en la interfaz gráfica.
Sprint Goal 1	Diseño de la interfaz gráfica para la implementación de los servicios del microservicio
Sprint Goal 2	Correcto funcionamiento del sistema de Login y registro de usuarios
Sprint Goal 3	Correcto funcionamiento del módulo de control y administración de deudas.

*Nota.* Esta tabla describe el objetivo del sprint 1. Fuente: Elaboración Propia.

#### ***Sprint 1***

La planificación del primer sprint engloba las tareas necesarias para el diseño y creación del login y autenticación del sistema, roles de usuario, cuentas del personal y control de deudas en lo que respecta al primer módulo. Esta planificación se presenta en la tabla 16.

**Tabla 16***Planificación Detallada del Sprint no. 1*

<b>Sprint N: 1</b>			
<b>Fecha de Inicio: 06/10/2022</b>			
<b>Fecha de finalización: 26/10/2022</b>			
<b>Código</b>	<b>Descripción</b>	<b>Tareas</b>	<b>Valor de Estimación (Días)</b>
RF01	Login usuario Administrador	<ul style="list-style-type: none"> <li>- Diseño de la base de datos</li> <li>- Diseño y creación del controlador login</li> <li>- Diseño y creación de los servicios que utilizará el controlador login</li> <li>- Diseño y creación de la ruta del api login</li> <li>- Diseño y creación de la interfaz de login en el frontend</li> <li>- Pruebas de interfaz y servicios.</li> </ul>	6
RF05	Login usuario Empleado	<ul style="list-style-type: none"> <li>- Implementación de roles de usuario en el controlador de login y sus servicios.</li> <li>- Envío y almacenamiento de los datos del perfil del usuario</li> <li>- Correcciones y Pruebas.</li> </ul>	1
RF03	Sistema para recuperar la contraseña de ingreso - Administrador	<ul style="list-style-type: none"> <li>- Diseño y creación del controlador de correos</li> <li>- Diseño y creación de los servicios que utilizará el controlador de correos</li> <li>- Diseño y creación de la interfaz de recuperación de contraseña y la interfaz que envía el código de recuperación.</li> </ul>	2

RF02	Crear cuentas de empleado – Administrador / Gerente	<ul style="list-style-type: none"> <li>- Pruebas de interfaz y servicios.</li> <li>- Diseño y creación del controlador de registro</li> <li>- Diseño y creación de los servicios que utilizará el controlador de registro</li> <li>- Implementación de todas las operaciones CRUD en el controlador y servicio correspondiente para las cuentas del sistema.</li> <li>- Diseño y creación de la ruta del api de registro</li> <li>- Diseño y creación de la interfaz de listado, registro y eliminación de cuentas de empleados.</li> <li>- Diseño y creación de la interfaz de edición de una cuenta empleado</li> </ul>	3
RF04	Sistema para cambiar la contraseña de ingreso - Administrador	<ul style="list-style-type: none"> <li>- Pruebas de interfaz y servicios.</li> <li>- Diseño de la interfaz para editar perfil</li> <li>- Conexión con el api de registro y utilización del servicio de actualizar.</li> </ul>	3
RF06	Sistema para recuperar la contraseña de ingreso - Empleado	<ul style="list-style-type: none"> <li>- Pruebas de interfaz y servicios.</li> <li>- Utilización del controlador de correos.</li> <li>- Diseño de una plantilla para el envío de un código de recuperación.</li> <li>- Correcciones y Pruebas.</li> </ul>	1
RF12	Control de deudas - Administrador	<ul style="list-style-type: none"> <li>- Diseño y creación del controlador de deudas</li> </ul>	4

---

- Diseño y creación de los servicios que utilizará el controlador de deudas
- Diseño y creación de la ruta del api de deudas.
- Diseño y creación de la interfaz de listado, registro, edición y eliminación de las deudas.
- Pruebas de interfaz y servicios.

---

*Nota.* Esta tabla describe la lista detallada de actividades del sprint no. 1. Fuente: Elaboración Propia.

### **Objetivo de sprint 2**

Se establece el objetivo o meta del segundo sprint que se puede lograr mediante la implementación de las diferentes tareas asignadas. La meta de este sprint se basa en el funcionamiento del segundo microservicio de clientes conjuntamente con la parte gráfica del mismo. En la tabla 17 se indica la definición los objetivos del segundo sprint.

### **Tabla 17**

#### *Sprint Goal 2*

---

<b>Product Goal - Sprint 2</b>	Funcionamiento del micro servicio de clientes e implementación en la interfaz gráfica.
Sprint Goal 1	Diseño de la interfaz gráfica para la implementación de los servicios del microservicio
Sprint Goal 2	Correcto funcionamiento del módulo de clientes y todas las operaciones CRUD

---

*Nota.* Esta tabla describe el objetivo del sprint 2. Fuente: Elaboración Propia.

### **Sprint 2**

En la planificación del segundo sprint abarca el desarrollo de las tareas correspondientes para el control de clientes en los diferentes roles de usuario. Esta planificación se presenta en la tabla 18.

**Tabla 18***Planificación Detallada del Sprint no. 2*

<b>Sprint N: 2</b>			
<b>Fecha de Inicio: 28/10/2022</b>			
<b>Fecha de finalización: 11/11/2022</b>			
<b>Código</b>	<b>Descripción</b>	<b>Tareas</b>	<b>Valor de Estimación (Días)</b>
RF11	Registro de clientes – Administrador	<ul style="list-style-type: none"> <li>- Diseño y creación del controlador de clientes</li> <li>- Diseño y creación de los servicios que utilizará el controlador de clientes</li> <li>- Implementación de los métodos CRUD para clientes</li> <li>- Diseño y creación de la ruta del api de clientes</li> <li>- Diseño y creación de la interfaz de listado, registro, edición y eliminación de las deudas</li> <li>- Pruebas de interfaz y servicios.</li> </ul>	10
RF17	Registro de clientes - Empleado	<ul style="list-style-type: none"> <li>- Configuración de la interfaz que deshabilite la opción de eliminar cliente</li> <li>- Implementación de filtros para el listado de clientes</li> <li>- Correcciones y pruebas.</li> </ul>	4

*Nota.* Esta tabla describe la lista detallada de actividades del sprint no. 2. Fuente: Elaboración Propia.

**Objetivo de sprint 3**

Se establece el objetivo o meta del tercer sprint que se puede lograr mediante la implementación de las diferentes tareas asignadas. La meta de este sprint se basa en el

funcionamiento del tercer microservicio de productos, de la misma manera con la parte gráfica del mismo. En la tabla 19 se indica la definición los objetivos del tercer sprint.

**Tabla 19**

*Sprint Goal 3*

<b>Product Goal - Sprint 3</b>	Funcionamiento del micro servicio de productos e implementación en la interfaz gráfica.
Sprint Goal 1	Diseño de la interfaz gráfica para la implementación de los servicios del microservicio
Sprint Goal 2	Correcto funcionamiento de las operaciones CRUD del módulo de productos y categorías.
Sprint Goal 3	Correcto funcionamiento de las operaciones CRUD del módulo de proveedores.
Sprint Goal 4	Correcto funcionamiento de las operaciones CRUD del módulo de pedidos.
Sprint Goal 5	Correcto funcionamiento de las operaciones CRUD del módulo de gestión de inventario.

*Nota.* Esta tabla describe el objetivo del sprint 3. Fuente: Elaboración Propia.

***Sprint 3***

En la planificación del tercer sprint, engloba las tareas relacionadas para la gestión de inventario y proveedores tanto para los dos usuarios como gerente y empleado, en base a ello asignar los permisos y acciones correspondientes en este módulo. Esta planificación se presenta en la tabla 20.

**Tabla 20**

*Planificación Detallada del Sprint no. 3*

<b>Sprint N:</b> 3
<b>Fecha de Inicio:</b> 14/11/2022
<b>Fecha de finalización:</b> 05/12/2022

Código	Descripción	Tareas	Valor de Estimación (Días)
RF07	Control de inventario – Administrador	<ul style="list-style-type: none"> <li>- Diseño y creación del controlador de productos.</li> <li>- Diseño y creación de los servicios que utilizará el controlador de productos.</li> <li>- Implementación de los métodos CRUD para productos</li> <li>- Diseño y creación de la ruta del api de productos</li> <li>- Diseño y creación de la interfaz de listado, registro, edición y eliminación de los productos</li> <li>- Diseño y creación del controlador de bodegas.</li> <li>- Diseño y creación de los servicios que utilizará el controlador de bodegas.</li> <li>- Implementación de los métodos CRUD para bodegas.</li> <li>- Diseño y creación de la ruta del api de bodegas.</li> <li>- Diseño y creación de la interfaz de listado, registro, edición y eliminación de las bodegas.</li> <li>- Diseño y creación del controlador de inventario.</li> <li>- Diseño y creación de los servicios que utilizará el controlador de inventario.</li> <li>- Implementación de los métodos CRUD para inventario.</li> </ul>	7

---

		- Diseño y creación de la ruta del api de inventario.	
		- Diseño y creación de la interfaz de listado, registro, edición y eliminación del inventario.	
		- Pruebas de interfaz y servicios.	
RF10	Registro de Proveedores - Administrador	- Diseño y creación del controlador de proveedores	8
		- Diseño y creación de los servicios que utilizará el controlador de proveedores	
		- Implementación de los métodos CRUD para proveedores	
		- Diseño y creación de la ruta del api de proveedores	
		- Diseño y creación de la interfaz de listado, registro, edición y eliminación de los proveedores	
		- Diseño y creación del controlador de pedidos	
		- Diseño y creación de los servicios que utilizará el controlador de pedidos	
		- Implementación de los métodos CRUD para pedidos	
		- Diseño y creación de la ruta del api de pedidos	
		- Diseño y creación de la interfaz de listado, registro, edición y eliminación de los pedidos.	
		- Pruebas de interfaz y servicios.	
RF13	Control de inventario - Empleado	- Configuración de la interfaz que deshabilite la opción de eliminar producto	3



		<ul style="list-style-type: none"> <li>- Configuración de la interfaz que deshabilite la opción de eliminar bodega</li> <li>- Configuración de la interfaz que deshabilite la opción de eliminar inventario</li> <li>- Implementación de filtros para el listado de productos</li> <li>- Implementación de filtros para el listado de bodega</li> <li>- Implementación de filtros para el listado del inventario.</li> <li>- Correcciones y pruebas.</li> </ul>	
RF16	Registro de Proveedores - Empleado	<ul style="list-style-type: none"> <li>- Configuración de la interfaz que deshabilite la opción de eliminar proveedor</li> <li>- Configuración de la interfaz que deshabilite la opción de eliminar pedido</li> <li>- Implementación de filtros para el listado de productos</li> <li>- Implementación de filtros para el listado de proveedores</li> <li>- Implementación de filtros para el listado de pedidos.</li> <li>- Correcciones y pruebas.</li> </ul>	3

---

*Nota.* Esta tabla describe la lista detallada de actividades del sprint no. 3. Fuente: Elaboración Propia.

#### **Objetivo de sprint 4**

Se establece el objetivo o meta del cuarto sprint que se puede lograr mediante la implementación de las diferentes tareas asignadas. La meta de este sprint se basa en el

funcionamiento del cuarto microservicio de ventas, conjuntamente con la parte gráfica del mismo del sistema web. En la tabla 21 se indica la definición los objetivos del cuarto sprint.

**Tabla 21**

*Sprint Goal 4*

<b>Product Goal - Sprint 4</b>	Funcionamiento del micro servicio de ventas e implementación en la interfaz gráfica.
Sprint Goal 1	Diseño de la interfaz gráfica para la implementación de los servicios del microservicio
Sprint Goal 2	Correcto funcionamiento de las operaciones CRUD del módulo de venas.
Sprint Goal 3	Correcto funcionamiento de las funciones de caja diaria y su interacción con los módulos relacionados.
Sprint Goal 4	Correcto funcionamiento de las del módulo de reportes y gráficas de un dashboard.

*Nota.* Esta tabla describe el objetivo del sprint 4. Fuente: Elaboración Propia.

***Sprint 4***

En la planificación del cuarto sprint, se enlistan las tareas relacionadas para realizar la gestión y control de ventas mediante las diferentes acciones según el rol de usuario. Esta planificación se presenta en la tabla 22.

**Tabla 22**

*Planificación Detallada del Sprint no. 4*

<b>Sprint N: 4</b>			
<b>Fecha de Inicio:</b> 07/12/2022			
<b>Fecha de finalización:</b> 27/12/2022			
<b>Código</b>	<b>Descripción</b>	<b>Tareas</b>	<b>Valor de Estimación (Días)</b>
RF09	Sistema de control de ventas – Administrador	- Diseño y creación del controlador de ventas	16

- 
- Diseño y creación de los servicios que utilizará el controlador de ventas
  - Implementación de los métodos CRUD para ventas
  - Diseño y creación de la ruta del api de ventas
  - Diseño y creación de la interfaz de listado, registro, edición y eliminación de las ventas
  - Diseño y creación del controlador de cajas
  - Diseño y creación de los servicios que utilizará el controlador de cajas
  - Implementación de los métodos CRUD para cajas
  - Diseño y creación de la ruta del api de cajas
  - Diseño y creación de la interfaz de listado, registro, edición y eliminación de las cajas
  - Diseño y creación del controlador de reportes
  - Diseño y creación de los servicios que utilizará el controlador de reportes
  - Implementación de los métodos CRUD para reportes
  - Diseño y creación de la ruta del api de reportes
  - Diseño y creación de la interfaz de listado, registro, edición y eliminación de los reportes.

RF15	Sistema de control de ventas - Empleado	<ul style="list-style-type: none"> <li>- Pruebas de interfaz y servicios.</li> <li>- Configuración de la interfaz que deshabilite la opción de eliminar venta.</li> <li>- Implementación de filtros para el listado de ventas.</li> <li>- Correcciones y pruebas.</li> </ul>	4
------	---	--	---

*Nota.* Esta tabla describe la lista detallada de actividades del sprint no. 4. Fuente: Elaboración Propia.

### Objetivo de sprint 5

Se establece el objetivo o meta del quinto sprint que se puede lograr mediante la implementación de las diferentes tareas asignadas. La meta de este sprint se basa en el funcionamiento del microservicio de facturación, conjuntamente con la parte gráfica del mismo en el sistema web. En la tabla 23 se indica la definición los objetivos del quinto sprint.

### Tabla 23

#### *Sprint Goal 5*

<b>Product Goal - Sprint 5</b>	Funcionamiento del micro servicio de facturación e implementación en la interfaz gráfica.
Sprint Goal 1	Diseño de la interfaz gráfica para la implementación de los servicios del microservicio.
Sprint Goal 2	Correcto funcionamiento de las del módulo de facturas.
Sprint Goal 3	Correcto funcionamiento de las del módulo de nota de crédito.

*Nota.* Esta tabla describe el objetivo del sprint 5. Fuente: Elaboración Propia.

### *Sprint 5*

En la planificación del quinto sprint, se enlistan o presentan las tareas relacionadas al apartado de gestionar el módulo de facturación electrónica con las acciones correspondientes al perfil de usuario en uso. La mencionada planificación se presenta en la tabla 24.

**Tabla 24***Planificación Detallada del Sprint no. 5*

<b>Sprint N: 5</b>			
<b>Fecha de Inicio:</b> 29/12/2022			
<b>Fecha de finalización:</b> 13/01/2022			
<b>Código</b>	<b>Descripción</b>	<b>Tareas</b>	<b>Valor de Estimación (Días)</b>
RF08	Sistema de facturación – Administrador	<ul style="list-style-type: none"> <li>- Diseño y creación del controlador de comprobantes</li> <li>- Diseño y creación de los servicios que utilizará el controlador de comprobantes</li> <li>- Implementación de los métodos CRUD para comprobantes</li> <li>- Diseño y creación de la ruta del api de comprobantes</li> <li>- Diseño y creación de la interfaz de listado, registro, edición y eliminación de las comprobantes</li> <li>- Conexión con los servicios de facturación</li> <li>- Pruebas de interfaz y servicios.</li> </ul>	10
RF14	Sistema de facturación - Empleado	<ul style="list-style-type: none"> <li>- Configuración de la interfaz que deshabilite la opción de eliminar un comprobante</li> <li>- Implementación de filtros para el listado de comprobantes.</li> <li>- Correcciones y pruebas.</li> </ul>	5

*Nota.* Esta tabla describe la lista detallada de actividades del sprint no. 5. Fuente: Elaboración Propia.

## Criterios de Aceptación

### *Sprint 1*

Los criterios definidos por las partes interesadas para la aceptación en el sprint uno, se detallan en la Tabla 25.

**Tabla 25**

*Criterios de aceptación del sprint no. 1*

<b>ID Requerimiento</b>	<b>ID Criterio</b>	<b>Criterio de Aceptación</b>	<b>Contexto</b>	<b>Evento</b>	<b>Resultado Esperado</b>
RF01	CA1-01	Credenciales correctas	En caso que ingrese las credenciales registradas en el sistema	Cuando trate de iniciar sesión en el sistema.	Ingreso al sistema
	CA1-02	Nombre de usuario incorrecto	En caso que ingrese la credencial de usuario no registrado en el sistema	Cuando trate de iniciar sesión en el sistema.	El sistema le indicará que ingresó un usuario invalido.
	CA1-03	Contraseña incorrecta	En caso que ingrese la credencial de contraseña no registrado en el sistema	Cuando trate de iniciar sesión en el sistema.	El sistema le indicará que ingresó una contraseña inválida.
	CA1-04	Usuario y contraseña incorrectos	En caso que ingrese credenciales no registrados en el sistema	Cuando trate de iniciar sesión en el sistema.	El sistema le indicará que ingresó un usuario o contraseña inválida.

RF05	CA1-05	Credenciales correctas	En caso que ingrese las credenciales registradas en el sistema	Cuando trate de iniciar sesión en el sistema.	Ingreso al sistema
	CA1-06	Nombre de usuario incorrecto	En caso que ingrese la credencial de usuario no registrado en el sistema	Cuando trate de iniciar sesión en el sistema.	El sistema le indicará que ingresó un usuario invalido.
	CA1-07	Contraseña incorrecta	En caso que ingrese la credencial de contraseña no registrado en el sistema	Cuando trate de iniciar sesión en el sistema.	El sistema le indicará que ingresó una contraseña inválida.
	CA1-08	Usuario y contraseña incorrectos	En caso que ingrese credenciales no registrados en el sistema	Cuando trate de iniciar sesión en el sistema.	El sistema le indicará que ingresó un usuario o contraseña inválida.
RF03	CA1-09	Ingreso de código validación correcto	En caso que ingrese el código que recibió en su correo	Cuando el usuario no recuerde su contraseña y utilice el sistema de recuperación	El sistema le permite ingresar una nueva contraseña
	CA1-10	Actualización de contraseña	En caso que haya validado el código que	Cuando el usuario no recuerde su contraseña y	El sistema actualiza la contraseña para el login

			recibió en su correo	utilice el sistema de recuperación	
	CA1-11	Ingreso de código validación incorrecto	En caso que ingrese un código diferente al que recibió en su correo	Cuando el usuario no recuerde su contraseña y utilice el sistema de recuperación	El sistema informa que el código ingresado es incorrecto
RF02	CA1-12	Registrar empleado	En caso que no exista el empleado en el sistema	Cuando trate de ingresar al sistema de registro de empleado	El sistema registra un nuevo empleado.
	CA1-12	Empleado repetido	En caso que exista el empleado en el sistema.	Cuando trate de ingresar al sistema de registro de empleado	El sistema informa que el empleado ya existe
	CA1-13	Eliminar empleado	En caso que el empleado finalice sus actividades en el establecimiento	Cuando trate de ingresar al sistema de registro de empleado	El sistema elimina el empleado cambiando su estado
	CA1-14	Modificar empleado	En caso que el empleado cambie sus datos	Cuando trate de ingresar al sistema de registro de empleado	El sistema actualiza los datos del empleado
	CA1-15	Listar empleado	N/A	Cuando trate de ingresar al	El sistema indicará todos



				sistema de registro de empleado	los empleados registrados
RF04	CA1-16	Actualización de contraseña	En caso que ingrese una nueva contraseña en el sistema	Cuando el administrador cambie la contraseña por defecto	El sistema actualiza la contraseña del administrador para el login
RF06	CA1-17	Ingreso de código validación correcto	En caso que ingrese el código que recibió en su correo	Cuando el usuario no recuerde su contraseña y utilice el sistema de recuperación	El sistema le permite ingresar una nueva contraseña
	CA1-18	Actualización de contraseña	En caso que haya validado el código que recibió en su correo	Cuando el usuario no recuerde su contraseña y utilice el sistema de recuperación	El sistema actualiza la contraseña para el login
	CA1-19	Ingreso de código validación incorrecto	En caso que ingrese un código diferente al que recibió en su correo	Cuando el usuario no recuerde su contraseña y utilice el sistema de recuperación	El sistema informa que el código ingresado es incorrecto
	CA1-20	Actualización de contraseña	En caso que ingrese una nueva	Cuando el administrador cambie la	El sistema actualiza la contraseña del

RF012	CA1-21	Registrar Categoría	contraseña en el sistema En caso que no exista una categoría	contraseña por defecto En caso que ingrese una categoría en el sistema de registro de categoría	administrador para el login. El sistema registrará la categoría ingresada
	CA1-22	Categoría Repetida	En caso que exista una categoría	En caso que ingrese una categoría en el sistema de registro de categoría	El sistema informara que la categoría ya se encuentra registrada.
	CA1-23	Registrar Deuda	En caso que no exista una deuda	Cuando trate de ingresar al sistema de registro de deuda	El sistema registra una nueva deuda
	CA1-24	Notificar de pago	En caso que el día de pago de una deuda este próximo a expirar	Cuando el sistema detecta la fecha límite de pago que fue configurada	El sistema enviara un correo notificando que se acerca la fecha de pago.
	CA1-25	Actualizar Deuda	En caso que realice un pago diferente al configurado en el registro inicial	Cuando trate de ingresar al sistema de registro de deuda	El sistema actualiza los datos de la deuda
	CA1-26	Listar Deudas	N/A	Cuando trate de ingresar al sistema de	El sistema indicara todas las deudas y

---

registro de  
deuda

los valores  
totales a  
pagar

*Nota.* Esta tabla presenta los principales criterios de aceptación del Sprint no 1. Fuente:

Elaboración propia.

### ***Sprint 2***

Los criterios definidos por las partes interesadas para la aceptación en el sprint dos, se detallan en la Tabla 26.

**Tabla 26**

*Criterios de Aceptación del Sprint no. 2*

<b>ID Requerimiento</b>	<b>ID Criterio</b>	<b>Criterio de Aceptación</b>	<b>Contexto</b>	<b>Evento</b>	<b>Resultado Esperado</b>
RF11	CA2-27	Registrar cliente	En caso que no exista el cliente en el sistema	Cuando trate de ingresar al sistema de registro de cliente	El sistema registra un nuevo cliente
	CA2-28	Cliente repetido	En caso que exista el cliente en el sistema	Cuando trate de ingresar al sistema de registro de cliente	El sistema informa que el cliente ya existe
	CA2-29	Eliminar cliente	En caso que el cliente finalice sus actividades en el establecimiento	Cuando trate de ingresar al sistema de registro de cliente	El sistema elimina el cliente cambiando su estado

	CA2-30	Modificar cliente	En caso que el cambie sus datos	Cuando trate de ingresar al sistema de registro de cliente	El sistema actualiza los datos del cliente
	CA2-31	Listar clientes	N/A	Cuando trate de ingresar al sistema de registro de cliente	El sistema indicará todos los clientes registrados
RF17	CA2-32	Registrar cliente	En caso que no exista el cliente en el sistema	Cuando trate de ingresar al sistema de registro de cliente	El sistema registra un nuevo cliente
	CA2-33	Cliente repetido	En caso que no exista el cliente en el sistema	Cuando trate de ingresar al sistema de registro de cliente	El sistema informa que el cliente ya existe
	CA2-34	Listar clientes	N/A	Cuando trate de ingresar al sistema de registro de cliente	El sistema indicará todos los clientes registrados

---

*Nota.* Esta tabla presenta los principales criterios de aceptación del Sprint no 2. Fuente:

Elaboración propia.

### **Sprint 3**

Los criterios definidos por las partes interesadas para la aceptación en el sprint tres, se detallan en la Tabla 27.

**Tabla 27**

*Criterios de Aceptación del Sprint no. 3*

<b>ID Requerimiento</b>	<b>ID Criterio</b>	<b>Criterio de Aceptación</b>	<b>Contexto</b>	<b>Evento</b>	<b>Resultado Esperado</b>
RF07	CA3-35	Ingresar un producto	En caso que no exista el producto en el sistema	En caso que ingrese un producto en el sistema de registro de producto	El sistema registrará el producto en el inventario
	CA3-36	Producto repetido	En caso que exista el producto en el sistema	En caso que ingrese un producto en el sistema de registro de producto	El sistema informará que el producto ya se encuentra registrado
	CA3-37	Ingreso de una categoría	En caso que no exista la categoría en el sistema	En caso que ingrese una categoría en el sistema de registro de categoría	El sistema registrará la categoría ingresada
	CA3-38	Categoría repetida	En caso que exista la categoría	En caso que ingrese una categoría	El sistema informará que la categoría ya está registrada

		categoria en el sistema	categoria en el sistema de registro de categoria	categoria ya se encuentra registrada
CA3-39	Modificar un producto	En caso que cambien los datos del producto	En caso que ingrese en el sistema de registro de producto	El sistema traerá los datos del producto y permitirá modificarlos
CA3-40	Modificar el stock del producto	En caso que haya un ingreso o salida de mercadería	En caso que ingrese en el sistema de inventario	El sistema modifica la existencia del producto en el inventario
CA3-41	Listar productos por categoría	N/A	Cuando se despliegue el listado de categorías a seleccionar	El sistema indicará todos los productos asociados a la categoría seleccionada
CA3-42	Categoría sin productos	En caso que una categoría no tenga productos asociados	Cuando se despliegue el listado de categorías a seleccionar	A continuación del nombre de la categoría, se mostrará entre paréntesis el siguiente texto "Sin

					Productos asociados"
RF10	CA3-43	Registrar proveedor	En caso que no exista el proveedor en el sistema.	Cuando trate de ingresar al sistema de registro de proveedor	El sistema registra un nuevo proveedor.
	CA3-44	Proveedor repetido	En caso que exista el proveedor en el sistema.	Cuando trate de ingresar al sistema de registro de proveedor	El sistema informa que el proveedor ya existe.
	CA3-45	Eliminar proveedor	En caso que el proveedor finalice sus actividades económicas en el establecimiento	Cuando trate de ingresar al sistema de registro de proveedor	El sistema elimina el proveedor cambiando su estado.
	CA3-46	Modificar proveedor	En caso que el proveedor cambie sus datos	Cuando trate de ingresar al sistema de registro de proveedor	El sistema actualiza los datos del proveedor.
	CA3-47	Listar proveedores	N/A	Cuando trate de ingresar al sistema de registro de proveedor	El sistema indicara todos los proveedores registrados.

RF13	CA3-48	Ingresar un producto	En caso que no exista el producto en el sistema	En caso que ingrese un producto en el sistema de registro de producto	El sistema registrará el producto en el inventario
	CA3-49	Producto repetido	En caso que exista el producto en el sistema	En caso que ingrese un producto en el sistema de registro de producto	El sistema informará que el producto ya se encuentra registrado
	CA3-50	Ingreso de una categoría	En caso que no exista la categoría en el sistema	En caso que ingrese una categoría en el sistema de registro de categoría	El sistema registrará la categoría ingresada
	CA3-51	Categoría repetida	En caso que exista la categoría en el sistema	En caso que ingrese una categoría en el sistema de registro de categoría	El sistema informará que la categoría ya se encuentra registrada
	CA3-52	Modificar el stock del producto	En caso que haya un ingreso o	En caso que ingrese en el	El sistema modifica la existencia del producto



			salida de mercadería	sistema de inventario	en el inventario
	CA3-53	Listar productos por categoría	N/A	Cuando se despliegue el listado de categorías a seleccionar	El sistema indicará todos los productos asociados a la categoría seleccionada
	CA3-54	Categoría sin productos	En caso que una categoría no tenga productos asociados	Cuando se despliegue el listado de categorías a seleccionar	A continuación del nombre de la categoría, se mostrará entre paréntesis el siguiente texto "Sin Productos asociados".
RF16	CA3-55	Registrar proveedor	En caso que no exista el proveedor en el sistema	Cuando trate de ingresar al sistema de registro de proveedor	El sistema registra un nuevo proveedor
	CA3-56	Proveedor repetido	En caso que exista el proveedor en el sistema	Cuando trate de ingresar al sistema de registro de proveedor	El sistema informa que el proveedor ya existe

CA3-57	Listar proveedores	N/A	Cuando trate de ingresar al sistema de registro de proveedor	El sistema indicará todos los proveedores registrados
--------	--------------------	-----	--	---

*Nota.* Esta tabla presenta los principales criterios de aceptación del Sprint no 4. Fuente:

Elaboración propia.

### ***Sprint 4***

Los criterios definidos por las partes interesadas para la aceptación en el sprint cuatro, se detallan en la Tabla 28.

**Tabla 28**

*Criterios de Aceptación del Sprint no. 4*

<b>ID Requerimiento</b>	<b>ID Criterio</b>	<b>Criterio de Aceptación</b>	<b>Contexto</b>	<b>Evento</b>	<b>Resultado Esperado</b>
RF09	CA6-55	Realizar venta	En caso que se realice la compra de un producto	Cuando realice una venta en el sistema	El sistema registrará la venta
	CA6-56	Realizar reporte de venta diaria	Al finalizar el turno del día	Cuando seleccione la opción de realizar caja diaria	El sistema indicará todas las ventas realizadas en el día y las registrará en la base de datos
	CA6-57	Realizar reporte de	En caso que se requiera verificar las	Cuando utilice el	El sistema imprimirá en pantalla el

		las ventas semanales	ventas semanales	sistema de reportes	total de las ventas por semana
	CA6-58	Realizar reporte de las ventas mensuales	En caso que se requiera verificar las ventas mensuales	Cuando utilice el sistema de reportes	El sistema imprimirá en pantalla el total de las ventas por mes
RF15	CA6-59	Realizar venta	En caso que se realice la compra de un producto	Cuando realice una venta en el sistema	El sistema registrará la venta
	CA6-60	Realizar reporte de venta diaria	Al finalizar el turno del día	Cuando seleccione la opción de realizar caja diaria	El sistema indicará todas las ventas realizadas en el día y las registrará en la base de datos

---

*Nota.* Esta tabla presenta los principales criterios de aceptación del Sprint no 4. Fuente:

Elaboración propia.

### ***Sprint 5***

Los criterios definidos por las partes interesadas para la aceptación en el sprint cinco, se detallan en la Tabla 29.

**Tabla 29***Criterios de Aceptación del Sprint no. 5*

<b>ID Requerimiento</b>	<b>ID Criterio</b>	<b>Criterio de Aceptación</b>	<b>Contexto</b>	<b>Evento</b>	<b>Resultado Esperado</b>
RF08	CA7-61	Enviar Factura	En caso que un cliente solicite una factura de la compra realizada	Cuando al realizar una venta se seleccione la opción de facturar	El sistema enviará la factura al sistema del SRI
	CA7-62	Listar Factura	N/A	Cuando utilice el sistema de reportes	El sistema mostrará en pantalla un listado de facturas
	CA7-63	Autorizar Factura	En caso de autorizar la factura al SRI de la venta realizada	Cuando al realizar el envío de la factura proceda con la autorización de la misma	El estado de la presente factura pasará a autorizado.
RF14	CA7-64	Enviar Factura con rol de empleado	En caso que un cliente solicite una factura de la compra realizada	Cuando al realizar una venta se seleccione la opción de facturar	El sistema enviará la factura al sistema del SRI.
	CA7-65	Listar Factura	N/A	Cuando utilice el sistema de reportes	El sistema mostrará en pantalla un

				listado de facturas.
CA7-64	Autorizar Factura	En caso de autorizar la factura al SRI de la venta realizada	Cuando al realizar el envío de la factura proceda con la autorización de la misma	El estado de la presente factura. pasará a autorizado.

---

*Nota.* Esta tabla presenta los principales criterios de aceptación del Sprint no 5. Fuente:

Elaboración propia.

## Capítulo IV: Resultados y Discusión

En el presente capítulo una vez completado el sistema web para la ferretería Electrocat, se presentan los resultados del desarrollo del sistema web administración. Se destacan principalmente los resultados de cada sprint realizado, que se lograron en la implementación del sistema, de la misma manera cumpliendo con los requerimientos del sistema. Finalizando los sprints correspondientes se procedió a realizar las respectivas pruebas de funcionamiento del sistema.

### Revisión y Retrospectiva

#### *Revisión Sprint 1*

Los detalles de la revisión del desarrollo del primer sprint se indican en la tabla 30. En la cual detalla las fechas, participantes, razones y resultados de la reunión.

**Tabla 30**

*Revisión del sprint 1.*

<b>Reunión Sprint Review</b>	
<b>Motivo</b>	Revisión de los respectivos entregables del sprint número 1. Así mismo el cumplimiento del sprint backlog.
<b>Participantes</b>	Product Owner, Scrum Master y Developers
<b>Fecha</b>	27/10/2022
<b>Duración</b>	1 hora
<b>Entrada</b>	Módulos del sistema
<b>Salida</b>	Retroalimentación y retrospectiva de la entrega realizada.

*Nota.* Información sobre la reunión para la revisión del sprint no. 1. Fuente: Elaboración propia.

#### *Retrospectiva Sprint 1*

En la tabla 31 se indican los resultados que se obtuvieron en la reunión de revisión del primer sprint, en el cual se dividen en diferentes apartados y con ello identificar los aspectos que salieron correctos, los que requieren revisión y los que requieren algún tipo de mejora para lograr el objetivo.

**Tabla 31***Revisión del sprint 1.*

<b>¿Qué salió bien en el desarrollo del sprint?</b>	<b>¿Qué salió mal en el desarrollo del sprint?</b>	<b>¿Qué mejoras se van a implementar?</b>
Correcta implementación del sistema de login y recuperación de contraseñas.	Fallo al enviar correos de notificación del control de deudas.	Mejora en la lógica de notificación de deudas. Corrección en la base de datos y solución del envío de correos.
Funcionamiento correcto de la interfaz gráfica. Registro correcto del control de deudas	Eliminación de campos incorrectos de la tabla deudas en la base de datos.	

*Nota.* Esta tabla muestra información resultante de la reunión realizada para la revisión del sprint no. 1. Fuente: Elaboración propia.

**Revisión Sprint 2**

Los detalles de la revisión del desarrollo del segundo sprint se indican en la tabla 32. En la cual detalla las fechas, participantes, razones y resultados de la reunión.

**Tabla 32***Revisión del sprint 2.*

<b>Reunión Sprint Review</b>	
<b>Motivo</b>	Revisión de los respectivos entregables del sprint número 2. Así mismo el cumplimiento del sprint backlog.
<b>Participantes</b>	Product Owner, Scrum Master y Developers
<b>Fecha</b>	11/11/2022
<b>Duración</b>	1 hora
<b>Entrada</b>	Módulos del sistema
<b>Salida</b>	Retroalimentación y retrospectiva de la entrega realizada.

*Nota.* Información sobre la reunión para la revisión del sprint no. 2. Fuente: Elaboración propia.

### **Retrospectiva Sprint 2**

En la tabla 33 se indican los resultados que se obtuvieron en la reunión de revisión del segundo sprint, en el cual se dividen en diferentes apartados y con ello identificar los aspectos que salieron correctos, los que requieren revisión y los que requieren algún tipo de mejora para lograr el objetivo.

**Tabla 33**

*Retrospectiva y Retroalimentación del Sprint no. 2.*

<b>¿Que salió bien en el desarrollo del sprint?</b>	<b>¿Que salió mal en el desarrollo del sprint?</b>	<b>¿Qué mejoras se van a implementar?</b>
Correcta implementación del sistema de clientes Correcto funcionamiento de los filtros y buscadores de clientes	Fallo en la validación del formulario de registro en los campos de teléfono e identificación	Implementar y mejorar la validación de los datos en frontend y backend.

*Nota.* Esta tabla muestra información resultante de la reunión realizada para la revisión del sprint no. 2. Fuente: Elaboración propia.

### **Revisión Sprint 3**

Los detalles de la revisión del desarrollo del tercer sprint se indican en la tabla 34. En la cual detalla las fechas, participantes, razones y resultados de la reunión.

**Tabla 34**

*Revisión del Sprint 3.*

<b>Reunión Sprint Review</b>	
<b>Motivo</b>	Revisión de los respectivos entregables del sprint número 3. Así mismo el cumplimiento del sprint backlog.
<b>Participantes</b>	Product Owner, Scrum Master y Developers
<b>Fecha</b>	06/12/2022
<b>Duración</b>	1 hora
<b>Entrada</b>	Módulos del sistema
<b>Salida</b>	Retroalimentación y retrospectiva de la entrega realizada.

*Nota.* Información sobre la reunión para la revisión del sprint no. 3. Fuente: Elaboración propia.



### **Retrospectiva Sprint 3**

En la tabla 35 se indican los resultados que se obtuvieron en la reunión de revisión del tercer sprint, en el cual se dividen en diferentes apartados y con ello identificar los aspectos que salieron correctos, los que requieren revisión y los que requieren algún tipo de mejora para lograr el objetivo.

**Tabla 35**

*Retrospectiva y Retroalimentación del Sprint no. 3.*

<b>¿Que salió bien en el desarrollo del sprint?</b>	<b>¿Que salió mal en el desarrollo del sprint?</b>	<b>¿Qué mejoras se van a implementar?</b>
Correcta implementación de los servicios de productos, inventario, proveedores y pedidos.	Error al eliminar productos.	Aumento de los campos faltantes en la tabla producto.
Funcionamiento correcto de la interfaz gráfica.	Falta de atributos al momento de registrar un producto.	Corrección y control del servicio de eliminar productos.
Correcto funcionamiento de los filtros y buscadores de los componentes.		

*Nota.* Esta tabla muestra información resultante de la reunión realizada para la revisión del sprint no. 3. Fuente: Elaboración propia.

### **Revisión Sprint 4**

Los detalles de la revisión del desarrollo del cuarto sprint se indican en la tabla 36. En la cual detalla las fechas, participantes, razones y resultados de la reunión.

**Tabla 36**

*Revisión del Sprint 4.*

<b>Reunión Sprint Review</b>	
<b>Motivo</b>	Revisión de los respectivos entregables del sprint número 4. Así mismo el cumplimiento del sprint backlog.
<b>Participantes</b>	Product Owner, Scrum Master y Developers

<b>Fecha</b>	28/12/2022
<b>Duración</b>	1 hora
<b>Entrada</b>	Módulos del sistema
<b>Salida</b>	Retroalimentación y retrospectiva de la entrega realizada.

*Nota.* Información sobre la reunión para la revisión del sprint no. 4. Fuente: Elaboración propia.

### **Retrospectiva Sprint 4**

En la tabla 37 se indican los resultados que se obtuvieron en la reunión de revisión del cuarto sprint, en el cual se dividen en diferentes apartados y con ello identificar los aspectos que salieron correctos, los que requieren revisión y los que requieren algún tipo de mejora para lograr el objetivo.

**Tabla 37**

*Retrospectiva y Retroalimentación del Sprint no. 4.*

<b>¿Que salió bien en el desarrollo del sprint?</b>	<b>¿Que salió mal en el desarrollo del sprint?</b>	<b>¿Qué mejoras se van a implementar?</b>
Correcta implementación de los servicios de ventas. Correcto funcionamiento de los filtros y buscadores de los componentes para productos y clientes.	Errores en la selección de precios en la interfaz gráfica al momento de realizar una venta.	Corrección en la selección de precios al realizar una venta.
Correcto control del inventario de los productos Correcta visualización de los reportes.	Falta de campos en la tabla de ventas para el manejo de facturas y notas de crédito. Error en la lógica del servicio de caja.	Aumento de los campos faltantes en la tabla de venta. Mejora en el diseño de los componentes Corrección e implementación de los servicios de caja.

*Nota.* Esta tabla muestra información resultante de la reunión realizada para la revisión del sprint no. 4. Fuente: Elaboración propia.

### **Revisión Sprint 5**

Los detalles de la revisión del desarrollo del quinto sprint se indican en la tabla 38. En la cual detalla las fechas, participantes, razones y resultados de la reunión.

**Tabla 38**

*Revisión del Sprint 5.*

<b>Reunión Sprint Review</b>	
<b>Motivo</b>	Revisión de los respectivos entregables del sprint número 5. Así mismo el cumplimiento del sprint backlog.
<b>Participantes</b>	Product Owner, Scrum Master y Developers
<b>Fecha</b>	14/01/2022
<b>Duración</b>	1 hora
<b>Entrada</b>	Módulos del sistema
<b>Salida</b>	Retroalimentación y retrospectiva de la entrega realizada.

*Nota.* Información sobre la reunión para la revisión del sprint no. 5. Fuente: Elaboración propia.

### **Retrospectiva Sprint 5**

En la tabla 39 se indican los resultados que se obtuvieron en la reunión de revisión del quinto sprint, en el cual se dividen en diferentes apartados y con ello identificar los aspectos que salieron correctos, los que requieren revisión y los que requieren algún tipo de mejora para lograr el objetivo.

**Tabla 39**

*Retrospectiva y Retroalimentación del Sprint no. 5.*

<b>¿Que salió bien en el desarrollo del sprint?</b>	<b>¿Que salió mal en el desarrollo del sprint?</b>	<b>¿Qué mejoras se van a implementar?</b>
Correcta generación de una factura.	Error al autorizar una nota de crédito (El consumidor final no puede generar notas de crédito)	Corrección en la creación de notas de crédito
Correcto envío de una factura.		Visualización del RIDE de los comprobantes generados, enviados y autorizados.
Correcta autorización de una factura.		

Correcta generación de una nota de crédito.

Correcto envío de una nota de crédito.

Envío de los comprobantes a los clientes.

---

*Nota.* Esta tabla muestra información resultante de la reunión realizada para la revisión del sprint no. 5. Fuente: Elaboración propia.

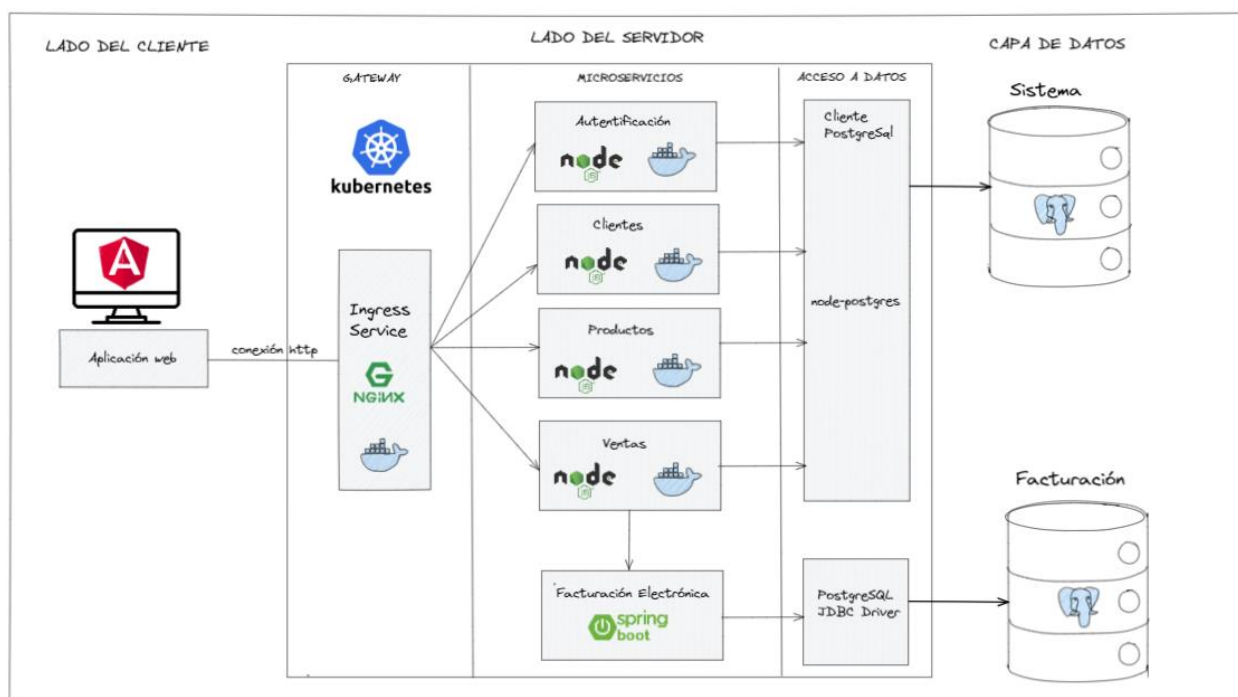
### **Arquitectura del Sistema**

Los usuarios o beneficiarios del sistema web podrán acceder al mismo, mediante conexión a internet; de esta manera, desde el lugar de trabajo el usuario podrá realizar las tareas correspondientes del sistema, alojada en un servidor que procesará las mismas. La implementación tecnológica consta de dos secciones; el primero se basa en el lado del cliente o la parte del frontend que corresponde a las diferentes partes que interactúan con el usuario; la segunda sección se basa en el lado del servidor o la parte de backend, esta parte consta de cinco módulos que serán los diferentes microservicios como, Autenticación, Clientes, Productos, Ventas y Facturación.

Para un mejor entendimiento del proceso de funcionamiento de los microservicios implementados, en la figura 2, se presenta la arquitectura implementada.

**Figura 2**

*Arquitectura del sistema web*



*Nota.* La figura presenta la arquitectura utilizada para el funcionamiento del sistema web.

Fuente: Elaboración propia.

El manejo y administración de los microservicios es por medio de Kubernetes, un orquestador de contenedores. Cada microservicio es ejecutado en su propio pod exponiendo un puerto en específico y asociado a una política con el uso de un servicio, en específico uno de tipo Clúster IP. Para lograr la comunicación de las APIs con el cliente es necesario el desarrollo de un gateway que redirija las peticiones a cada micro servicio según corresponda, para lo cual se utiliza una herramienta en Kubernetes que se denomina Ingress.

Un Ingress en Kubernetes es un recurso que permite controlar el acceso a los servicios dentro de un clúster de Kubernetes a través de la configuración de reglas de enrutamiento. Estas reglas especifican cómo los clientes externos deben conectarse a los servicios dentro del clúster, según su sitio oficial (<https://kubernetes.io/>). El servicio Ingress desplegado se basa en

una imagen del servidor web Nginx, uno de los más utilizados para el enrutamiento, redirección y balanceo de carga.

Para una comunicación segura entre el cliente y los microservicios se utiliza un add-on de Kubernetes conocido como Cert-manager que facilita la gestión de certificados SSL/TLS en un clúster. Permite automatizar el proceso de solicitud, renovación y revocación de certificados, proporcionando una configuración segura y fácil de usar. Cert-manager funciona con proveedores de certificados externos como Let's Encrypt o CA internos (<https://cert-manager.io/>).

### ***Dependencias de desarrollo***

En el desarrollo de microservicios con Node JS se han utilizado diversas dependencias, es decir, paquetes de software externos que pueden utilizarse para mejorar la funcionalidad y la seguridad de los servicios, entre las que cabe destacar las siguientes.

**Jsonwebtoken.** Es un paquete que permite trabajar con JWT (JSON Web Tokens), que pueden utilizarse para crear, firmar y validar tokens para autenticar y autorizar el uso de APIs. De esta forma, se pueden proteger las rutas definiendo un perfil de usuario y restringiendo el acceso a las mismas, siguiendo

una de las recomendaciones de la categoría A01:2021 - Pérdida de Control de Acceso del Top 10 2021 de OWASP.

**Bcryptjs.** Se trata de un paquete que permite cifrar contraseñas y es unidireccional, lo que significa que una vez cifrada una contraseña, no se puede descifrar. En su lugar, se realiza una comparación con una nueva entrada para comprobar que son iguales, tal y como se especifica en la categoría A02:2021 - Fallas criptográficas y como lo sugiere la categoría A07:2021 – Fallas de Identificación y Autenticación.

**express-validator.** Es un paquete Node.js utilizado para validar datos en una aplicación Express. Proporciona un conjunto de herramientas para la validación de tipos de datos, validación de longitudes y validación de expresiones regulares, añadiendo una capa extra de seguridad a los servicios y reduciendo las vulnerabilidades de categoría A03:2021 - Inyección.

**body-parser.** Es un paquete utilizado para analizar los datos enviados en el cuerpo de una petición HTTP, en el que se puede configurar el tamaño y el número de parámetros que recibe la petición, además de definir el tipo de medio que analiza el middleware. De este modo, cada API tiene su propia configuración de middleware en función del tipo de datos y de la cantidad de datos que va a recibir, separando las capas de sistema y de red en función de los requisitos de exposición y protección, como se especifica en una de las recomendaciones de la categoría A04:2021 - Diseño inseguro.

**Helmet.** Contiene un conjunto de cabeceras HTTP que pueden configurarse en una aplicación para protegerla de varios tipos de ataques de seguridad. Estos incluyen la protección contra la inyección de scripts XSS, la prevención del clickjacking y la protección contra ataques de content sniffing siguiendo una de las recomendaciones de la categoría A05:2021 - Configuración de seguridad incorrecta.

**express-rate-limit.** Se utiliza para limitar el número de solicitudes que una aplicación puede recibir de la misma fuente en un periodo de tiempo determinado. Esto es útil para evitar ataques de denegación de servicio (DoS) y proteger el rendimiento y la disponibilidad de una aplicación. Es una forma de prevenir las vulnerabilidades de la categoría A07:2021 - Fallos de identificación y autenticación.

**node-postgres.** Es una colección de módulos node.js para interactuar con una base de datos PostgreSQL. Soporta callbacks, promesas, async/wait, connection pooling, prepared statements, cursores, streaming de resultados y bindings C/C++, entre otros. Utiliza

transacciones cuando se realiza más de una operación en la base de datos para evitar cambios en caso de error. También dispone de consultas parametrizadas para reducir las vulnerabilidades en la categoría A03:2021 - Inyección.

### **Base de datos**

El gestor de base de datos utilizado es PostgreSQL, en el cual se utiliza una base de datos conformada por un total de 15 tablas, encargada de guardar toda la información que se envía desde el frontend y otra base de datos encargada exclusivamente de la facturación electrónica y otros comprobantes emitidos por el SRI.

Para visualizar el diseño completo de la base de datos, refiérase al Anexo 1.

### **Backend**

Los servicios web levantados se presentan como el punto central para el buen funcionamiento del sistema web, los cuales por medio de una API REST que contenga los servicios RESTful y con ello cubrir las diferentes necesidades identificadas. De esta manera brindar la comunicación entre la instancia de alojamiento del sistema y los módulos correspondientes que son gestionados por el usuario.

En las tablas 40, 41, 42, 43 y 44, se indican los detalles de los servicios levantados en los microservicios respectivos para el funcionamiento del sistema web.

### **Tabla 40**

#### *Servicios RESTful de Microservicio Autenticación*

<b>ID</b>	<b>Servicio</b>	<b>URL</b>	<b>Tipo</b>	<b>Resultado</b>
SR1-01	Obtener información de todos los usuarios de tipo empleado de un establecimiento.	/api/auth/auth/v1/empleados/:idEstb	GET	Status: 200 ok



SR1-02	Obtener información de un usuario mediante el ID.	/api/auth/auth/v1/usuario/:idUsuario	GET	Status: 200 ok
SR1-03	Envía las credenciales de autenticación del usuario y retorna un token.	/api/auth/auth/v1/login	POST	Status: 200 ok
SR1-04	Registra un nuevo usuario de tipo empleado.	/api/auth/auth/v1/registro	POST	Status: 200 ok
SR1-05	Actualizar la información de un usuario.	/api/auth/auth/v1/editar	PUT	Status: 200 ok
SR1-06	Envía un correo con un código de verificación para la recuperación de la contraseña.	/api/auth/auth/v1/recuperacion	PUT	Status: 200 ok
SR1-07	Actualizar la contraseña de un usuario.	/api/auth/auth/v1/actualizar	PUT	Status: 200 ok
SR1-08	Eliminar un usuario mediante el ID.	/api/auth/auth/v1/eliminar/:idUsuario	DELETE	Status: 200 ok
SR1-09	Obtener información de las deudas de un usuario mediante el ID y un estado.	/api/auth/deudas/v1/listar/:idUsuario/:estado	GET	Status: 200 ok
SR1-10	Obtener información de una deuda mediante el ID.	/api/auth/deudas/v1/:idDeuda	GET	Status: 200 ok

SR1-11	Envía un correo al usuario notificando el pago próximo de una deuda.	/api/auth/deudas/v1/notificacion/:idUsuario/:idDeuda	GET	Status: 200 ok
SR1-12	Guardar información de una nueva deuda	/api/auth/deudas/v1/registrar	POST	Status: 200 ok
SR1-13	Actualizar la información de una deuda.	/api/auth/deudas/v1/editar	PUT	Status: 200 ok
SR1-14	Eliminar una deuda mediante el ID.	/api/auth/deudas/v1/eliminar/:idDeuda	DELETE	Status: 200 ok
SR1-15	Obtener información de las categorías de las deudas.	/api/auth/categoriadeudas/v1/listar	GET	Status: 200 ok
SR1-16	Obtener información de una categoría mediante el ID.	/api/auth/categoriadeudas/v1/:idCategoria	GET	Status: 200 ok
SR1-17	Guardar información de una categoría deuda.	/api/auth/categoriadeudas/v1/registrar	POST	Status: 200 ok
SR1-18	Actualizar información de una categoría deuda.	/api/auth/categoriadeudas/v1/editar	PUT	Status: 200 ok
SR1-19	Eliminar una categoría deuda mediante el ID.	/api/auth/categoriadeudas/v1/eliminar/:idCategoria	DELETE	Status: 200 ok
SR1-20	Obtener información de los establecimientos.	/api/auth/establecimiento/v1/listar	GET	Status: 200 ok
SR1-21	Obtener información de un establecimiento mediante el ID.	/api/auth/establecimiento/v1/:idEstb	GET	Status: 200 ok

SR1-22	Guardar información de un establecimiento.	/api/auth/establecimiento/v1/registrar	POST	Status: 200 ok
SR1-23	Actualizar información de un establecimiento.	/api/auth/establecimiento/v1/editar	PUT	Status: 200 ok
SR1-24	Eliminar información de un establecimiento.	/api/auth/establecimiento/v1/eliminar/:idEstb	DELETE	Status: 200 ok

*Nota.* Esta tabla presenta la información de los servicios RESTful que forman parte del microservicio de Autenticación del sistema web. Fuente: Elaboración propia.

**Tabla 41**

*Servicios RESTful del Microservicio Clientes*

ID	Servicio	URL	Tipo	Resultado
SR2-25	Obtener información de los clientes de un establecimiento mediante el ID	/api/cliente/v1/listar/:idEstb	GET	Status: 200 ok
SR2-26	Obtener información de un cliente	/api/cliente/v1/:idCliente	GET	Status: 200 ok
SR2-27	Obtener información de los clientes frecuentes	/api/cliente/v1/listar/:idEstb/frecuentes	GET	Status: 200 ok
SR2-28	Obtener información de un cliente por búsqueda mediante el ID	/api/cliente/v1/Busqueda/:idEstb/:busqueda	GET	Status: 200 ok
SR2-29	Guardar información de un nuevo cliente	/api/cliente/v1/registrar	POST	Status: 200 ok
SR2-30	Actualizar la información de un cliente	/api/cliente/v1/editar	PUT	Status: 200 ok

SR2-31	Eliminar información de un cliente mediante el ID	/api/cliente/v1/eliminar/:id Cliente	DELETE	Status: 200 ok
--------	---	---	--------	-------------------

*Nota.* Esta tabla presenta la información de los servicios RESTful que forman parte del microservicio de Clientes del sistema web. Fuente: Elaboración propia.

**Tabla 42**

*Servicios RESTful del Microservicio Productos*

ID	Servicio	URL	Tipo	Resultado
SR3-32	Obtener información de bodega según el establecimiento mediante el ID	/api/bodega/v1/listar/:idEstb	GET	Status: 200 ok
SR3-33	Obtener información de una bodega mediante el ID	/api/bodega/v1/:idBodega	GET	Status: 200 ok
SR3-34	Guardar información de una nueva bodega	/api/bodega/v1/registrar	POST	Status: 200 ok
SR3-35	Actualizar la información de una bodega	/api/bodega/v1/editar	PUT	Status: 200 ok
SR3-36	Eliminar información de una bodega mediante el ID	/api/bodega/v1/eliminar/:idBodega	DELETE	Status: 200 ok
SR3-37	Obtener información de categorías de productos	/api/categoriaproducto/v1/listar	GET	Status: 200 ok
SR3-38	Obtener información de una categoría de producto mediante el ID	/api/categoriaproducto/listar/:idCatProducto	GET	Status: 200 ok
SR3-39	Guardar la información de una nueva categoría de producto	/api/categoriaproducto/registrar	POST	Status: 200 ok
SR3-40	Actualizar la información de una categoría de producto	/api/categoriaproducto/editar	PUT	Status: 200 ok
SR3-41	Eliminar información de una categoría de producto mediante el ID	/api/categoriaproducto/eliminar/:idCatProducto	DELETE	Status: 200 ok

SR3-42	Obtener información de productos según el establecimiento	/api/producto/v1/listarE/:idEstb	GET	Status: 200 ok
SR3-43	Obtener información de un producto mediante el ID	/api/producto/v1/listar/:idProducto	GET	Status: 200 ok
SR3-44	Obtener la información de productos más vendidos	/api/producto/v1/listar/:idEstb/vendidos	GET	Status: 200 ok
SR3-45	Obtener información de productos según el proveedor	/api/producto/v1/listarProveedor/:idProveed	GET	Status: 200 ok
SR3-46	Obtener información de productos por búsqueda	/api/producto/v1/Busqueda/:idEstb:busqueda	GET	Status: 200 ok
SR3-47	Guardar la información de un nuevo producto	/api/producto/v1/registrar	POST	Status: 200 ok
SR3-48	Actualizar la información de un nuevo producto	/api/producto/v1/editar	PUT	Status: 200 ok
SR3-49	Eliminar la información de un producto mediante el ID	/api/producto/v1/eliminar/:idProducto	DELETE	Status: 200 ok
SR3-50	Obtener información del inventario según la bodega	/api/inventario/v1/:idBodega	GET	Status: 200 ok
SR3-51	Obtener información de un inventario según el ID	/api/inventario/v1/listar/:idInventario	GET	Status: 200 ok
SR3-52	Guardar Información de un nuevo inventario	/api/inventario/v1/registrar	POST	Status: 200 ok
SR3-53	Actualizar información de un inventario	/api/inventario/v1/editar	PUT	Status: 200 ok
SR3-54	Eliminar la información de un inventario mediante el ID	/api/inventario/v1/eliminar/:idInventario	DELETE	Status: 200 ok
SR3-55	Obtener la información de los pedidos según el establecimiento	/api/pedidos/v1/:idEstb	GET	Status: 200 ok
SR3-56	Obtener la información de un pedido mediante el ID	/api/pedidos/v1/listar/:idPedido	GET	Status: 200 ok

SR3-57	Obtener la información de los pedidos según el estado	/api/pedidos/v1/listarEstado/:estado/:idEstb	GET	Status: 200 ok
SR3-58	Obtener información de las compras por meses	/api/pedidos/v1/:idEstb/meses	GET	Status: 200 ok
SR3-59	Guardar información de un nuevo pedido	/api/pedidos/v1/registrar	POST	Status: 200 ok
SR3-60	Actualizar la información de un pedido	/api/pedidos/v1/editar	PUT	Status: 200 ok
SR3-61	Cambiar el estado de un pedido	/api/pedidos/v1/cambiarEstado	PUT	Status: 200 ok
SR3-62	Eliminar la información de un pedido mediante el ID	/api/pedidos/v1/eliminar/:idPedido	DELETE	Status: 200 ok
SR3-63	Obtener información de los proveedores	/api/proveedor/v1/listar	GET	Status: 200 ok
SR3-64	Obtener información de un proveedor mediante el ID	/api/proveedor/v1/listar/:idProveedor	GET	Status: 200 ok
SR3-65	Guardar información de un nuevo proveedor	/api/proveedor/v1/registrar"	POST	Status: 200 ok
SR3-66	Actualizar información de un proveedor	/api/proveedor/v1/editar	PUT	Status: 200 ok
SR3-67	Eliminar la información de un proveedor mediante el ID	/api/proveedor/v1/eliminar/:idProveedor	DELETE	Status: 200 ok

*Nota.* Esta tabla presenta la información de los servicios RESTful que forman parte del microservicio de Productos del sistema web. Fuente: Elaboración propia.

### Tabla 43

#### *Servicios RESTful del Microservicio Ventas*

ID	Servicio	URL	Tipo	Resultado
SR4-68	Obtener información de las ventas según el	/api/venta/v1/listar/:idEstb/:estado/:tipo	GET	Status: 200 ok

---

	establecimiento, estado y tipo			
SR4-69	Obtener información de una venta mediante el ID	/api/venta/v1/:idVenta	GET	Status: 200 ok
SR4-70	Obtener la información del total de ventas según el establecimiento	/api/venta/v1/listar/:idEstb	GET	Status: 200 ok
SR4-71	Obtener la información de las ventas del mes	/api/venta/v1/listar/:idEstb/meses	GET	Status: 200 ok
SR4-72	Obtener la información de las ventas por mes	/api/venta/v1/fechas/:fechaInicio/:fechaFin/:idEstb	GET	Status: 200 ok
SR4-73	Obtener la información de las ventas según el secuencial, tipo, estado y establecimiento	/api/venta/v1/ventas/:secuencial/:tipo/:estado/:idEstb	GET	Status: 200 ok
SR4-74	Obtener la información de las ventas según la caja	/api/venta/v1/caja/:idCaja	GET	Status: 200 ok
SR4-75	Guardar información de una nueva venta	/api/venta/v1/regar	POST	Status: 200 ok
SR4-76	Actualizar la información de una venta	/api/venta/v1/editar	PUT	Status: 200 ok
SR4-77	Eliminar la información de una venta mediante el ID	/api/venta/v1/eliminar/:idVenta	DELETE	Status: 200 ok
SR4-78	Actualizar la información de un tipo de venta	/api/venta/v1/editar-tipo	PUT	Status: 200 ok

SR4-79	Reporte de ventas en excel	/api/venta/v1/reporte/excel	POST	Status: 200 ok
SR4-80	Reporte de ventas en pdf	/api/venta/v1/reporte/pdf	POST	Status: 200 ok
SR4-81	Obtener la información de caja según el establecimiento	/api/caja/v1/listar/:idEstb	GET	Status: 200 ok
SR4-82	Obtener la información de una caja mediante el ID	/api/caja/v1/:idCaja	GET	Status: 200 ok
SR4-83	Obtener la información de cajas según la fecha mediante la fecha de inicio, fin y establecimiento	/api/caja/v1/fechas/:fechaInicio/:fechaFin/:idEstb	GET	Status: 200 ok
SR4-84	Guardar la información de una nueva caja	/api/caja/v1/registrar	POST	Status: 200 ok
SR4-85	Actualizar la información de una caja	/api/caja/v1/editar	PUT	Status: 200 ok
SR4-86	Cerrar una caja	/api/caja/v1/cerrar	PUT	Status: 200 ok
SR4-87	Abrir una caja	/api/caja/v1/abrir	PUT	Status: 200 ok
SR4-88	Actualizar la información de ventas de caja	/api/caja/v1/actualizar	PUT	Status: 200 ok
SR4-89	Obtener la información de detalles	/api/detalle/v1/listar	GET	Status: 200 ok



SR4-90	Obtener la información de un detalle mediante el ID	/api/detalle/v1/:idDetalle	GET	Status: 200 ok
SR4-91	Obtener la información de detalles según la venta	/api/detalle/v1/ventas/:idVenta	GET	Status: 200 ok
SR4-92	Guardar información de un nuevo detalle	/api/detalle/v1/regar	POST	Status: 200 ok
SR4-93	Actualizar la información de un detalle	/api/detalle/v1/editar	PUT	Status: 200 ok
SR4-94	Eliminar la información de un detalle mediante el ID	/api/detalle/v1/eliminar/:idDetalle	DELETE	Status: 200 ok
SR4-95	Generar Json de factura según la venta mediante el ID	/api/comprobantes/v1/jsomfatura/:idVenta	GET	Status: 200 ok
SR4-96	Eliminar factura mediante el ID de venta	/api/comprobantes/v1/eliminar/:idVenta	PUT	Status: 200 ok
SR4-97	Enviar factura de la venta mediante el ID	/api/comprobantes/v1/enviar/:idVenta	PUT	Status: 200 ok
SR4-98	Autorizar factura de la venta mediante el ID	/api/comprobantes/v1/autorizar/:idVenta	PUT	Status: 200 ok
SR4-99	Ver el pdf de la factura mediante el ID	/api/comprobantes/v1/pdf/:idVenta	GET	Status: 200 ok
SR4-100	Eliminar el pdf de la factura mediante el ID	/api/comprobantes/v1/pdf/:idVenta/eliminar	GET	Status: 200 ok

---

*Nota.* Esta tabla presenta la información de los servicios RESTful que forman parte del microservicio de Ventas del sistema web. Fuente: Elaboración propia.

**Tabla 44***RESTful del Microservicio Facturación*

<b>ID</b>	<b>Servicio</b>	<b>URL</b>	<b>Tipo</b>	<b>Resultado</b>
SR5-101	Generar token de autenticación para utilizar las APIs de facturación.	/veronica/oauth/token	POST	Status: 200 ok
SR5-102	Guardar firma electrónica.	/veronica/operaciones/certificados-digitales/empresas/:numeroRuc	POST	Status: 200 ok
SR5-103	Guardar información de la empresa.	/veronica/operaciones/empresa	POST	Status: 200 ok
SR5-104	Obtener información de una empresa mediante el RUC.	/veronica/operaciones/empresa/:identificacion-empresa	GET	Status: 200 ok
SR5-105	Generar xml firmado de la factura electrónica.	/veronica/api/v1.0/facturas	POST	Status: 200 ok
SR5-106	Enviar factura electrónica al SRI.	/veronica/api/v1.0/facturas/:clave-acceso/enviar	PUT	Status: 200 ok
SR5-107	Autorizar factura electrónica mediante la clave de acceso.	/veronica/api/v1.0/facturas/:clave-acceso/autorizar	PUT	Status: 200 ok
SR5-108	Eliminar factura electrónica mediante la clave de acceso.	/veronica/api/v1.0/facturas/:clave-acceso	DELETE	Status: 200 ok

SR5-109	Obtener archivo PDF de una factura electrónica mediante la clave de acceso.	/veronica/api/v1.0/facturas/:clave-acceso/archivos/pdf	GET	Status: 200 ok
SR5-110	Generar XML firmado de la nota de crédito.	/veronica/api/v1.0/notas-credito	POST	Status: 200 ok
SR5-111	Enviar nota de crédito al SRI.	/veronica/api/v1.0/notas-credito/:clave-acceso/enviar	PUT	Status: 200 ok
SR5-112	Autorizar nota de crédito mediante la clave de acceso.	/veronica/api/v1.0/notas-credito/:clave-acceso/autorizar	PUT	Status: 200 ok
SR5-113	Eliminar nota de crédito mediante la clave de acceso.	/veronica/api/v1.0/notas-credito/:clave-acceso	DELETE	Status: 200 ok
SR5-114	Obtener archivo PDF de una nota de crédito mediante la clave de acceso.	/veronica/api/v1.0/notas-credito/:clave-acceso/archivos/pdf	GET	Status: 200 ok

---

*Nota.* Esta tabla presenta la información de los servicios RESTful que forman parte del microservicio de Facturación del sistema web. Fuente: Elaboración propia.

## Implementación

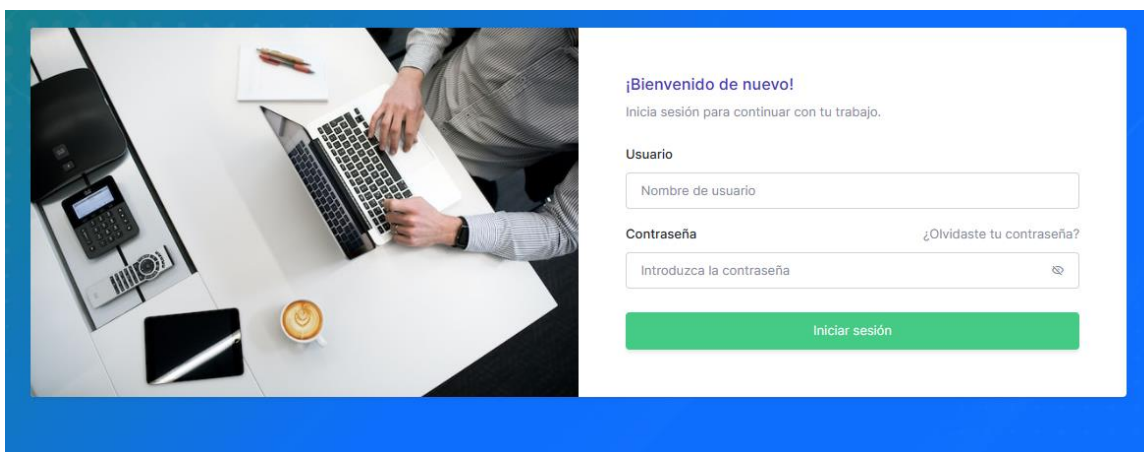
### **Resultado del Sprint 1**

**Modulo para la Gestión y Autenticación de Usuarios.** Esta es la parte del sistema que gestiona el ingreso al mismo. El usuario administrador entregará al usuario gerente el nombre de usuario y contraseña respectiva para que le permita el ingreso. Así mismo podrá hacer uso de las opciones como, cambiar contraseña y recuperar contraseña.

En la Figura 3 se muestra la interfaz para el sistema de login del usuario en el cual podrá ingresar al sistema. Por otra parte, la figura 4 se muestra la interfaz para recuperar la contraseña de acceso en caso de olvido, en el cual se debe ingresar el correo electrónico para recibir el código de recuperación.

### Figura 3

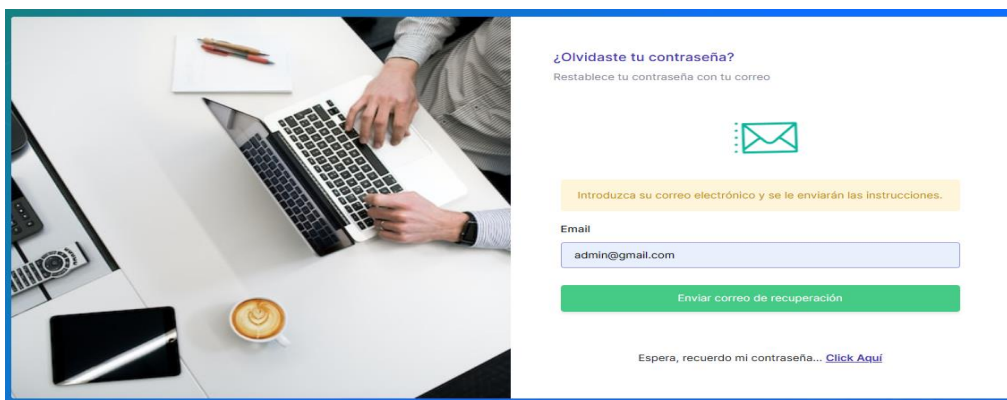
*Interfaz gráfica del login del sistema*



*Nota.* La figura presenta del módulo de autenticación del sistema la interfaz del login de usuario del sistema. Fuente: Elaboración propia.

### Figura 4

*Interfaz gráfica para la recuperación de contraseña*

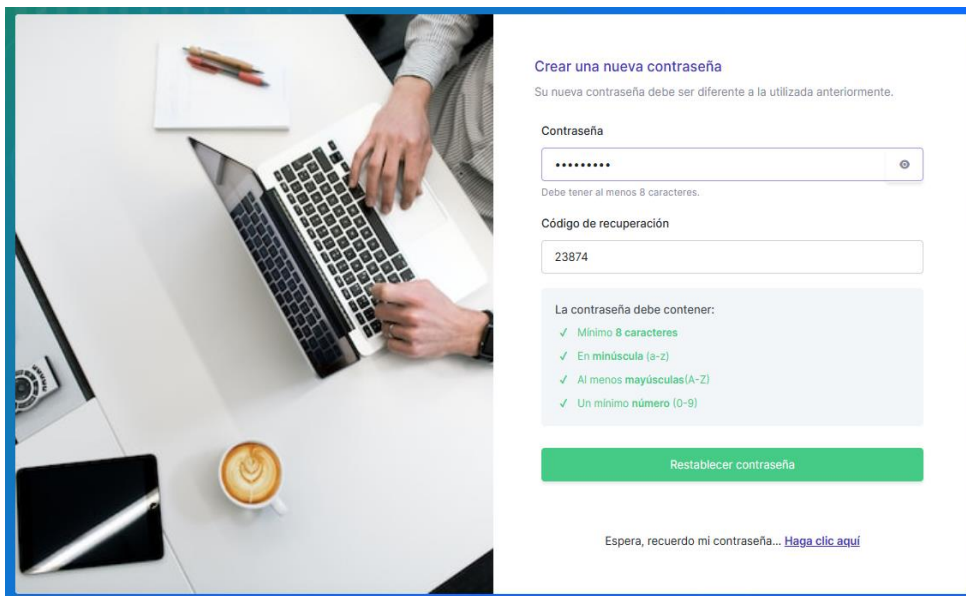


*Nota.* La figura presenta la interfaz del módulo de autenticación del sistema para la recuperación de contraseña del usuario. Fuente: Elaboración propia.

Posteriormente al ingreso del correo para recibir el código de recuperación de contraseña, en la figura 5 se indica lo que el sistema mostrará, la interfaz correspondiente para ingresar los datos que solicita y completar la acción de recuperación. Se revisa la bandeja de entrada del correo ingresado y verificar que se recibió el código solicitado, tal como se indica en la figura 6. De esta manera poder restablecer la contraseña del usuario correspondiente.

## Figura 5

*Interfaz gráfica para crear nueva contraseña*



*Nota.* La figura presenta la interfaz para crear la nueva contraseña. Fuente: Elaboración propia.

## Figura 6

### Correo del código de recuperación



*Nota.* La figura presenta el correo electrónico del código de recuperación de contraseña.

Fuente: Elaboración propia.

Las figuras 7, 8 y 9, son interfaces del apartado del control del usuario, en el cual podrá realizar las diferentes acciones como registrar, listar las deudas pendientes y las deudas canceladas con la información correspondiente.

## Figura 7

Interfaz gráfica para el registro de deudas.

**Añadir Deuda**

Entidad  
Banco de Guayaquil

Responsable  
Ferretería

Categoría  
Bancos

Día de Pago  
12

Plazo  
12

Total a Pagar  
500

Cerrar Añadir Deuda

*Nota.* La figura presenta la interfaz para registrar una deuda. Fuente: Elaboración propia.

## Figura 8

Interfaz gráfica para listar las deudas pendientes.

DEUDAS Deudas > Usuarios

Q Buscar... + Agregar Categoría + Agregar Deuda

Pendientes Canceladas

#	Entidad	Día de Pago	Responsable	Pagos	Total	Acciones
1	Banco del Pacífico	14	Ferretería	12	5200	Editar Eliminar
2	Banco de Guayaquil	21	Ferretería	12	500	Editar Eliminar

Mostrando 2 resultados totales Anteriores Siguientes

*Nota.* La figura presenta la interfaz para visualizar las deudas pendientes. Fuente: Elaboración propia.

## Figura 9

Interfaz gráfica para listar las deudas canceladas.

#	Entidad	Día de Pago	Responsable	Pagos	Total	Acciones
1	Banco del Pichincha	3	Ferretería	10	200	<a href="#">Editar</a> <a href="#">Eliminar</a>

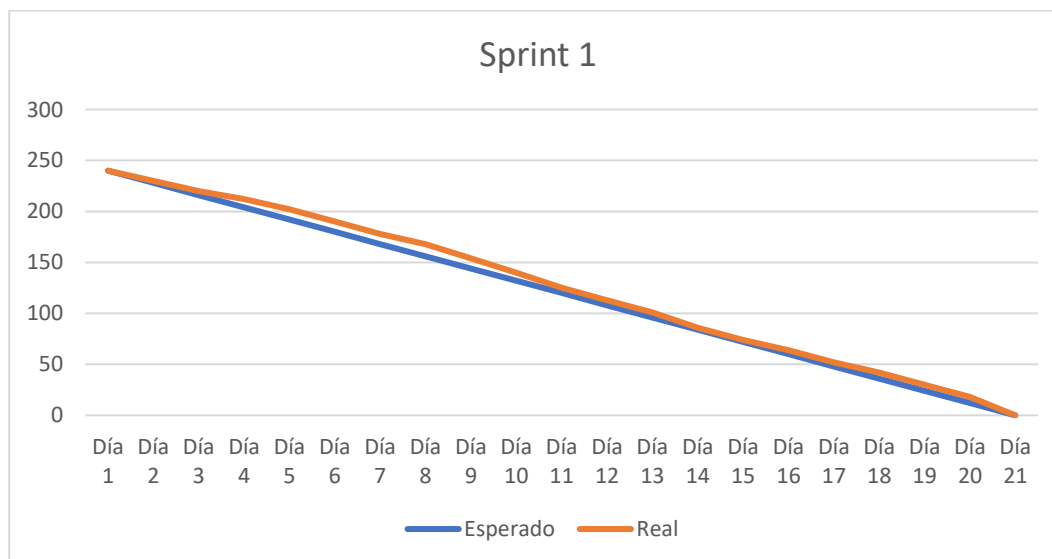
Mostrando 1 resultados totales [Anteriores](#)

*Nota.* La figura presenta la interfaz para visualizar las deudas canceladas. Fuente: Elaboración propia.

**Burndown Chart Sprint 1.** En la realización del presente sprint se presentaron pocos inconvenientes que no influenciaron en la tendencia del tiempo esperado y tiempo real, tal como se indica en la figura 10. Los pequeños retrasos presentados son por el tiempo dedicado a investigación de cómo mejorar la seguridad de los servicios. De esta forma se llegó a culminar y cerrar el ciclo de desarrollo de las diferentes tareas presentadas en el rango de fechas establecidas.

## Figura 10

*Burndown Chart Sprint 1.*





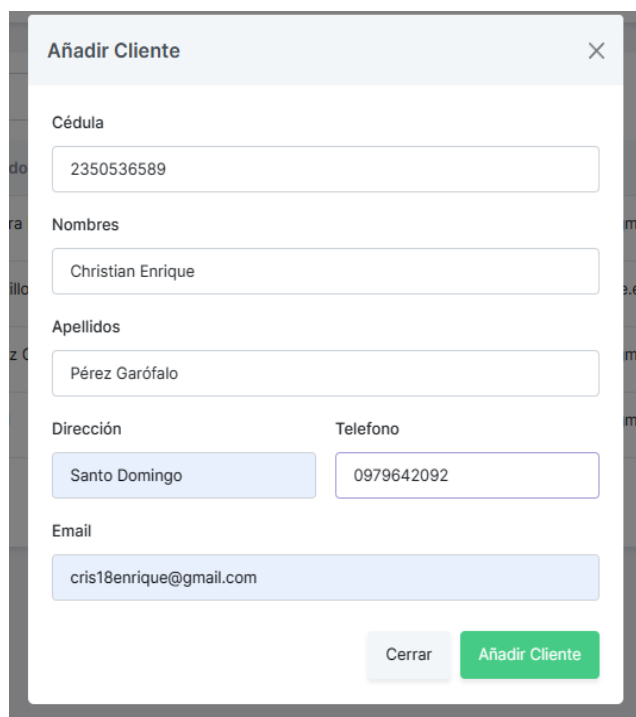
*Nota.* El gráfico presenta el trabajo realizado en comparación a la planificación esperada en función del tiempo real para completar el sprint. Fuente: Elaboración propia.

### **Resultado del Sprint 2**

**Módulo para la gestión de clientes.** En esta parte el sistema permite gestionar la información de los clientes, se enlistan todos los clientes registrados de manera que se visualicen los datos correspondientes, así mismo con las acciones de actualizar la información o eliminar la misma. En las figuras 11, 12, 13 y 14 son interfaces del módulo para la gestión de clientes.

#### **Figura 11**

*Interfaz gráfica para registrar clientes.*



The image shows a web form titled "Añadir Cliente" (Add Client) with a close button (X) in the top right corner. The form contains several input fields for client information:

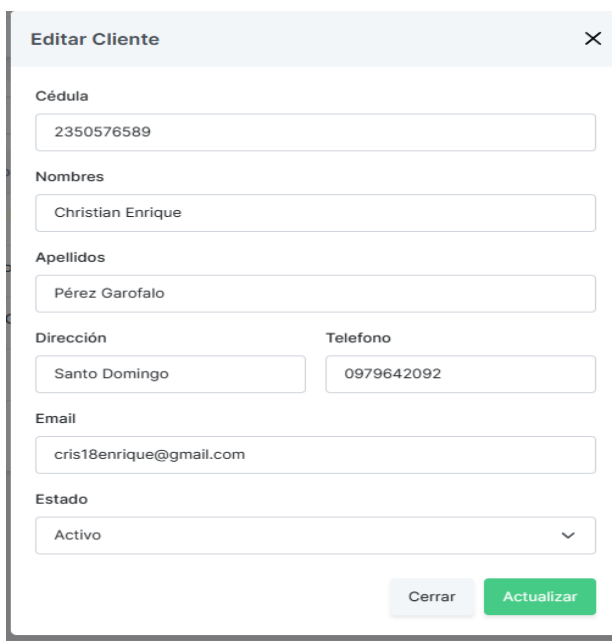
- Cédula:** A text input field containing the value "2350536589".
- Nombres:** A text input field containing the value "Christian Enrique".
- Apellidos:** A text input field containing the value "Pérez Garófalo".
- Dirección:** A text input field containing the value "Santo Domingo".
- Telefono:** A text input field containing the value "0979642092".
- Email:** A text input field containing the value "cris18enrique@gmail.com".

At the bottom right of the form, there are two buttons: a grey "Cerrar" (Close) button and a green "Añadir Cliente" (Add Client) button.

*Nota.* La figura presenta la interfaz para el registro de un nuevo cliente. Fuente: Elaboración propia.

## Figura 12

Interfaz gráfica para actualizar información del cliente.



Editar Cliente

Cédula  
2350576589

Nombres  
Christian Enrique

Apellidos  
Pérez Garofalo

Dirección  
Santo Domingo

Telefono  
0979642092

Email  
cris18enrique@gmail.com

Estado  
Activo

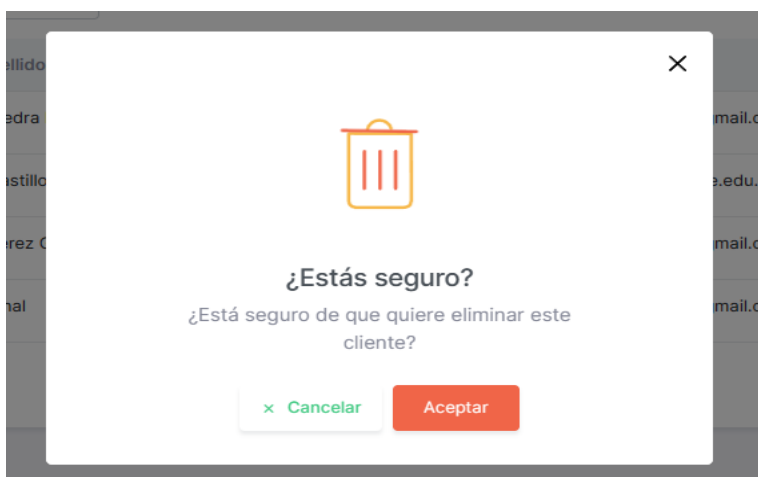
Cerrar Actualizar

*Nota.* La figura presenta la interfaz para actualizar o editar la información del cliente registrado.

Fuente: Elaboración propia.

## Figura 13

Interfaz gráfica para eliminar el cliente.



¿Estás seguro?

¿Está seguro de que quiere eliminar este cliente?

Cancelar Aceptar

*Nota.* La figura presenta la interfaz para eliminar la información del cliente registrado. Fuente:

Elaboración propia.

## Figura 14

Interfaz gráfica para listar los clientes.

AGREGAR CLIENTES Clientes > Agregar Clientes

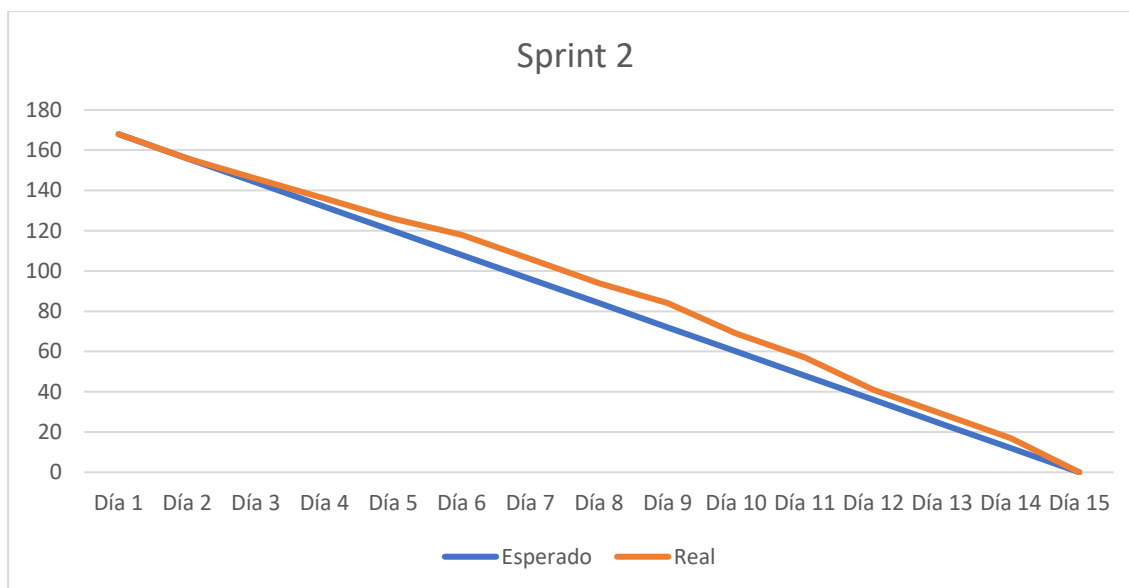
Q Buscar... + Añadir Cliente

#	Nombres	Apellidos	Identificación	Telefono	Email	Dirección	Estado	Acciones
1	Andrick	Piedra Lopez	1723578710	0985658745	cris18enrique@gmail.comt	Santo Domingo	ACTIVO	<a href="#">Editar</a> <a href="#">Eliminar</a>
2	Eddy Fernando	Castillo Cujilan	2300172547	0999459787	efcastillo2@espe.edu.ec	Santo Domingo	ACTIVO	<a href="#">Editar</a> <a href="#">Eliminar</a>
3	Christian Enrique	Pérez Garofalo	2350576589	0979642092	cris18enrique@gmail.com	Santo Domingo	ACTIVO	<a href="#">Editar</a> <a href="#">Eliminar</a>
4	Consumidor	Final	99999999999999	9999999999	cris18enrique@gmail.com	santo domingo	ACTIVO	<a href="#">Editar</a> <a href="#">Eliminar</a>

Mostrando 4 resultados totales [Anteriores](#) [Siguietes](#)

*Nota.* La figura presenta la interfaz para listar la información de los clientes registrados. Fuente: Elaboración propia.

**Burndown Chart Sprint 2.** En la realización del presente sprint se presentaron pocos inconvenientes ya que el tiempo planificado fue corto en comparación al resto y se realizaron algunas correcciones en la arquitectura de los servicios, aunque no influenciaron en la tendencia del tiempo planificado, tal como se indica en la figura 15. De esta forma se llegó a culminar y cerrar el ciclo de desarrollo de las diferentes tareas presentadas en el rango de fechas establecidas.

**Figura 15***Burndown Chart Sprint 2.*

*Nota.* El gráfico presenta el trabajo realizado en comparación a la planificación esperada en función del tiempo real para completar el sprint. Fuente: Elaboración propia.

### **Resultado del Sprint 3**

**Módulo para la Gestión de Inventario.** Esta es la parte del sistema que permite el registro, actualización de stock de productos y visualización de los mismos con sus respectivas categorías.

Las figuras 16, 17, 18 y 19 son interfaces del sub módulo para la gestión de productos. Así mismo las figuras 20, 21 y 22 son interfaces del sub módulo para la gestión de bodega, mientras que las figuras 23, 24 y 25 son sub módulo para la gestión del inventario.

## Figura 16

Interfaz gráfica para listar los productos.

*Nota.* La figura presenta la interfaz para listar la información de los productos registrados.

Fuente: Elaboración propia.

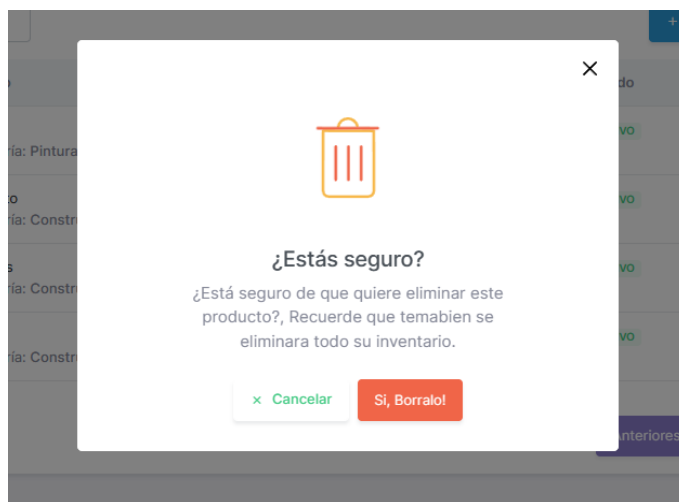
## Figura 17

Interfaz gráfica para actualizar la información de los productos.

*Nota.* La figura presenta la interfaz para actualizar o editar la información de los productos registrados. Fuente: Elaboración propia.

### Figura 18

*Interfaz gráfica para eliminar el registro del producto.*



*Nota.* La figura presenta la interfaz para eliminar la información de los productos registrados. Fuente: Elaboración propia.

### Figura 19

*Interfaz gráfica para listar los registros de productos*

PRODUCTOS								
#	Código	Producto	Marca	Costo	PvP	IVA	Estado	Acciones
1	001	pintura Categoría: Pinturas	pintuco	\$ 5	\$ 8	12	ACTIVO	
2	002	Cemento Categoría: Construccion	Holcim	\$ 5	\$ 15	0	ACTIVO	
3	003	Tornillos Categoría: Construccion	Philip	\$ 0.35	\$ 0.45	12	ACTIVO	
4	004	ducha Categoría: Construccion	duchas	\$ 5	\$ 7.68	12	ACTIVO	

Mostrando 4 resultados totales


Anteriores    Siguientes

*Nota.* La figura presenta la interfaz para listar la información de los productos registrados.

Fuente: Elaboración propia.

## Figura 20

*Interfaz gráfica para registrar la información de bodega.*



The screenshot shows a web application interface for adding a warehouse. At the top right, there is a breadcrumb trail: "Empresa > Bodegas". The main form is titled "Añadir Bodega" and contains three input fields: "Establecimiento" (Establishment) with a dropdown menu showing "Ferretería Electrocat", "Nombre" (Name) with a text input field containing "Bodega 1", and "Dirección" (Address) with a text input field containing "santo domingo". A green button labeled "Añadir Bodega" is positioned at the bottom right of the form.

*Nota.* La figura presenta la interfaz para registrar la información de bodega Fuente: Elaboración propia.

## Figura 21

*Interfaz gráfica para actualizar la información de bodega.*



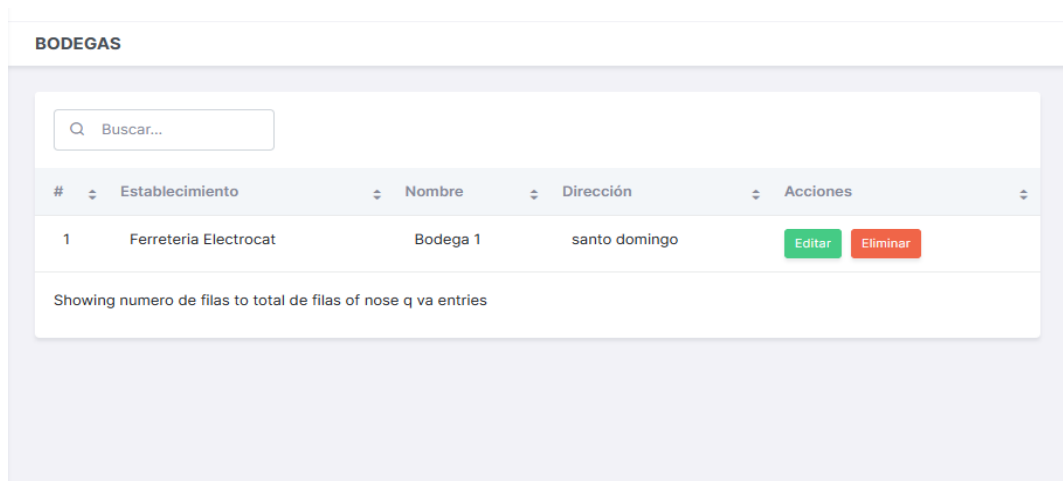
The screenshot shows a web application interface for editing a warehouse. The form is titled "Editar Bodega" and has a close button (X) in the top right corner. It contains three input fields: "Establecimiento" (Establishment) with a dropdown menu showing "Ferretería Electrocat", "Nombre" (Name) with a text input field containing "Bodega 1", and "Dirección" (Address) with a text input field containing "santo domingo". At the bottom right, there are two buttons: a grey "Cerrar" (Close) button and a green "Actualizar" (Update) button.

*Nota.* La figura presenta la interfaz para actualizar o editar la información de bodega Fuente:

Elaboración propia.

### Figura 22

*Interfaz gráfica para listar la información de bodega.*



*Nota.* La figura presenta la interfaz para listar la información de bodega con sus acciones correspondientes. Fuente: Elaboración propia.

### Figura 23

*Interfaz gráfica para añadir stock.*

The screenshot shows a modal form titled "Añadir Stock" with a close button (X) in the top right corner. The form contains the following fields:

- Bodega:** A dropdown menu with "Bodega 1" selected and a close button (X).
- Producto:** A dropdown menu with "Cemento" selected and a close button (X).
- Stock:** A text input field containing the value "15".
- Cantidad Mínima:** A text input field containing the value "5".

At the bottom of the form, there are two buttons: "Cerrar" (grey) and "Añadir Stock" (green).



*Nota.* La figura presenta la interfaz para registrar la información del stock. Fuente: Elaboración propia.

### Figura 24

*Interfaz gráfica para actualizar información del inventario.*



Editar Inventario - Cemento

Stock actual

8

Stock Mínimo

10

Cerrar Actualizar

*Nota.* La figura presenta la interfaz para actualizar la información del inventario. Fuente: Elaboración propia.

### Figura 25

*Interfaz gráfica para actualizar información del inventario.*

GESTIÓN DE INVENTARIO Inventario > Gestión de Inventario

Q. Buscar... Bodega 1 + Agregar Stock

#	Producto	Bodega	Stock	Stock Mínimo	Acciones
1	Cemento	Bodega 1	8	10	<span style="color: green;">Editar</span> <span style="color: red;">Eliminar</span>
2	ducha	Bodega 1	48	10	<span style="color: green;">Editar</span> <span style="color: red;">Eliminar</span>
3	pintura	Bodega 1	19	5	<span style="color: green;">Editar</span> <span style="color: red;">Eliminar</span>
4	Tornillos	Bodega 1	267	50	<span style="color: green;">Editar</span> <span style="color: red;">Eliminar</span>

Mostrando 4 resultados totales Anteriores Siguintes

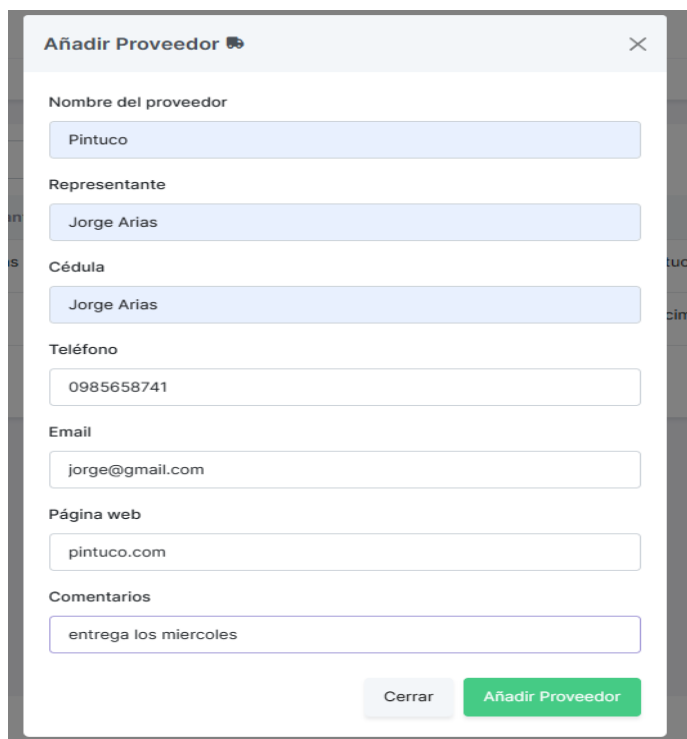
*Nota.* La figura presenta la interfaz para listar la información del inventario con sus acciones correspondientes. Fuente: Elaboración propia.

**Módulo para la gestión de proveedores y pedidos.** Esta es la parte del sistema que permite el registro, actualización y visualización de los proveedores, dependiendo del rol de usuario tendrá acceso a las diferentes acciones según corresponda.

Las figuras 26, 27 y 28 son las interfaces del módulo de gestión de proveedores para el usuario del sistema. Por otro lado, las figuras 29, 30, 31 y 32 son interfaces del apartado para la gestión de pedidos.

### Figura 26

*Interfaz gráfica para registrar proveedor.*



The image shows a web form titled "Añadir Proveedor" with a close button in the top right corner. The form contains several input fields with the following labels and values:

- Nombre del proveedor: Pintuco
- Representante: Jorge Arias
- Cédula: Jorge Arias
- Teléfono: 0985658741
- Email: jorge@gmail.com
- Página web: pintuco.com
- Comentarios: entrega los miercoles

At the bottom of the form, there are two buttons: "Cerrar" (grey) and "Añadir Proveedor" (green).

*Nota.* La figura presenta la interfaz para registrar la información del proveedor. Fuente:

Elaboración propia.

## Figura 27

Interfaz gráfica para actualizar la información del proveedor.

**Editar Proveedor** ✕

Nombre del proveedor

Representante

Cédula

Teléfono

Email

Página web

Comentarios

*Nota.* La figura presenta la interfaz para actualizar o editar la información del proveedor.

Fuente: Elaboración propia.

## Figura 28

Interfaz gráfica para listar los proveedores.

PROVEEDORES Compras > Proveedores

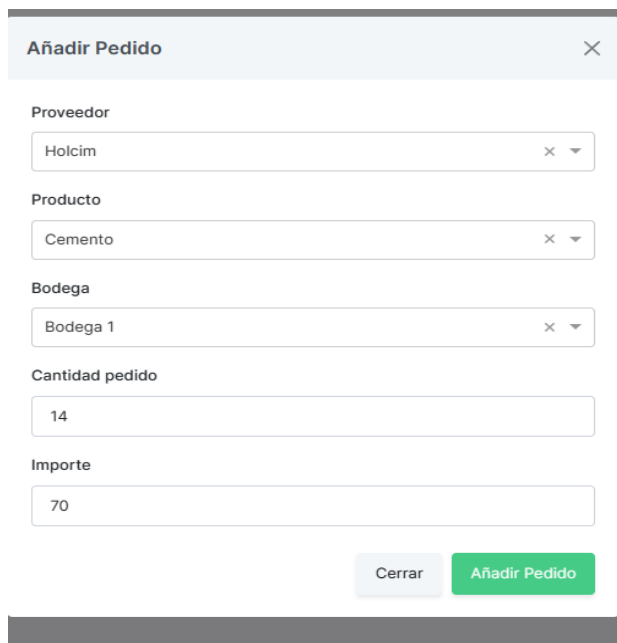
#	Proveedor	Representante	telefono	email	web	Comentarios	Acciones
1	Pintuco	Jorge Arias	098568745	admin@gmail.com	pintuco.com	entrega los miercoles	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
2	Holcim	Pedro	0985685474	ceperez4@espe.edu.ec	holcim.com	n/a	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

Mostrando 2 resultados totales

*Nota.* La figura presenta la interfaz para listar la información del proveedor con sus acciones correspondientes. Fuente: Elaboración propia.

**Figura 29**

*Interfaz gráfica para registro de pedidos.*



The screenshot shows a modal window titled "Añadir Pedido" with a close button (X) in the top right corner. The form contains the following fields:

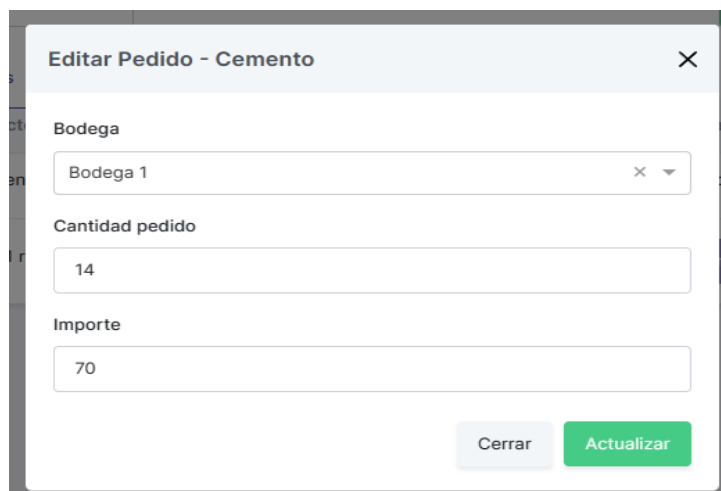
- Proveedor:** A dropdown menu with "Holcim" selected.
- Producto:** A dropdown menu with "Cemento" selected.
- Bodega:** A dropdown menu with "Bodega 1" selected.
- Cantidad pedido:** A text input field containing the number "14".
- Importe:** A text input field containing the number "70".

At the bottom right of the form, there are two buttons: "Cerrar" (Close) and "Añadir Pedido" (Add Order).

*Nota.* La figura presenta la interfaz para registrar la información del pedido. Fuente: Elaboración propia.

**Figura 30**

*Interfaz gráfica para actualizar la información de pedidos.*



The screenshot shows a modal window titled "Editar Pedido - Cemento" with a close button (X) in the top right corner. The form contains the following fields:



- Bodega:** A dropdown menu with "Bodega 1" selected.
- Cantidad pedido:** A text input field containing the number "14".
- Importe:** A text input field containing the number "70".

At the bottom right of the form, there are two buttons: "Cerrar" (Close) and "Actualizar" (Update).

*Nota.* La figura presenta la interfaz para actualizar o editar la información del pedido. Fuente: Elaboración propia.

### Figura 31

Interfaz gráfica para listar la información de pedidos ordenados.

#	Producto	Proveedor	Bodega	Cantidad	Importe	Fecha del Pedido	Estado	Acciones
1	Cemento	Holcim	Bodega 1	14	\$70	19/01/2023	Ordenado	 

Mostrando 1 resultados totales

[Anteriores](#) [Sigüientes](#)

*Nota.* La figura presenta la interfaz para listar la información registrada de los pedidos ordenados. Fuente: Elaboración propia.

### Figura 32

Interfaz gráfica para listar la información de pedidos entregados.

#	Producto	Proveedor	Bodega	Cantidad	Importe	Fecha del Pedido	Estado	Acciones
1	pintura	Pintuco	Bodega 1	25	\$125	11/01/2023	Entregado	<a href="#">Ordenado</a>
2	pintura	Pintuco	Bodega 1	10	\$50	13/01/2023	Entregado	<a href="#">Ordenado</a>

Mostrando 2 resultados totales

[Anteriores](#) [Sigüientes](#)

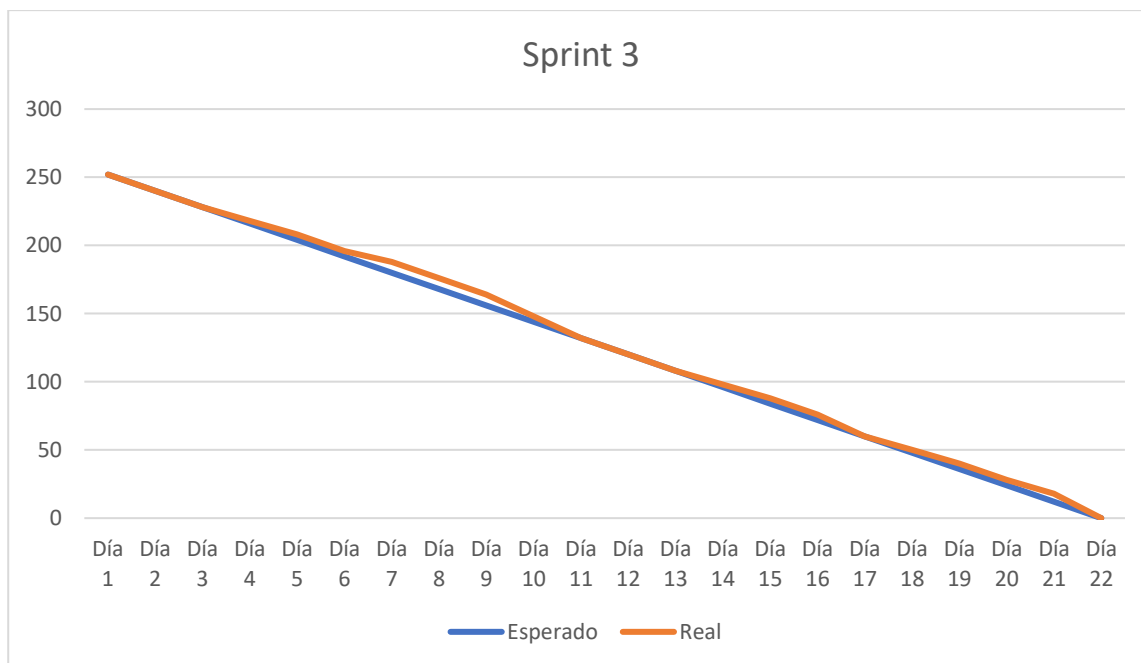
*Nota.* La figura presenta la interfaz para listar la información registrada de los pedidos entregados. Fuente: Elaboración propia.

**Burndown Chart Sprint 3.** La ejecución de este sprint presentó algunos inconvenientes en después del tercer día debido a procesos de manipulación e implementación en la gestión de los productos y su relación con el inventario, de esta manera cumplir con los requerimientos

planteados inicialmente. A pesar de aquello, y como se observa en la figura 33, el sprint se cumplió dentro de la fecha establecida, pero con un flujo de trabajo pausada en relación al tiempo esperado.

### Figura 33

*Burndown Chart Sprint 3.*



*Nota.* El gráfico presenta el trabajo realizado en comparación a la planificación esperada en función del tiempo real para completar el sprint 3. Fuente: Elaboración propia.

### **Resultado del Sprint 4**

**Módulo para la Gestión de Ventas.** En este apartado se enfoca en el control de las ventas, el sistema permite realizar la venta, visualizar y realizar los reportes correspondientes de las mismas. Las diferentes acciones en este apartado dependerán del rol de usuario que este haciendo uso del sistema, ya sea gerente o empleado.

En las figuras 34, 35 y 36 se puede observar la parte grafica implementada para el control de ventas, en este caso realizar la misma. Por otra parte, en la figura 37, 38 y 39 se observa los reportes de las ventas realizadas.

## Figura 34

Interfaz gráfica para realizar la venta.

**DATOS DE LA VENTA**

Ferretería Electrocat

Cooperativa Nuevo Amanecer, Av. Los Anturios

2350353187001

**DATOS DEL CLIENTE** +

🔍 Busca un cliente aquí ...

Consumidor Final

Santo Domingo

9999999999999999

9999999999

🔍 Busca un producto aquí ...

#	Detalles del producto	PVP (\$)	Cantidad	Descuento	Importe	Existencia	Acciones
1	004 - ducha	7,68	1	0	7,68	48	<span style="color: green;">Quitar</span>
2	002 - Cemento	15	1	0	15	8	<span style="color: green;">Quitar</span>

Sub Total 22,68

Sub Total IVA 0,92

Descuento 0,00

**Total 23,60**

📄 Guardar Venta

*Nota.* La figura presenta la interfaz para realizar una venta. Fuente: Elaboración propia.

## Figura 35

Interfaz gráfica para actualizar la venta.

18-01-2023 ← Volver

**DATOS DE LA VENTA**

Ferretería Electrocat

Cooperativa Nuevo Amanecer, Av. Los Anturios

2350353187001

**DATOS DEL CLIENTE**

🔍 Busca un cliente aquí ...

Consumidor Final

santo domingo

9999999999999999

9999999999

🔍 Busca un producto aquí ...

#	Detalles del producto	PVP (\$)	Cantidad	Descuento	Importe	Existencia	Acciones
1	002 - Cemento	15	1	0	15	8	<span style="color: green;">Quitar</span>

Sub Total 15

Sub Total IVA 0

Descuento 0

**Total 15**

← Cancelar 📄 Guardar Venta

*Nota.* La figura presenta la interfaz para actualizar o editar la información de una venta. Fuente: Elaboración propia.

### Figura 36

*Interfaz gráfica para listar las ventas.*



Q Buscar...

#	Cliente	fecha	Sub Total	Sub Total IVA	Descuento	Total	Estado	Acciones
1	Consumidor Final	18/01/2023	\$ 15	\$ 0	\$ 0	\$ 15	Venta	 
2	Consumidor Final	18/01/2023	\$ 45	\$ 0	\$ 0	\$ 45	Venta	 
3	Consumidor Final	17/01/2023	\$ 15.36	\$ 1.84	\$ 0	\$ 17.2	Venta	 
4	Consumidor Final	06/01/2023	\$ 15	\$ 0	\$ 0	\$ 15	Venta	 
5	Consumidor Final	04/01/2023	\$ 14	\$ 1.68	\$ 0	\$ 15.68	Venta	 

Mostrando 1 a 10 de 5 resultados

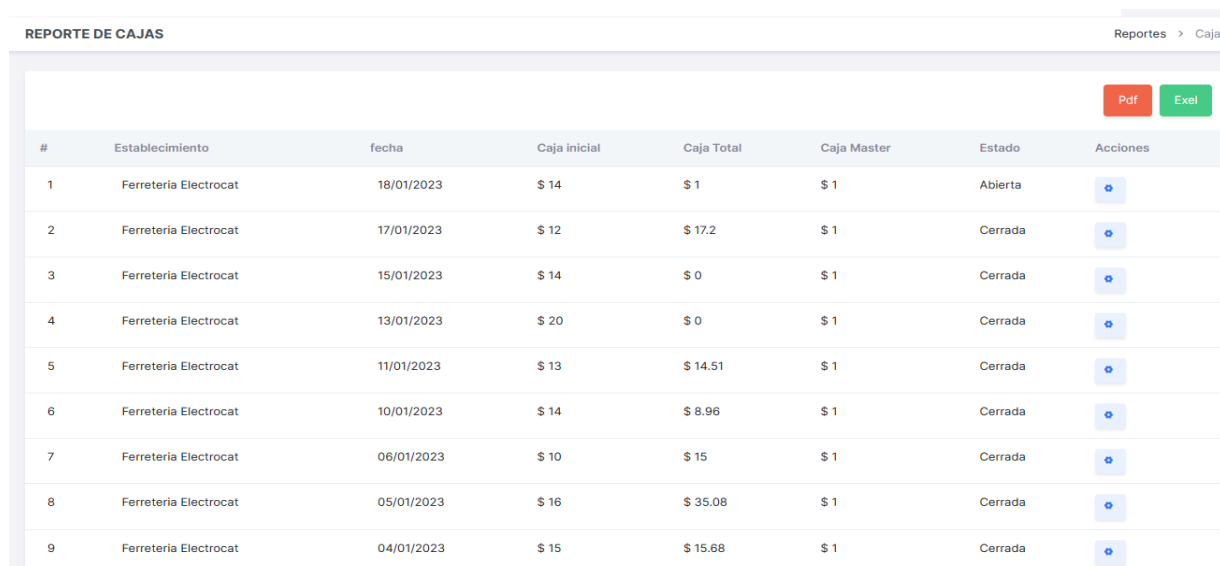
Anteriores Siguientes

*Nota.* La figura presenta la interfaz para listar las ventas con sus acciones correspondientes.

Fuente: Elaboración propia.

### Figura 37










*Interfaz gráfica para reporte de caja.*



REPORTES DE CAJAS

Reportes > Cajas

Pdf Exel

#	Establecimiento	fecha	Caja inicial	Caja Total	Caja Master	Estado	Acciones
1	Ferretería Electrocat	18/01/2023	\$ 14	\$ 1	\$ 1	Abierta	
2	Ferretería Electrocat	17/01/2023	\$ 12	\$ 17.2	\$ 1	Cerrada	
3	Ferretería Electrocat	15/01/2023	\$ 14	\$ 0	\$ 1	Cerrada	
4	Ferretería Electrocat	13/01/2023	\$ 20	\$ 0	\$ 1	Cerrada	
5	Ferretería Electrocat	11/01/2023	\$ 13	\$ 14.51	\$ 1	Cerrada	
6	Ferretería Electrocat	10/01/2023	\$ 14	\$ 8.96	\$ 1	Cerrada	
7	Ferretería Electrocat	06/01/2023	\$ 10	\$ 15	\$ 1	Cerrada	
8	Ferretería Electrocat	05/01/2023	\$ 16	\$ 35.08	\$ 1	Cerrada	
9	Ferretería Electrocat	04/01/2023	\$ 15	\$ 15.68	\$ 1	Cerrada	



*Nota.* La figura presenta la interfaz para realizar los reportes de caja ya sea en formato PDF o Excel. Fuente: Elaboración propia.

### Figura 38

*Interfaz gráfica para reporte de ingresos o egresos.*

INGRESOS Y EGRESOS				
Ingresos		Pdf Excel		
#	Establecimiento	Año	Mes	Total
1	Ferreteria Electrocat	2023	Enero	\$451.55
Egresos		Pdf Excel		
#	Establecimiento	Año	Mes	Total
1	Ferreteria Electrocat	2023	Enero	\$175

*Nota.* La figura presenta la interfaz para realizar los reportes de ingresos o egresos ya sea en formato PDF o Excel. Fuente: Elaboración propia.

### Figura 39

*Interfaz gráfica para reporte de productos.*

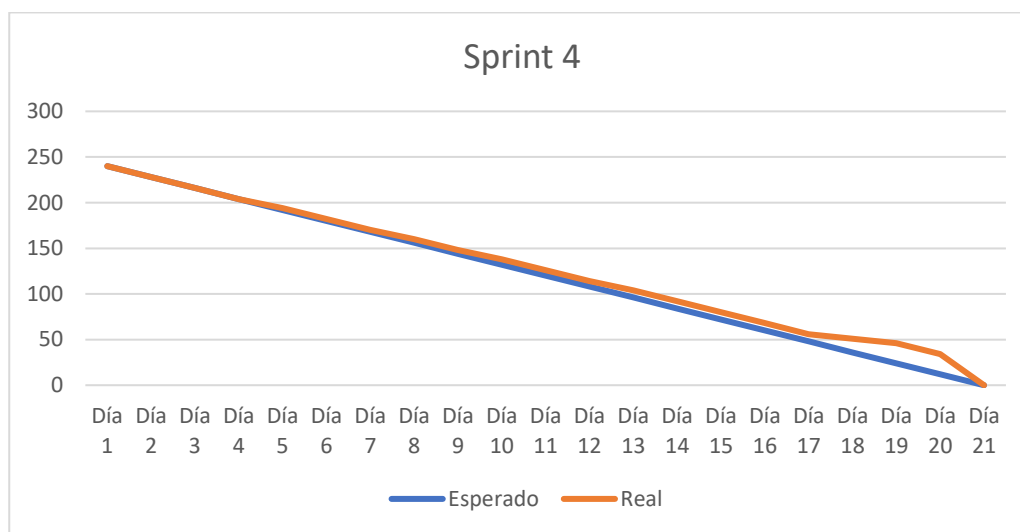
PRODUCTOS				
Q Buscar...				
		Pdf Excel		
#	Código	Nombre	Categoria	Total vendidos
1	002	Cemento	Construccion	17
2	001	pintura	Pinturas	16
3	003	Tornillos	Construccion	3
4	004	ducha	Construccion	1
Mostrando 4 resultados totales				Anteriores Siguietes

*Nota.* La figura presenta la interfaz para realizar los reportes de productos ya sea en formato PDF o Excel. Fuente: Elaboración propia.

**Burndown Chart Sprint 4.** La ejecución de este sprint presentó algunos inconvenientes en los primeros días debido a corrección de la lógica del negocio en el proceso de ventas, de esta manera cumplir con los requerimientos planteados inicialmente. A pesar de aquello, y como se puede observar en la figura 40, el sprint se cumplió dentro de la fecha establecida, pero con un flujo de trabajo pausado en relación al tiempo esperado.

**Figura 40**

*Burndown Chart Sprint 4.*



*Nota.* El gráfico presenta el trabajo realizado en comparación a la planificación esperada en función del tiempo real para completar el sprint 4. Fuente: Elaboración propia.

### **Resultado del Sprint 5**

**Módulo para la Gestión de Facturación.** Esta es la parte del sistema que gestiona la facturación y permite realizar el proceso de la facturación electrónica de las ventas realizadas, como también emisión de las notas de crédito. Se podrá revisar las facturas generadas de las como también él envió y autorización de las mismas.

En las figuras 41, 42, 43 y 44 se puede observar la interfaz implementada para el proceso de facturación electrónica. Por otra parte, en la figura # se observa las facturas enviadas y autorizadas.



## Figura 41

Interfaz gráfica de facturas generadas.

FACTURAS Comprobantes > Factura

Q Buscar...

Generadas Enviadas Autorizadas Anuladas

#	Secuencial	Cliente	fecha	Sub Total	Sub Total IVA	Descuento	Total	Acciones
1	000000027	Consumidor Final	17/01/2023	\$ 15.36	\$ 1.84	\$ 0	\$ 17.2	 

Mostrando 1 resultados totales Anteriores Siguietes

*Nota.* La figura presenta la interfaz de las facturas generadas con sus acciones correspondientes. Fuente: Elaboración propia.



## Figura 42

Interfaz gráfica de facturas enviadas.

FACTURAS Comprobantes > Factura

Q Buscar...

Generadas Enviadas Autorizadas Anuladas

#	Secuencial	Cliente	fecha	Sub Total	Sub Total IVA	Descuento	Total	Acciones
1	000000027	Consumidor Final	17/01/2023	\$ 15.36	\$ 1.84	\$ 0	\$ 17.2	 

Mostrando 1 resultados totales Anteriores Siguietes

*Nota.* La figura presenta la interfaz de las facturas enviadas con sus acciones correspondientes. Fuente: Elaboración propia.

### Figura 43

Interfaz gráfica de facturas autorizadas.

<a href="#">Generadas</a> <a href="#">Enviadas</a> <a href="#">Autorizadas</a> <a href="#">Anuladas</a>								
#	Secuencial	Cliente	fecha	Sub Total	Sub Total IVA	Descuento	Total	Acciones
1	000000027	Consumidor Final	17/01/2023	\$ 15.36	\$ 1.84	\$ 0	\$ 17.2	<a href="#">Ver</a> <a href="#">Enviar</a>
2	000000026	Consumidor Final	11/01/2023	\$ 14.35	\$ 0.16	\$ 0	\$ 14.51	<a href="#">Ver</a> <a href="#">Enviar</a>
3	000000025	Consumidor Final	10/01/2023	\$ 8	\$ 0.96	\$ 0	\$ 8.96	<a href="#">Ver</a> <a href="#">Enviar</a>
4	000000021	Eddy Fernando Castillo Cujilan	03/01/2023	\$ 15	\$ 0	\$ 0	\$ 15	<a href="#">Ver</a> <a href="#">Enviar</a>
5	000000019	Consumidor Final	03/01/2023	\$ 13	\$ 0	\$ 0	\$ 13	<a href="#">Ver</a> <a href="#">Enviar</a>
6	000000018	Christian Enrique Pérez Garofalo	03/01/2023	\$ 15	\$ 0	\$ 0	\$ 15	<a href="#">Ver</a> <a href="#">Enviar</a>
7	000000017	Christian Enrique Pérez Garofalo	03/01/2023	\$ 8	\$ 0.96	\$ 0	\$ 8.96	<a href="#">Ver</a> <a href="#">Enviar</a>
8	000000016	Eddy Fernando Castillo Cujilan	03/01/2023	\$ 8	\$ 0.96	\$ 0	\$ 8.96	<a href="#">Ver</a> <a href="#">Enviar</a>
9	000000015	Consumidor Final	03/01/2023	\$ 45	\$ 0	\$ 0	\$ 45	<a href="#">Ver</a> <a href="#">Enviar</a>

*Nota.* La figura presenta la interfaz de las facturas autorizadas con sus acciones correspondientes. Fuente: Elaboración propia.

### Figura 44

Interfaz gráfica de facturas anuladas.

FACTURAS									Comprobantes > Factura
<input type="text" value="Q Buscar..."/>									
<a href="#">Generadas</a> <a href="#">Enviadas</a> <a href="#">Autorizadas</a> <a href="#">Anuladas</a>									
#	Secuencial	Cliente	fecha	Sub Total	Sub Total IVA	Descuento	Total	Acciones	
1	000000023	Consumidor Final	05/01/2023	\$ 33.2	\$ 2.58	\$ 3.3	\$ 35.78	<a href="#">Ver</a>	
2	000000020	Christian Enrique Pérez Garofalo	03/01/2023	\$ 8	\$ 0.96	\$ 0	\$ 8.96	<a href="#">Ver</a>	
3	000000015	Christian Enrique Pérez Garofalo	03/01/2023	\$ 4.37	\$ 0.96	\$ 3.63	\$ 5.33	<a href="#">Ver</a>	
4	000000014	Christian Enrique Pérez Garofalo	03/01/2023	\$ 48.5	\$ 1.92	\$ 12.5	\$ 50.42	<a href="#">Ver</a>	
5	000000013	Consumidor Final	03/01/2023	\$ 42	\$ 0	\$ 3	\$ 42	<a href="#">Ver</a>	

Mostrando 5 resultados totales

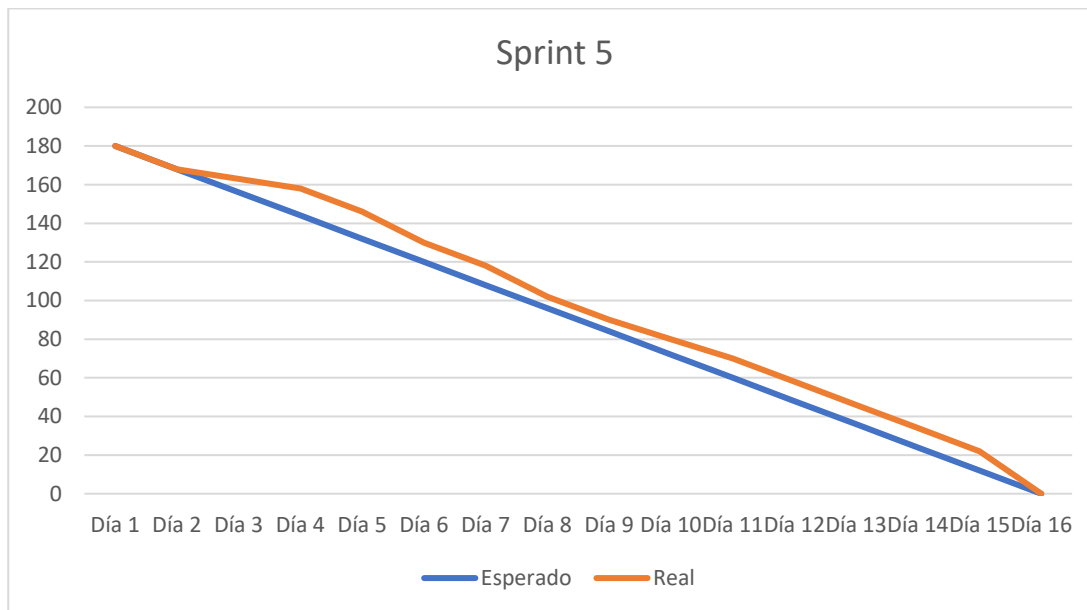
[Anteriores](#) [Siguietes](#)

*Nota.* La figura presenta la interfaz de las facturas anuladas con sus acciones correspondientes. Fuente: Elaboración propia.

**Burndown Chart Sprint 5.** La ejecución de este sprint presentó inconvenientes en los primeros días debido a errores en la generación de las facturas, pero con ayuda de la guía que ofrece el SRI se solventó los inconvenientes, de esta manera cumplir con los requisitos de los procesos en la emisión de comprobantes, como también, cumplir con los requerimientos planteados inicialmente. A pesar de aquello, y como se observa en la figura 45, el sprint se cumplió dentro de la fecha establecida, pero con un flujo de trabajo pausado en relación al tiempo esperado.

### Figura 45

*Burndown Chart Sprint 5.*



*Nota.* El gráfico presenta el trabajo realizado en comparación a la planificación esperada en función del tiempo real para completar el sprint 5. Fuente: Elaboración propia.

Para una información más precisa sobre el proceso de uso del sistema y las especificaciones de funcionamiento de cada módulo implementado, refiérase al Anexo 2, denominado manual de usuario.

## Pruebas de Calidad de Código

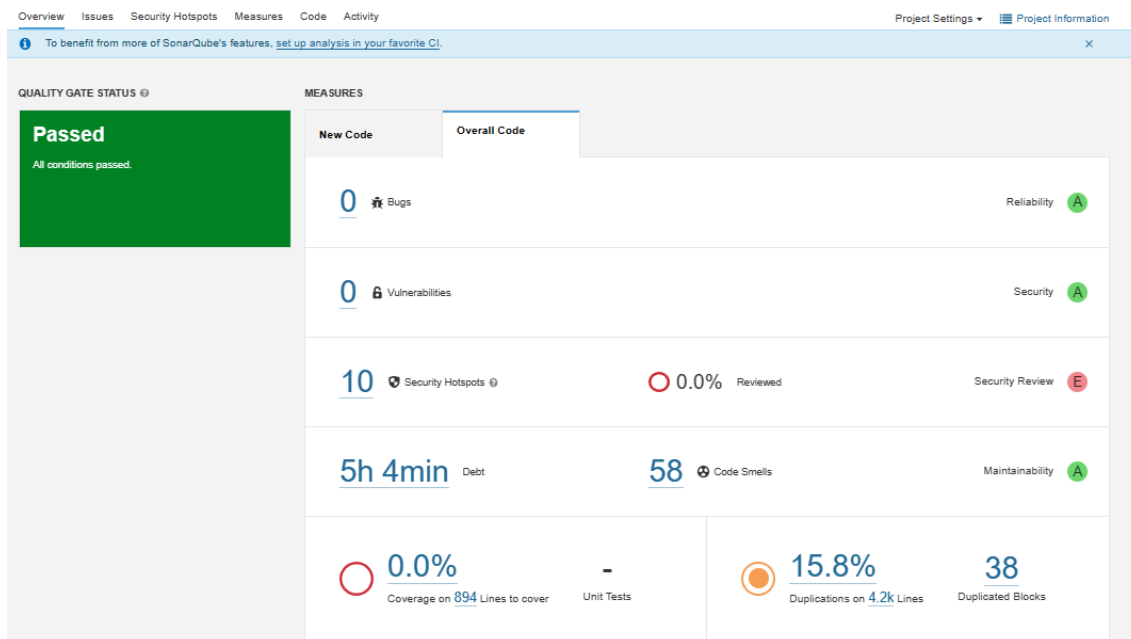
Con la finalidad de tener un código legible seguro y mantenible, es necesario realizar un análisis del mismo para lo cual existen diversas herramientas que permiten realizar un análisis estático automático del código fuente, buscando patrones con errores, malas prácticas o incidencias. La herramienta seleccionada para realizar el análisis es SonarQube ya que admite diversos lenguajes de programaciones como lo es JavaScript y TypeScript, lenguajes utilizados para el desarrollo de los microservicios.

### **Análisis de Código del Microservicio de Autenticación**

Como se observa en la Figura 46 El análisis de todo el código dio como resultado aprobado, indicando un total de cero bugs, cero vulnerabilidades, cincuenta y ocho code smells y diez puntos críticos de seguridad que hace referencia a código sensible desde el punto de vista de la seguridad que requiere una revisión manual para evaluar si existe o no una vulnerabilidad. Una vez analizado el código se determinó que existen varios riesgos de seguridad y malas prácticas que deben de corregirse.

### Figura 46

*Calidad de código microservicio de autenticación.*

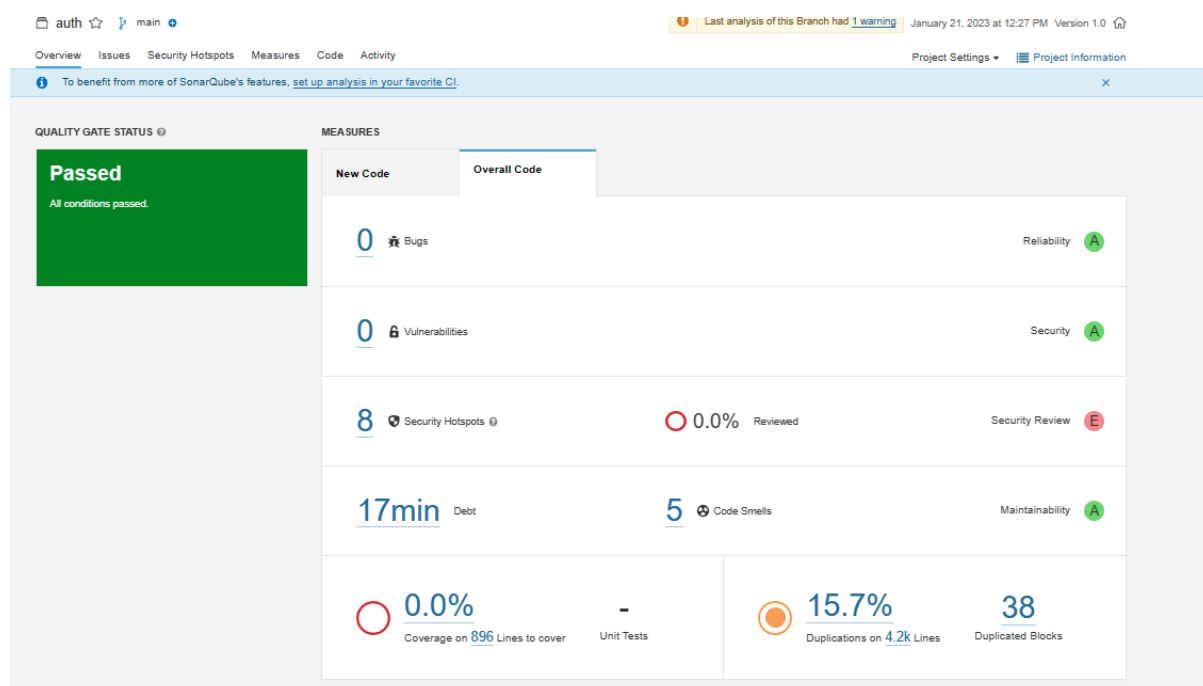


*Nota.* El gráfico presenta las condiciones de calidad de código del microservicio de autenticación. Fuente: Elaboración propia.

Como se observa en la Figura 47 Una vez corregido el código se realizó nuevamente el análisis, indicando un total de cero bugs, cero vulnerabilidades, ocho puntos críticos de seguridad y solo cinco code smells. Una vez analizado el código se determinó que no existen vulnerabilidades, y obteniendo la calificación A en la mantenibilidad del código.

## Figura 47

### *Correcciones de calidad de código de microservicio Autenticación*



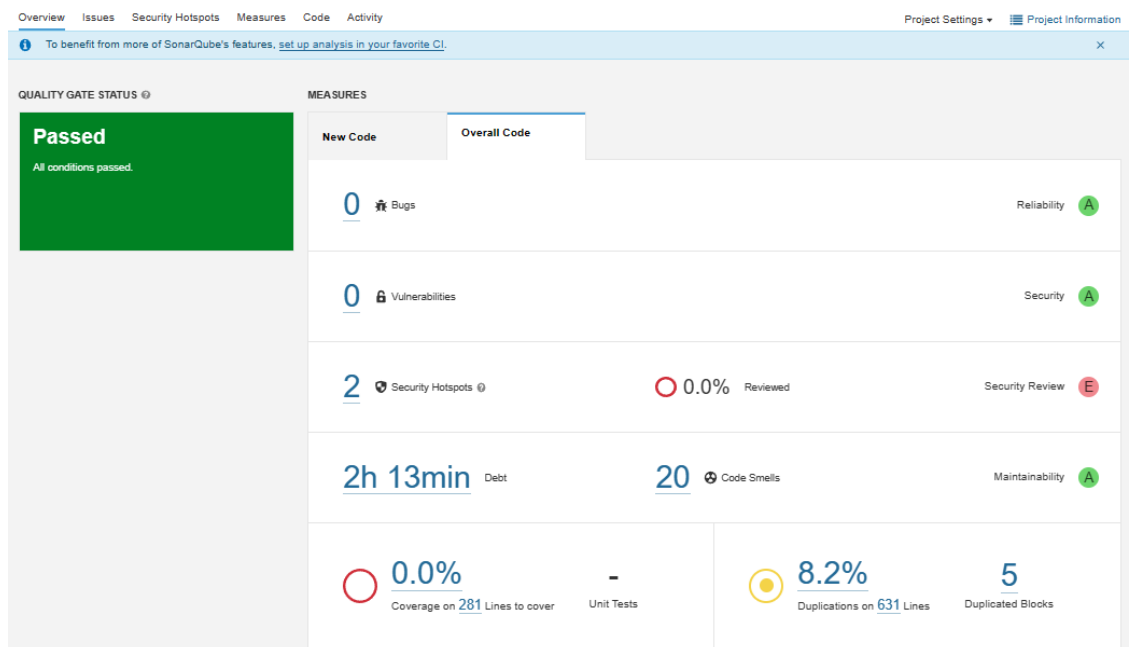
*Nota.* El gráfico presenta las condiciones de calidad de código aprobadas del microservicio de autenticación. Fuente: Elaboración propia.

### **Análisis de Código del Microservicio de Clientes**

Como se observa en la Figura 48, el análisis de todo el código dio como resultado aprobado, indicando un total de cero bugs, cero vulnerabilidades, dos puntos críticos de seguridad y veinte code smells, determinando que no existen vulnerabilidades, pero hay varias partes del código que deben mejorarse y corregirse.

## Figura 48

Calidad de código microservicio de clientes.



*Nota.* El gráfico presenta las condiciones de calidad de código del microservicio de clientes.

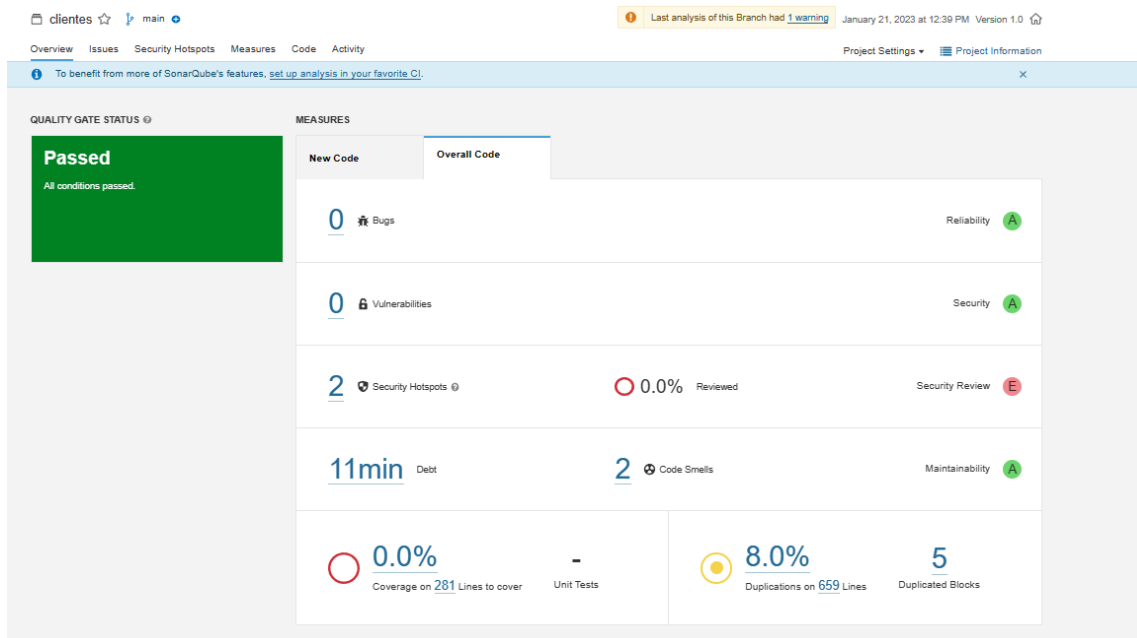
Fuente: Elaboración propia.

Como se observa en la Figura 49. Una vez corregido todo el código dio como resultado aprobado, indicando un total de cero bugs, cero vulnerabilidades, dos puntos críticos de seguridad determinando que no existen vulnerabilidades, y solo dos code smells obteniendo la calificación A en la mantenibilidad del código.



Figura 49

## Correcciones de calidad de código de microservicio Clientes



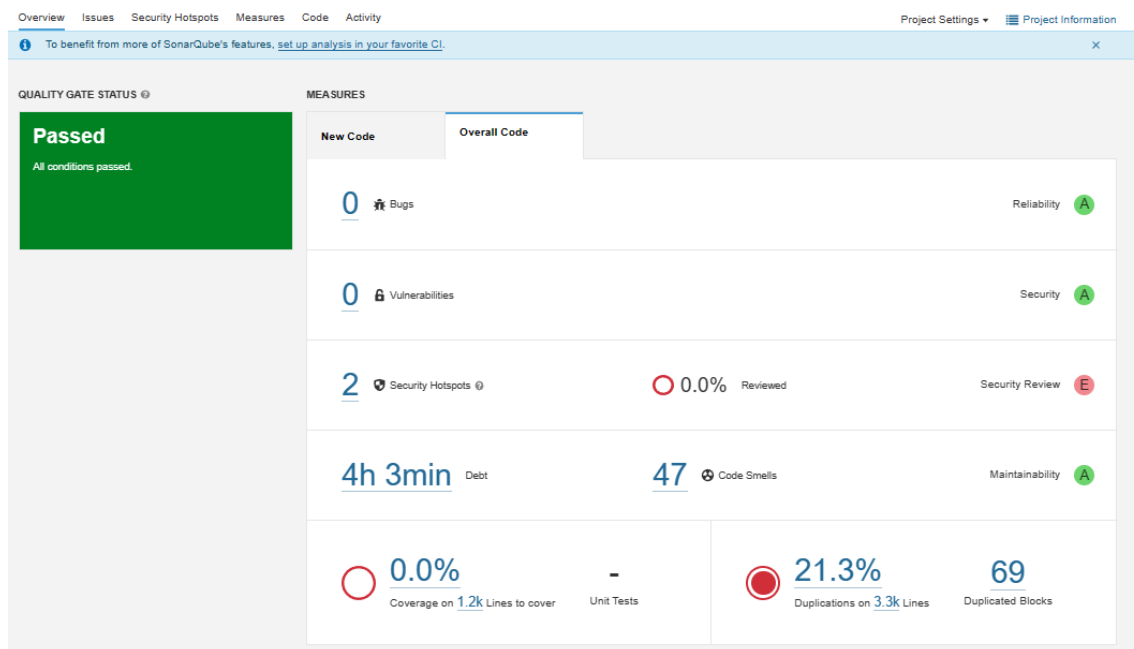
*Nota.* El gráfico presenta las condiciones de calidad de código corregidas del microservicio de clientes. Fuente: Elaboración propia.

**Análisis de Código del Microservicio de Productos**

Como se observa en la Figura 50 El análisis de todo el código dio como resultado aprobado, indicando un total de cero bugs, cero vulnerabilidades, dos puntos críticos de seguridad determinando que no existen vulnerabilidades, y un total de cuarenta y siete code smells, secciones de código que deben ser modificarse.

## Figura 50

Calidad de código microservicio de productos.



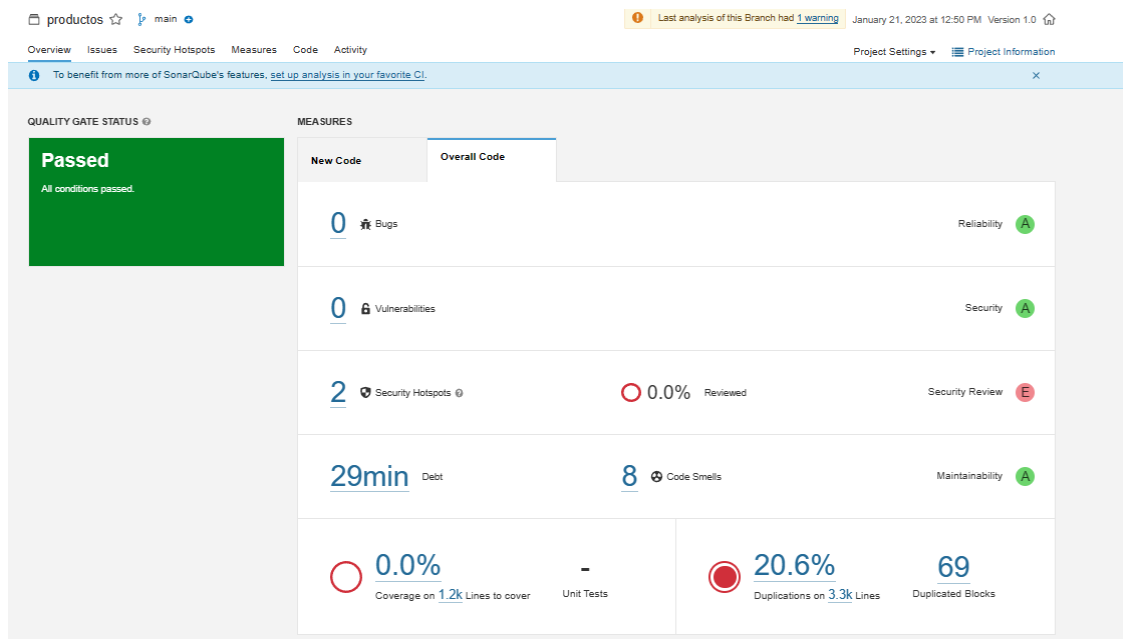
*Nota.* El gráfico presenta las condiciones de calidad de código del microservicio de productos.

Fuente: Elaboración propia.

Como se observa en la Figura 51 El análisis de todo el código luego de las correcciones dio como resultado aprobado, indicando un total de cero bugs, cero vulnerabilidades, dos puntos críticos de seguridad, y obteniendo la calificación A en la mantenibilidad del código con solo ocho code smells.

## Figura 51

### Correcciones de calidad de código de microservicio Productos



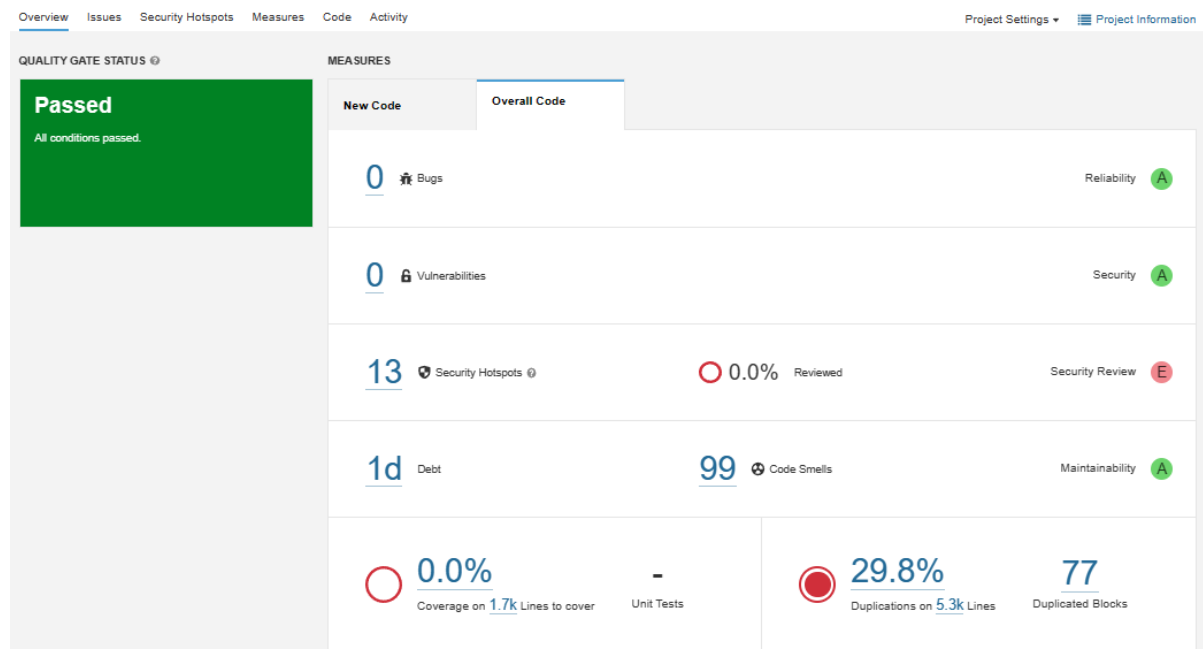
*Nota.* El gráfico presenta las condiciones de calidad de código aprobadas del microservicio de productos. Fuente: Elaboración propia.

### **Análisis de Código del Microservicio de Ventas**

Como se observa en la Figura 52 El análisis de todo el código dio como resultado aprobado, indicando un total de cero bugs, cero vulnerabilidades, tres puntos críticos de seguridad determinando que no existen vulnerabilidades, y un total de noventa y nueve code smells, secciones de código que se deben mejorar y corregir.

## Figura 52

### Calidad de código microservicio de Ventas.



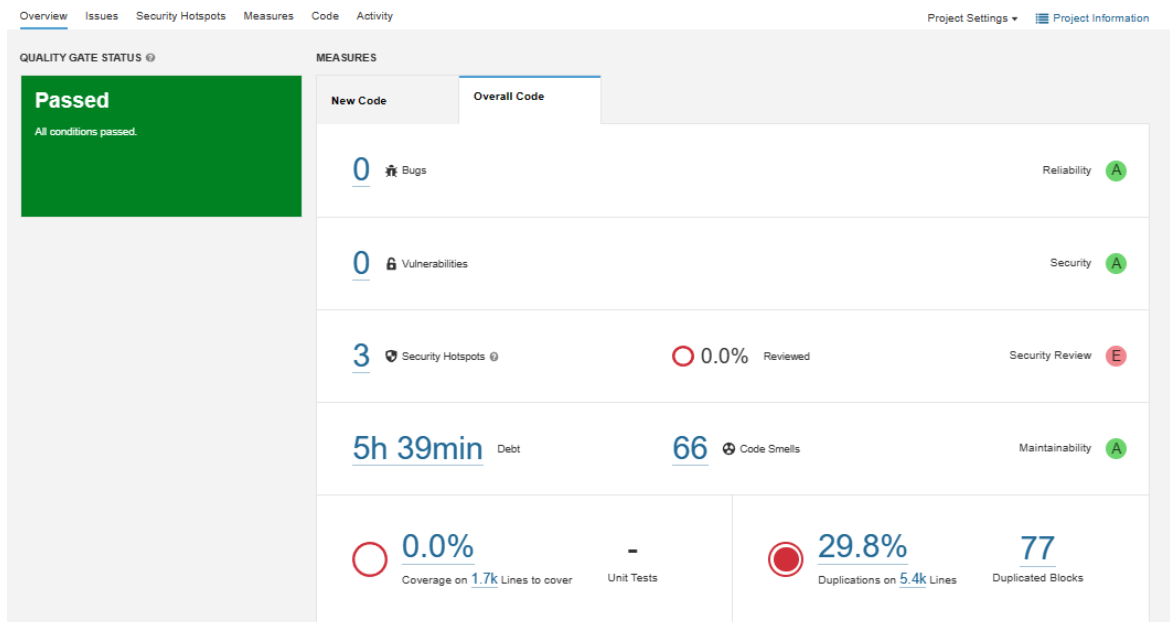
*Nota.* El gráfico presenta las condiciones de calidad de código del microservicio de ventas.

Fuente: Elaboración propia.

Como se observa en la Figura 53 El análisis de todo el código luego de las correcciones dio como resultado aprobado, indicando un total de cero bugs, cero vulnerabilidades, tres puntos críticos de seguridad, y obteniendo la calificación A en la mantenibilidad del código con 66 code smells.

## Figura 53

### Correcciones de calidad de código de microservicio Ventas



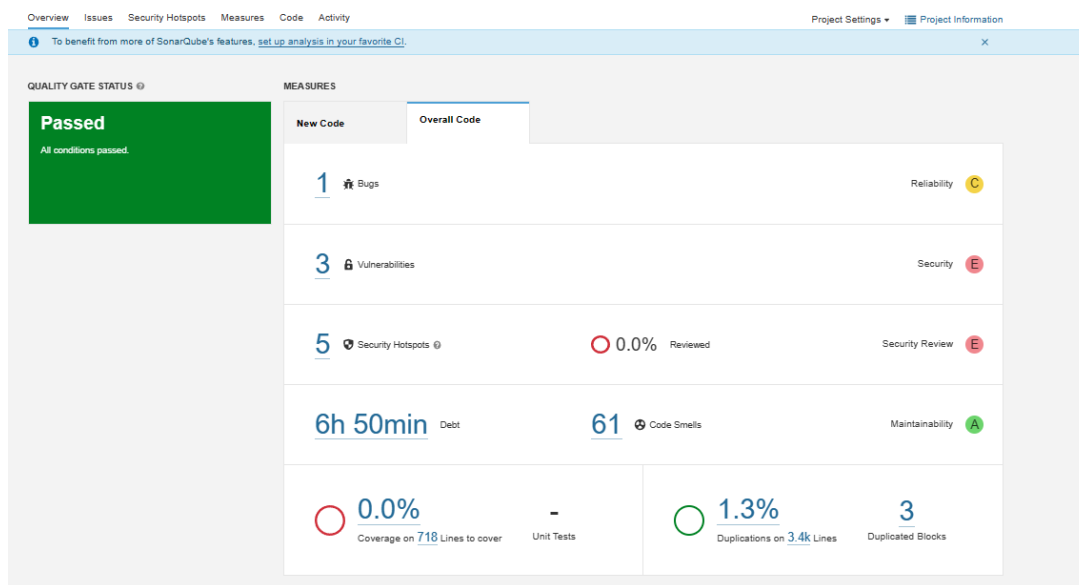
*Nota.* El gráfico presenta las condiciones de calidad de código corregidas del microservicio de ventas. Fuente: Elaboración propia.

### **Análisis de Código del Microservicio de Facturación**

Como se observa en la Figura 54 El análisis de todo el código dio como resultado aprobado, pero esta vez indicando un mayor número de vulnerabilidades en el código que deben corregirse, indicando un bug, tres vulnerabilidades, 5 puntos críticos de seguridad, y un total de sesenta y uno code smells, teniendo como prioridad la resolver los problemas de bugs y vulnerabilidades.

## Figura 54

Calidad de código microservicio de Facturación.



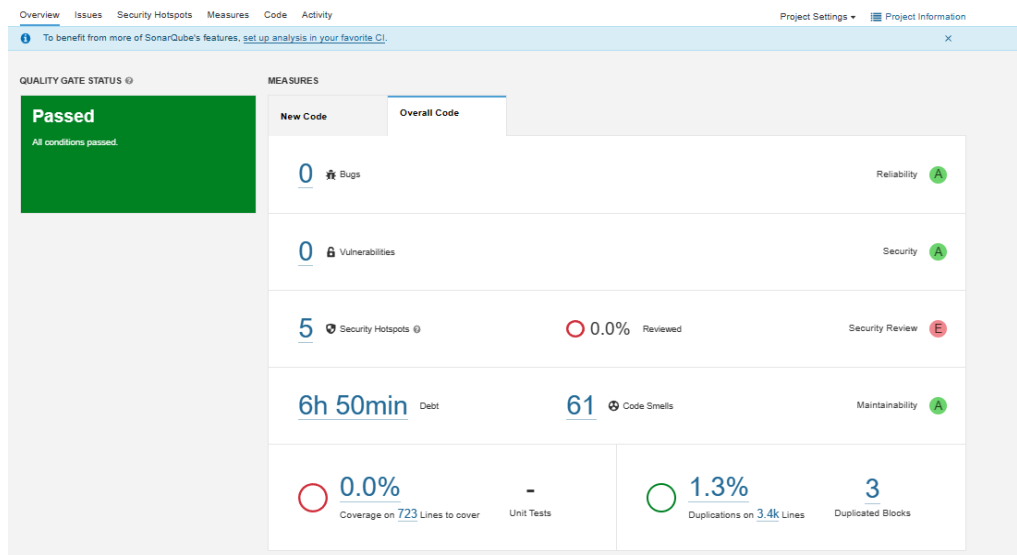
*Nota.* El gráfico presenta las condiciones de calidad de código del microservicio de Facturación.

Fuente: Elaboración propia.

Como se observa en la Figura 55 El análisis de todo el código luego de las correcciones dio como resultado aprobado, indicando esta vez ya un total de cero bugs, cero vulnerabilidades, cinco puntos críticos de seguridad, que una vez analizados se determinaron que no son riesgosos y obteniendo la calificación A en la mantenibilidad del código con sesenta y uno code smells.

## Figura 55

### Correcciones de calidad de código de microservicio Ventas



*Nota.* El gráfico presenta las condiciones de calidad de código corregidas del microservicio de facturación. Fuente: Elaboración propia.

### Pruebas de carga

Según como lo menciona (Wolf, 2019), las pruebas de carga o estrés a una API son una técnica utilizada para evaluar el rendimiento de una aplicación o sistema. Estas pruebas simulan varias solicitudes a la API para medir cómo se comporta bajo una carga elevada de tráfico y determinar si puede manejar una cantidad adecuada de solicitudes simultáneas (Barber & Sime, 2017). Estas pruebas también se utilizan para identificar cuellos de botella y problemas de escalabilidad en una aplicación (Richardson, 2020).

Algunas herramientas populares para realizar pruebas de carga en APIs incluyen Apache JMeter, LoadRunner, Gatling, y Siege. La herramienta seleccionada es Apache JMeter, en el cual se realizan varias pruebas con diversas configuraciones a las APIs desplegadas y orquestadas por Kubernetes.

### **Escenario**

Se realizarán diversas peticiones GET a diferentes servicios que se conectan la base de datos PostgreSQL y envían los datos al cliente, en cada petición se envía un total de 500 registros que están almacenados en cada tabla.

Cada microservicio desplegado cuenta con solo una réplica orquestada por Kubernetes y con una configuración en la cantidad de recursos que puede utilizar y el límite al que puede llegar, como se muestra en la Figura 56.

### **Figura 56**

*Recursos asignados a cada contenedor.*

```
resources:  
  limits:  
    cpu: 200m  
    memory: 200Mi  
  requests:  
    cpu: 100m  
    memory: 100Mi
```

*Nota.* El gráfico presenta los recursos asignados a cada contenedor o POD con la utilización de Kubernetes. Fuente: Elaboración propia.

**Listar Empleados.** Se realiza la configuración inicial en Apache JMeter indicando a qué API se realizará la solicitud y otras configuraciones adicionales, como poder establecer las cabeceras de la solicitud, entre otros. En la figura 57 se puede observar la configuración correspondiente.



## Figura 57

Configuración API listar empleados.

Basic Advanced

Web Server

Protocol [http]:  Server Name or IP:  Port Number:

HTTP Request

GET  Content encoding:

Redirect Automatically  Follow Redirects  Use KeepAlive  Use multipart/form-data  Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?

*Nota.* El gráfico presenta la configuración de la API de listar empleados para listar los clientes.

Fuente: Elaboración propia

Al empezar la prueba simulando la petición de solo un usuario como se muestra en la Figura 58 se obtiene como resultado que el tiempo de respuesta promedio es de 757 ms y un 0% de errores. Con 10 usuarios como se observa en la Figura 59 se obtiene como resultado que el tiempo de respuesta promedio es de 721 ms y un 0% de errores. Por último, con 100 usuarios como indica la Figura 60 los resultados obtenidos dan un tiempo promedio para la respuesta de 4881 ms y un 0% de errores.

Para un total de 100 usuarios realizando una petición se puede observar que el servicio tiene un rendimiento de 14.7 solicitudes por segundo, donde el tiempo de respuesta varía del promedio 659.95 ms aproximadamente, valores que cumplen satisfactoriamente los tiempos de respuesta esperados para el sistema web.

## Figura 58

*Petición con un usuario.*

Summary Report

Name:

Comments:

Write results to file / Read from file

Filename:   Log/Display Only:  Errors  Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
usuarios	1	757	757	757	0.00	0.00%	1.3/sec	143.59	0.42	111305.0
TOTAL	1	757	757	757	0.00	0.00%	1.3/sec	143.59	0.42	111305.0

*Nota.* El gráfico presenta la simulación de petición con un usuario. Fuente: Elaboración propia.

## Figura 59

*Petición con 10 usuarios.*

Summary Report

Name:

Comments:

Write results to file / Read from file

Filename:   Log/Display Only:  Errors  Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
usuarios	10	721	683	862	53.66	0.00%	6.3/sec	681.48	2.01	111305.0
TOTAL	10	721	683	862	53.66	0.00%	6.3/sec	681.48	2.01	111305.0

*Nota.* El gráfico presenta la simulación de petición con 10 usuarios. Fuente: Elaboración propia.

## Figura 60

*Petición con 100 usuarios.*

Summary Report

Name:

Comments:

Write results to file / Read from file

Filename:   Log/Display Only:  Errors  Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
usuarios	100	4881	2456	6184	659.95	0.00%	14.7/sec	1593.55	4.70	111305.0
TOTAL	100	4881	2456	6184	659.95	0.00%	14.7/sec	1593.55	4.70	111305.0

*Nota.* El gráfico presenta la simulación de petición con 100 usuarios. Fuente: Elaboración propia.

**Listar ventas.** De igual manera se realiza la configuración inicial en Apache JMeter indicando a qué API se realizará la solicitud y todos los datos que sean necesarios como se muestra en la Figura 61.

## Figura 61

Configuración API listar ventas.

*Nota.* El gráfico presenta la configuración de la API para listar las ventas. Fuente: Elaboración propia

La primera prueba empieza simulando la petición de solo un usuario como se muestra en la Figura 62 se obtiene como resultado que el tiempo de respuesta promedio es de 1110 ms y un 0% de errores. Con 10 usuarios como se observa en la Figura 63 se obtiene como resultado que el tiempo de respuesta promedio es de 2776 ms y un 0% de errores. Con 25 usuarios como indica la Figura 64 los resultados obtenidos dan un tiempo promedio para la respuesta de 3543 ms y un 0% de errores, siendo este el límite que puede abarcar solo un pod con los recursos indicados anteriormente. Obteniendo solo un rendimiento de 4.5 solicitudes por segundo.

## Figura 62

Petición con un usuario.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
ventas	1	1110	1110	1110	0.00	0.00%	54.1/min	133.39	0.31	151617.0
TOTAL	1	1110	1110	1110	0.00	0.00%	54.1/min	133.39	0.31	151617.0

*Nota.* El gráfico presenta la simulación de petición con un usuario. Fuente: Elaboración propia.

### Figura 63

*Petición con 10 usuarios.*

Summary Report

Name:

Comments:

Write results to file / Read from file

Filename:   Log/Display Only:  Errors  Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
ventas	1	1110	1110	1110	0.00	0.00%	54.1/min	133.39	0.31	151617.0
TOTAL	1	1110	1110	1110	0.00	0.00%	54.1/min	133.39	0.31	151617.0

*Nota.* El gráfico presenta la simulación de petición con 10 usuarios. Fuente: Elaboración propia.

### Figura 64

*Petición con 25 usuarios.*

Summary Report

Name:

Comments:

Write results to file / Read from file

Filename:   Log/Display Only:  Errors  Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
ventas	25	3543	1026	4911	1092.71	0.00%	4.5/sec	668.64	1.54	151617.0
TOTAL	25	3543	1026	4911	1092.71	0.00%	4.5/sec	668.64	1.54	151617.0

*Nota.* El gráfico presenta la simulación de petición con 25 usuarios el límite en este apartado.

Fuente: Elaboración propia.

**Listar Ventas - Sistema Escalado.** Existe una gran diferencia entre la prueba de listar usuarios que llegó a las 100 peticiones simultáneas a comparación que la prueba de listar ventas que solo llega a 25 peticiones. Esto se debe a la complejidad de procesos que realiza cada api, en el caso de los usuarios la api se conecta a la base de datos y envía la información al cliente, en cambio la api de ventas no solo recupera la información, también realiza un proceso de paginado y filtrado si es necesario, por lo cual realiza más de una consulta a la base de datos, lo que implica el consumo de más recursos.

Gracias a la flexibilidad de la arquitectura de microservicios y las funcionalidades que brinda Kubernetes se puede solucionar el inconveniente de forma sencilla, en esta ocasión para la prueba se incrementó el número de réplicas a 2 para el servicio de ventas conservando los mismos recursos indicados anteriormente.

Esta vez con 2 réplicas del servicio se realiza la prueba empieza simulando la petición de 75 usuarios como se muestra en la Figura 65 obteniendo como resultado que el tiempo de respuesta promedio es de 4549 ms y un 0% de errores. Con 100 usuarios como se observa en la Figura 66 se obtiene como resultado que el tiempo de respuesta promedio es de 4400 ms y un 12% de errores, observando una gran mejora a comparación de utilizar una sola réplica del servicio, esta vez observando que el rendimiento se incrementó a 13.2 solicitudes por segundo.

### Figura 65

*Petición de prueba con 75 usuarios.*

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
ventas	75	4549	1765	6918	1531.29	0.00%	10.0/sec	1485.59	3.43	151617.0
TOTAL	75	4549	1765	6918	1531.29	0.00%	10.0/sec	1485.59	3.43	151617.0

*Nota.* El gráfico presenta la simulación de petición con 75 usuarios haciendo el uso de dos réplicas del servicio. Fuente: Elaboración propia.

### Figura 66

*Petición de prueba con 100 usuarios.*

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
ventas	100	4400	963	7304	1799.44	12.00%	13.2/sec	1715.99	4.50	133457.2
TOTAL	100	4400	963	7304	1799.44	12.00%	13.2/sec	1715.99	4.50	133457.2

*Nota.* El gráfico presenta la simulación de petición con 100 usuarios haciendo el uso de dos réplicas del servicio. Fuente: Elaboración propia.

**Listar Clientes.** Se realiza la configuración inicial en Apache JMeter indicando a qué API se realizará la solicitud como se indica en la Figura 67.

## Figura 67

*Configuración de API listar clientes.*

*Nota.* El gráfico presenta la configuración de la API de listar. Fuente: Elaboración propia

La primera prueba empieza simulando la petición de solo un usuario como se muestra en la Figura 68 se obtiene como resultado que el tiempo de respuesta promedio es de 1456 ms y un 0% de errores. Con 10 usuarios como se observa en la Figura 69 se obtiene como resultado que el tiempo de respuesta promedio es de 1622 ms y un 0% de errores. Con 40 usuarios como indica la Figura 70 los resultados obtenidos dan un tiempo promedio para la respuesta de 5485 ms y un 0% de errores, obteniendo un rendimiento de 5.6 solicitudes por segundo, siendo este el límite que puede abarcar solo un pod con los recursos indicados anteriormente. De igual manera el funcionamiento de la API es similar a la de ventas por lo que la solución para mejorar los resultados de las pruebas es agregando una réplica más como se comprobó anteriormente.

## Figura 68

*Petición de prueba con un usuario.*

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
clientes	1	1456	1456	1456	0.00	0.00%	41.2/min	65.48	0.22	97630.0
TOTAL	1	1456	1456	1456	0.00	0.00%	41.2/min	65.48	0.22	97630.0

*Nota.* El gráfico presenta la simulación de petición con un usuario. Fuente: Elaboración propia.

## Figura 69

*Petición de prueba con 10 usuarios.*

Summary Report

Name:

Comments:

Write results to file / Read from file

Filename   Log/Display Only:  Errors  Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB...	Sent KB/sec	Avg. Bytes
clientes	10	1622	1258	2036	282.41	0.00%	3.9/sec	375.21	1.29	97630.0
TOTAL	10	1622	1258	2036	282.41	0.00%	3.9/sec	375.21	1.29	97630.0

*Nota.* El gráfico presenta la simulación de petición con 10 usuarios. Fuente: Elaboración propia.

## Figura 70

*Petición de prueba con 40 usuarios.*

Summary Report

Name:

Comments:

Write results to file / Read from file

Filename   Log/Display Only:  Errors  Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB...	Sent KB/sec	Avg. Bytes
clientes	10	1622	1258	2036	282.41	0.00%	3.9/sec	375.21	1.29	97630.0
TOTAL	10	1622	1258	2036	282.41	0.00%	3.9/sec	375.21	1.29	97630.0

*Nota.* El gráfico presenta la simulación de petición con 40 usuarios el límite en este apartado.

Fuente: Elaboración propia.

**Listar productos.** Se realiza la configuración inicial en Apache JMeter indicando a qué API se realizará la solicitud como se indica en la Figura 71.

## Figura 71

*Configuración de API listar productos.*

*Nota.* El gráfico presenta la configuración de la API de listar productos. Fuente: Elaboración propia

La primera prueba empieza simulando la petición de solo un usuario como se muestra en la Figura 72 se obtiene como resultado que el tiempo de respuesta promedio es de 1123 ms y un 0% de errores. Con 10 usuarios como se observa en la Figura 73 se obtiene como resultado que el tiempo de respuesta promedio es de 1001 ms y un 0% de errores. Con 60 usuarios como indica la Figura 74 los resultados obtenidos dan un tiempo promedio para la respuesta de 3611 ms y un 0% de errores, obteniendo un rendimiento de 11.8 solicitudes por segundo, siendo este el límite al que puede llegar.

## Figura 72

*Petición de prueba con un usuario.*

Summary Report

Name: productos

Comments:

Write results to file / Read from file

Filename:   Log/Display Only:  Errors  Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
productos	1	1123	1123	1123	0.00	0.00%	53.4/min	141.98	0.30	163268.0
TOTAL	1	1123	1123	1123	0.00	0.00%	53.4/min	141.98	0.30	163268.0

*Nota.* El gráfico presenta la simulación de petición con un usuario. Fuente: Elaboración propia.



### Figura 73

*Petición de prueba con 10 usuarios.*

Summary Report

Name:

Comments:

Write results to file / Read from file

Filename:   Log/Display Only:  Errors  Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
productos	10	1001	926	1170	76.78	0.00%	5.5/sec	873.17	1.82	163268.0
TOTAL	10	1001	926	1170	76.78	0.00%	5.5/sec	873.17	1.82	163268.0

*Nota.* El gráfico presenta la simulación de petición con 10 usuarios. Fuente: Elaboración propia.

### Figura 74

*Petición de prueba con 60 usuarios.*

Summary Report

Name:

Comments:

Write results to file / Read from file

Filename:   Log/Display Only:  Errors  Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
productos	10	1001	926	1170	76.78	0.00%	5.5/sec	873.17	1.82	163268.0
TOTAL	10	1001	926	1170	76.78	0.00%	5.5/sec	873.17	1.82	163268.0

*Nota.* El gráfico presenta la simulación de petición con 60 usuarios el límite en este apartado.

Fuente: Elaboración propia.

### Pruebas de Funcionalidad

Se realizaron pruebas de funcionamiento del sistema con el gerente de la ferretería Electrocat, las cuales se realizaron las pruebas según las funcionalidades de cada microservicio implementado. Los detalles de las pruebas del primer microservicio se observan en las figuras 75, 76, 77 y 78.

**Figura 75***Prueba de Funcionamiento Login del Sistema*

<b>ID Prueba:</b>	PF-01	<b>N° de Sprint:</b>	1	<b>Fecha:</b>	23-01-2023
<b>Nombre del tester del software:</b>	Antony Mantuano				
<b>Módulo:</b>	Autenticación				
<b>Objetivo:</b>	Iniciar sesión dentro del sistema				
<b>Acción:</b>	Inicio de sesión				
<b>Resultado esperado:</b>	El usuario ingresa al sistema de forma correcta utilizando las credenciales generadas.				
<b>Resultado obtenido:</b>	El usuario logra ingresar al sistema de forma correcta.				
<b>Observaciones:</b>	Ninguna				
<b>Se encontro algun error:</b>	SI		NO	X	

*Nota.* La figura presenta los detalles de la prueba de login y autenticación del sistema. Fuente:

Elaboración propia.

**Figura 76***Prueba de Recuperación de Contraseña*

<b>ID Prueba:</b>	PF-02	<b>N° de Sprint:</b>	1	<b>Fecha:</b>	23-01-2023
<b>Nombre del tester del software:</b>	Antony Mantuano				
<b>Módulo:</b>	Autenticación				
<b>Objetivo:</b>	El usuario puede recuperar su contraseña por medio de su email.				
<b>Acción:</b>	Solicitar un código de recuperación al email y actualizar la contraseña.				
<b>Resultado esperado:</b>	El usuario recibe en su email un código de recuperación con el cual puede cambiar de contraseña.				
<b>Resultado obtenido:</b>	El usuario ingresó su email y recibio un código de recuperación con el actual cambio de forma correcta la contraseña.				
<b>Observaciones:</b>	Ninguna				
<b>Se encontro algun error:</b>	SI		NO	X	

*Nota.* La figura presenta los detalles de la prueba de recuperación de contraseña. Fuente:

Elaboración propia.

### Figura 77

#### *Prueba de Funcionamiento Registro Empleados*

<b>ID Prueba:</b>	PF-03	<b>N° de Sprint:</b>	1	<b>Fecha:</b>	23-01-2023
<b>Nombre del tester del software:</b>	Antony Mantuano				
<b>Módulo:</b>	Autenticación.				
<b>Objetivo:</b>	Registro de un nuevo empleado.				
<b>Acción:</b>	Registrar la cuenta de un nuevo usuario del perfil empleado.				
<b>Resultado esperado:</b>	El usuario registra un usuario empleado de forma correcta.				
<b>Resultado obtenido:</b>	El usuario de forma exitosa registra los datos de una nueva cuenta de perfil empleado.				
<b>Observaciones:</b>	Ninguna				
<b>Se encontro algun error:</b>	SI		NO	X	

*Nota.* La figura presenta los detalles de la prueba de funcionamiento del registro de empleados.

Fuente: Elaboración propia.

**Figura 78***Prueba de Funcionamiento Control de Deudas*

<b>ID Prueba:</b>	PF-04	<b>N° de Sprint:</b>	1	<b>Fecha:</b>	23-01-2023
<b>Nombre del tester del software:</b>	Antony Mantuano				
<b>Módulo:</b>	Autenticación.				
<b>Objetivo:</b>	Registro de una nueva deuda				
<b>Acción:</b>	El usuario registra una nueva deuda				
<b>Resultado esperado:</b>	El usuario registra una nueva deuda y recibe una notificación de pago al email dependiendo la fecha registrada.				
<b>Resultado obtenido:</b>	El usuario registra correctamente una deuda y recibe una notificación en su email.				
<b>Observaciones:</b>	Ninguna				
<b>Se encontro algun error:</b>	SI		NO	X	

*Nota.* La figura presenta los detalles de la prueba de funcionamiento de control de deudas.

Fuente: Elaboración propia.

Se realizaron las pruebas de funcionamiento del segundo microservicio que corresponde al cliente. En la figura 79 se presentan los detalles de la prueba de funcionamiento.

**Figura 79***Prueba de Funcionamiento de Clientes*

<b>ID Prueba:</b>	PF-05	<b>N° de Sprint:</b>	2	<b>Fecha:</b>	24-01-2023
<b>Nombre del tester del software:</b>	Antony Mantuano				
<b>Módulo:</b>	Clientes				
<b>Objetivo:</b>	Registrar nuevos clientes.				
<b>Acción:</b>	Registrar los datos de un nuevo cliente.				
<b>Resultado esperado:</b>	El usuario registra los datos de un nuevo cliente.				
<b>Resultado obtenido:</b>	El usuario registra correctamente los datos de un nuevo cliente.				
<b>Observaciones:</b>	El usuario recomendó utilizar un servicio que autocomplete los datos del cliente solo con el número de su identificación.				
<b>Se encontro algun error:</b>	SI		NO	X	

*Nota.* La figura presenta los detalles de la prueba de funcionamiento de control de clientes.

Fuente: Elaboración propia.

En las Figuras 80, 81 y 82 se presentan los detalles de las pruebas de funcionamiento del sistema, las cuales corresponden al microservicio de productos.

**Figura 80***Prueba de Funcionamiento de Registro Productos*

<b>ID Prueba:</b>	PF-06	<b>N° de Sprint:</b>	3	<b>Fecha:</b>	24-01-2023
<b>Nombre del tester del software:</b>	Antony Mantuano				
<b>Módulo:</b>	Productos				
<b>Objetivo:</b>	Registrar un nuevo producto.				
<b>Acción:</b>	Registrar los datos de un nuevo producto.				
<b>Resultado esperado:</b>	El usuario registra un nuevo producto.				
<b>Resultado obtenido:</b>	El usuario registra de manera exitosa un nuevo producto, sin embargo al calcular el precio de venta el sistema coloca tres decimales.				
<b>Observaciones:</b>	Corregir el cálculo del precio de venta a solo dos decimales.				
<b>Se encontro algun error:</b>	SI	X	NO		

*Nota.* La figura presenta los detalles de la prueba de funcionamiento de registro de productos.

Fuente: Elaboración propia.

**Figura 81***Prueba de Funcionamiento de Registro Proveedores*

<b>ID Prueba:</b>	PF-07	<b>N° de Sprint:</b>	3	<b>Fecha:</b>	24-01-2023
<b>Nombre del tester del software:</b>	Antony Mantuano				
<b>Módulo:</b>	Productos				
<b>Objetivo:</b>	Registrar un nuevo proveedor.				
<b>Acción:</b>	Registrar los datos de un nuevo proveedor.				
<b>Resultado esperado:</b>	El usuario registra los datos de un nuevo proveedor.				
<b>Resultado obtenido:</b>	El usuario registra correctamente los datos de un nuevo proveedor.				
<b>Observaciones:</b>	El usuario sugiere que ciertos datos como el comentario y página web sean opcionales.				
<b>Se encontro algun error:</b>	SI		NO	X	

*Nota.* La figura presenta los detalles de la prueba de funcionamiento de registro de proveedores. Fuente: Elaboración propia.

### **Figura 82**

#### *Prueba de Funcionamiento de Registro de Inventario*

<b>ID Prueba:</b>	PF-08	<b>N° de Sprint:</b>	3	<b>Fecha:</b>	24-01-2023
<b>Nombre del tester del software:</b>	Antony Mantuano				
<b>Módulo:</b>	Productos				
<b>Objetivo:</b>	Registrar el inventario de un producto agregado.				
<b>Acción:</b>	Registrar el stock de un producto agregado en una bodega específica.				
<b>Resultado esperado:</b>	El usuario registra un nuevo inventario de un producto agregado.				
<b>Resultado obtenido:</b>	El usuario registra de forma correcta un nuevo inventario en una bodega de un producto agregado.				
<b>Observaciones:</b>	Ninguna.				
<b>Se encontro algun error:</b>	SI		NO	X	

*Nota.* La figura presenta los detalles de la prueba de funcionamiento de gestión de inventario.

Fuente: Elaboración propia.

En las Figuras 83 se presentan los detalles de la prueba de funcionamiento del sistema, la cual corresponde al microservicio de ventas.

**Figura 83***Prueba de Funcionamiento de Realizar una Venta*

<b>ID Prueba:</b>	PF-09	<b>N° de Sprint:</b>	4	<b>Fecha:</b>	24-01-2023
<b>Nombre del tester del software:</b>	Antony Mantuano				
<b>Módulo:</b>	Ventas				
<b>Objetivo:</b>	Realizar una venta				
<b>Acción:</b>	Realizar una nueva venta en una caja ya iniciada.				
<b>Resultado esperado:</b>	El usuario realiza una venta.				
<b>Resultado obtenido:</b>	El usuario realiza una venta de forma correcta, pero el cálculo del iva difiere en algunos decimales.				
<b>Observaciones:</b>	Corregir el cálculo del iva en la venta.				
<b>Se encontro algun error:</b>	SI	X	NO		

*Nota.* La figura presenta los detalles de la prueba de funcionamiento del control de ventas.

Fuente: Elaboración propia.

En las Figuras 84 y 85 se presentan los detalles de las pruebas de funcionamiento del sistema, las cuales corresponden al microservicio de facturación.



**Figura 84***Prueba de Funcionamiento de Realizar una Factura*

<b>ID Prueba:</b>	PF-10	<b>N° de Sprint:</b>	5	<b>Fecha:</b>	25-01-2023
<b>Nombre del tester del software:</b>	Antony Mantuano				
<b>Módulo:</b>	Facturación				
<b>Objetivo:</b>	Enviar una factura al SRI en el ambiente de pruebas.				
<b>Acción:</b>	Realizar una factura a partir de una venta generada.				
<b>Resultado esperado:</b>	El usuario crea y envía una factura en ambiente de pruebas al SRI.				
<b>Resultado obtenido:</b>	El usuario genera de forma correcta una factura y la envía al SRI en un ambiente de pruebas.				
<b>Observaciones:</b>	Ninguna.				
<b>Se encontro algun error:</b>	SI		NO	X	

*Nota.* La figura presenta los detalles de la prueba de funcionamiento de realizar factura. Fuente: Elaboración propia.

**Figura 85***Prueba de Funcionamiento de Realizar una Nota de Crédito*

<b>ID Prueba:</b>	PF-11	<b>N° de Sprint:</b>	5	<b>Fecha:</b>	25-01-2023
<b>Nombre del tester del software:</b>	Antony Mantuano				
<b>Módulo:</b>	Facturación				
<b>Objetivo:</b>	Enviar una nota de crédito al SRI en el ambiente de pruebas.				
<b>Acción:</b>	Realizar una nota de crédito a partir de una venta generada.				
<b>Resultado esperado:</b>	El usuario crea y envía una nota de crédito en ambiente de pruebas al SRI.				
<b>Resultado obtenido:</b>	El usuario genera de forma correcta una nota de crédito y la envía al SRI en un ambiente de pruebas.				
<b>Observaciones:</b>	Ninguna				
<b>Se encontro algun error:</b>	SI		NO	X	

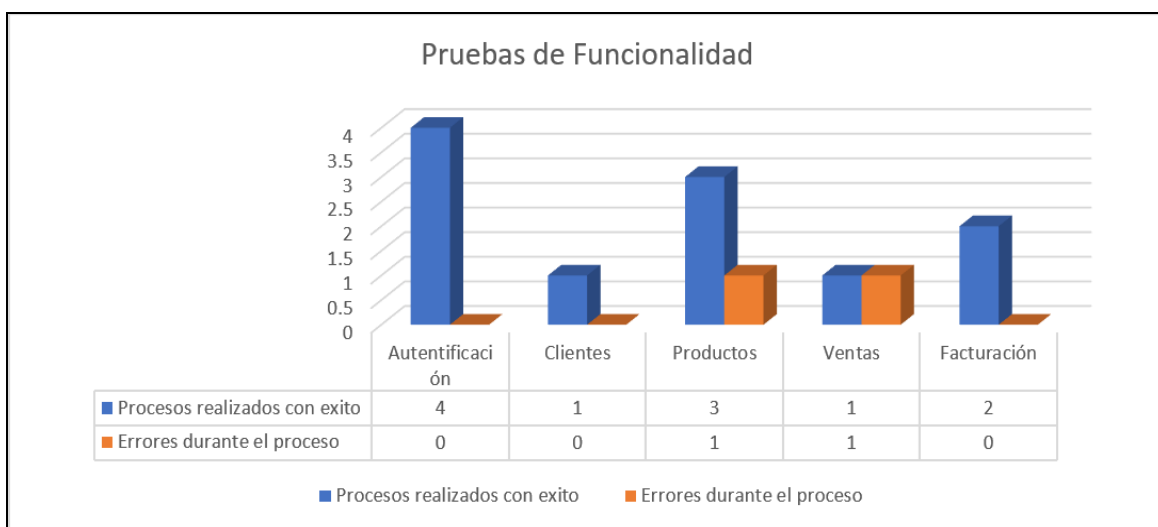
*Nota.* La figura presenta los detalles de la prueba de funcionamiento de realizar una nota de crédito. Fuente: Elaboración propia.

### **Análisis de Resultados Obtenidos**

Después de que el beneficiario principal realizara las pruebas funcionales de los principales módulos del sistema, se obtuvieron los siguientes datos: Del total de 11 pruebas realizadas, sólo 2 presentaron fallos, como muestra la figura 86, lo que significa que, del 100%, el 81,82% fueron pruebas satisfactorias, mientras que el 18,18% presentaron fallos en el sistema.

### **Figura 86**

*Acciones realizadas con éxito y errores de proceso*



*Nota.* La figura presenta los detalles de los procesos realizados con éxito y los errores encontrados durante las pruebas de funcionamiento. Fuente: Elaboración propia.

### **Corrección y Mejoras del Sistema**

Una vez identificados y analizados los errores detectados, se efectúan las correcciones oportunas. En este caso, los errores encontrados están relacionados con el manejo de números decimales, que llevan a cálculos incorrectos a la hora de vender o guardar un

producto. Para comprobar que las correcciones han sido correctas, se vuelven a realizar las pruebas con el beneficiario del sistema.

A continuación, se indicarán los resultados obtenidos luego de realizar las correcciones respectivas de los procesos mencionados.

### Figura 87

#### *Corrección de prueba de registro de producto*

<b>ID Prueba:</b>	PF-06-V2	<b>N° de Sprint:</b>	3	<b>Fecha:</b>	27-01-2023
<b>Nombre del tester del software:</b>	Antony Mantuano				
<b>Módulo:</b>	Productos				
<b>Objetivo:</b>	Registrar un nuevo producto.				
<b>Acción:</b>	Registrar los datos de un nuevo producto.				
<b>Solución implementada:</b>	Se implementó una función que redondea los valores de los diferentes precios a dos decimales en caso de ser necesario.				
<b>Resultado esperado:</b>	El usuario registra un nuevo producto.				
<b>Resultado obtenido:</b>	El usuario registra de manera exitosa un nuevo producto.				
<b>Observaciones:</b>	Ninguna.				
<b>Se encontro algun error:</b>	SI		NO	X	

*Nota.* La figura presenta la corrección de la prueba de funcionamiento para registrar un producto al sistema. Fuente: Elaboración propia.

## Figura 88

### Corrección de prueba de realizar una venta

<b>ID Prueba:</b>	PF-09-V2	<b>N° de Sprint:</b>	4	<b>Fecha:</b>	27-01-2023
<b>Nombre del tester del software:</b>	Antony Mantuano				
<b>Módulo:</b>	Ventas				
<b>Objetivo:</b>	Realizar una venta				
<b>Acción:</b>	Realizar una nueva venta en una caja ya iniciada.				
<b>Solución implementada:</b>	Se implementó una función que redondea los valores del cálculo del iva a dos decimales.				
<b>Resultado esperado:</b>	El usuario realiza una venta.				
<b>Resultado obtenido:</b>	El usuario realiza una venta de forma correcta.				
<b>Observaciones:</b>	Ninguna.				
<b>Se encontro algun error:</b>	SI		NO	X	

*Nota.* La figura presenta la corrección de la prueba de funcionamiento para realizar una venta.

Fuente: Elaboración propia.

## Trabajos Futuros

Una vez que el sistema está en marcha y es utilizado por los beneficiarios, se pone de manifiesto que aún hay mucho margen de mejora y desarrollo, por lo que en el futuro podrán realizarse diversas actualizaciones, en función del enfoque y los ajustes que la empresa quiera dar al sistema. Algunas de las mejoras que pueden aplicarse son:

- Ampliar y mejorar la funcionalidad de las notificaciones para que no sean solo correos electrónicos, sino que también tengan la capacidad de enviar mensajes a una aplicación de mensajería instantánea para smartphones.
- Completar el módulo de recibos electrónicos añadiendo los documentos que faltan, como guías de remisión, notas de débito y retenciones.

- Crear más roles de usuario para realizar tareas específicas dentro del sistema y mejorar la seguridad y gestión del mismo.
- Implementar la funcionalidad de lector de códigos de barras o códigos QR cuando proceda.
- En función del alcance de la empresa, añadir un módulo de contabilidad para ayudar a gestionar los activos y pasivos y mejorar los controles financieros.

## Capítulo V: Conclusiones y Recomendaciones

### Conclusiones

Por lo tanto, una vez culminado el presente trabajo, el resultado obtenido fue un sistema web escalable desarrollado sobre la base de una arquitectura de microservicios que puede adaptarse fácilmente a cualquier negocio retail. Con ello proporcionar todas las funciones básicas necesarias para una gestión adecuada, desde el control de inventarios y el control de ventas hasta los métodos de facturación electrónica y la visualización de informes.

A la hora de recopilar y analizar los requerimientos funcionales y no funcionales, se determinó que una comunicación adecuada entre los desarrolladores y el cliente es esencial para comprender mejor la lógica de negocio de la empresa. Utilizando el marco de trabajo ágil SCRUM, fue posible estructurar adecuadamente las tareas y actividades a realizar en el plazo establecido, obteniendo un feedback tras cada sprint y mejorando así el resultado final de la solución tecnológica.

El uso de Angular permitió una forma modular de trabajar con componentes reutilizables, obteniendo un código limpio y fácil de mantener. Es una gran opción para crear aplicaciones robustas y escalables, gracias a su popularidad y vasta documentación. Durante el desarrollo del backend, se eligió Java con el framework Spring Boot para la generación y envío de comprobantes y Node JS para implementar los distintos microservicios. El uso de esta arquitectura ofrece mayor flexibilidad, ya que cada módulo del sistema puede desarrollarse de forma independiente, también cabe destacar que tiene una mejor tolerancia a fallos, ya que, en caso de fallo de un microservicio, sólo se ve afectada una parte de la aplicación, evitando que todo el sistema se colapse.

Durante el desarrollo del backend, se eligió Java con el framework Spring Boot para la generación y envío de comprobantes y Node JS para implementar los distintos microservicios. El uso de esta arquitectura ofrecía mayor flexibilidad, ya que cada módulo del sistema podía desarrollarse de forma independiente en función de los requisitos a resolver. Además de ser

una solución que facilita la escalabilidad implementando nuevas mejoras o actualizaciones para el microservicio concreto que las necesite, también cabe destacar que tiene una mejor tolerancia a fallos, ya que, en caso de fallo de un microservicio, sólo se ve afectada una parte de la aplicación, evitando que todo el sistema se colapse, lo que mejora la experiencia del cliente.

Se realizaron diferentes pruebas para corroborar el correcto funcionamiento del sistema entre las que se evaluó la calidad del código, obteniendo un aprobado en la mayoría de las métricas concluyendo que se escribió un código limpio y mantenible. En las pruebas de carga se evidencio el rendimiento de cada microservicio, los resultados obtenidos indican que el sistema podrá funcionar correctamente, además de corroborar que en caso de existir un mayor tráfico en las peticiones de las APIs se puede escalar cada microservicio de forma sencilla. En lo que se refiere a las pruebas de funcionalidad se evaluaron los módulos principales del sistema descubriendo errores que no se tenían en consideración en el momento del desarrollo, obteniendo recomendaciones de cómo solucionarlos.

Para desplegar cada microservicio, se utilizó Kubernetes, para gestionar cada microservicio, establecer los recursos, el número de réplicas, la seguridad y los servicios de redirección. El proveedor Linode dispone de amplia documentación para el uso de sus servicios, lo que facilita la fase de despliegue y mantenimiento.

## **Recomendaciones**

Aunque los microservicios tienen la ventaja de ser fácilmente escalables, en términos de recursos es necesario evaluar la infraestructura utilizada, ya que mayores recursos significan alquilar un servicio más caro, por lo que es necesario analizar y determinar qué microservicio necesita ser mejorado.

Si bien el sistema web está diseñado para adaptarse a dispositivos pequeños como tabletas o teléfonos, no todas las funcionalidades están disponibles u optimizadas, por lo que,

si el sistema se va a utilizar en dispositivos móviles, es necesario desarrollar una versión optimizada para dichos dispositivos.

En caso de actualización de la normativa fiscal o similar, se deberá crear una nueva versión del sistema web, ya que estos datos no están parametrizados, como la inclusión de nuevos impuestos, formas de pago, recibos, entre otros.

Las tecnologías de la información se han convertido en una herramienta fundamental en los negocios, por lo que es muy importante la adopción de estas herramientas en los negocios retail, con ello ser más eficiente en sus actividades y elevar el nivel de servicio al cliente. Con la implementación de estos sistemas controlar de mejor manera el flujo de información que necesita el personal.



## Bibliografía

- Adamczyk, P., & Smith, P. (2011). *REST and Web Services: In Theory and in Practice*.  
doi:10.1007/978-1-4419-8303-9\_2
- Anrango, L. M., & Romero, R. V. (2019). *Sistema de facturación electrónica para la web de la distribuidora Mateito*. Quito. Obtenido de  
<https://dspace.ups.edu.ec/handle/123456789/16913>
- Barber, S., & Sime, A. (2017). *Load Testing Your Website: How to Make Sure Your Site Can Handle Traffic*.
- Barrios, D. A. (2018). *Arquitectura de Microservicios*. Bogotá. Obtenido de  
<https://revistas.udistrital.edu.co/index.php/tia/article/view/9687>
- Barrios, D. (2015). *Modelado Bajo una Arquitectura de Microservicios de un Prototipo de Software, para el Comercio de Tiendas de Barrio*. Bogotá. Obtenido de  
<https://repository.udistrital.edu.co/handle/11349/4794>
- Canqui, J. A. (2021). *Microservicios como estrategia en la facturación electrónica*. La Paz. Obtenido de <https://repositorio.umsa.bo/handle/123456789/28903>
- Castro, D. B. (2018). *PROCEDIMIENTO PARA UTILIZAR EL FRAMEWORK ANGULAR EN LA CREACIÓN DE APLICACIONES WEB MODERNAS SPA*. Pamplona. Obtenido de  
[http://repositoriodspace.unipamplona.edu.co/jspui/bitstream/20.500.12744/5540/1/CASTRO\\_2018\\_TG.pdf](http://repositoriodspace.unipamplona.edu.co/jspui/bitstream/20.500.12744/5540/1/CASTRO_2018_TG.pdf)
- Dini, M., Gligo, N., & Patiño, A. (2021). *Transformación digital de las mipymes: Elementos para el diseño de políticas*. Santiago: Documentos de Proyectos (LC/TS.2021/99). Obtenido de <https://hdl.handle.net/11362/47183>
- Fisher, J. (2013). *Utilizing Atlassian Jira For Large-Scale Software Development Management*. San Francisco. Obtenido de <https://www.osti.gov/servlets/purl/1097750>
- Gibert, M., & Pérez, O. (2012). *Bases de datos en PostgreSQL*. Obtenido de  
<https://artpatcusco.com/sis/pdf/20140527172742en.pdf>

- Gracia, H., & Yanes, O. (2013). *Bases de datos NoSQL*. Obtenido de <https://revistatelematica.cujae.edu.cu/index.php/tele/article/view/74>
- Guerrero, D. G. (2012). *Factores clave de éxito en el negocio del retail*. México. Obtenido de [https://revistas.ulima.edu.pe/index.php/Ingenieria\\_industrial/article/view/223](https://revistas.ulima.edu.pe/index.php/Ingenieria_industrial/article/view/223)
- Haro, E., & Guarda, T. (2018). *Desarrollo backend para aplicaciones web, Servicios Web Restful: Node.js vs Spring Boot*. Quito. Obtenido de <https://www.proquest.com/openview/a78cfaa62708fd24f38ac8d1025050eb/1?pq-origsite=gscholar&cbl=1006393>
- Linode. (2022). *Linode*. Obtenido de Métricas de comparación propuestas por el sitio web de Linode: <https://www.linode.com/es/compare/>
- Márques, M. (2011). *Bases de datos*. Sapientia. Obtenido de <https://repositori.uji.es/xmlui/bitstream/handle/10234/24183/s18.pdf?sequence=6&isAllowed=y>
- Memon, P., & Tahseen, H. (2018). *Comparative Study of Testing Tools Blazemeter and Apache Jmeter*. Obtenido de <http://journal.iba-suk.edu.pk:8089/sibajournals/index.php/sjcms/article/view/66/91>
- Muñoz, J. L. (2017). *Tecnología de Contenedores Docker*. Barcelona. Obtenido de [https://upcommons.upc.edu/bitstream/handle/2117/113040/Degree\\_thesis.pdf?sequence=1&isAllowed=y](https://upcommons.upc.edu/bitstream/handle/2117/113040/Degree_thesis.pdf?sequence=1&isAllowed=y)
- Ortiz, K., Guevera, J., Santos, J., Alcántara, O., Tenorio, J., & Sánchez, R. (2022). *Aplicación web con arquitectura de microservicios y el incremento de la eficiencia en el comercio electrónico de una empresa peruana de calzado*. Trujillo. Obtenido de <https://www.iiiis.org/CDs2022/CD2022Summer/papers/CA313DC.pdf>
- Owasp. (2021). *Owasp*. Obtenido de Los diez mejores de OWASP: <https://owasp.org/www-project-top-ten/>

- Pozo, M. (2021). *AWS: El mayor proveedor de Cloud para Startups*. Sevilla. Obtenido de <https://idus.us.es/handle/11441/127987>
- Primicias. (Junio de 2022). *Facturación electrónica es obligatoria desde noviembre de 2022*. Obtenido de Primicias: <https://www.primicias.ec/noticias/economia/facturacion-electronica-obligatoria-noviembre/>
- Primicias. (Febrero de 2022). *Primicias*. Obtenido de Pymes en Ecuador invierte en tecnología: <https://www.primicias.ec/noticias/economia/pymes-ecuador-inversion-tecnologia-pandemia/>
- Puciarelli, L. (2020). *Node JS-Vol. 1: Instalación-Arquitectura-node y npm (Vol. 1)*. Buenos Aires: Plandos.
- Ramos, A., & Recalde, P. (2022). *Balanceo y despliegue de carga en aplicaciones web mediante kubernetes*. Quito. Obtenido de <https://revista.uisrael.edu.ec/index.php/ro/article/view/585/608>
- Richardson, A. (2020). *API Testing with Postman*.
- Rodriguez, A. (2008). *RESTful Web services: The basics*. Obtenido de <https://cs.calvin.edu/courses/cs/262/kvlinden/references/rodriguez-restfulWS.pdf>
- Rojas, W., & Silva, M. (2016). *Introducción a Java: Guía de actividades prácticas*. Bogota.
- Schwaber, K., & Sutherland, J. (2020). *La Guía de Scrum*. Obtenido de <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>
- Sentrio. (12 de 2021). *Qué es SonarQube: Verifica y analiza la calidad de tu código*. Obtenido de Sentrio: [https://sentrio.io/blog/que-es-sonarqube/#%C2%BFQue\\_es\\_SonarQube](https://sentrio.io/blog/que-es-sonarqube/#%C2%BFQue_es_SonarQube)
- Silva, J. A. (2021). *Análisis de contenedores Docker y sus implicaciones de seguridad*. Madrid. Obtenido de [https://www.researchgate.net/profile/Javier-Alonso-Silva-2/publication/352882918\\_Analisis\\_de\\_contenedores\\_Docker\\_y\\_sus\\_implicaciones\\_de\\_](https://www.researchgate.net/profile/Javier-Alonso-Silva-2/publication/352882918_Analisis_de_contenedores_Docker_y_sus_implicaciones_de_)

seguridad/links/60ddf1e992851ca9449eade2/Analisis-de-contenedores-Docker-y-sus-implicaciones-de-seguridad.pdf

Urteaga, A. (2015). *Aplicación de la metodología de desarrollo ágil Scrum para el desarrollo de un sistema de gestión de empresas*. Madrid. Obtenido de <https://e-archivo.uc3m.es/handle/10016/23750>

Wolf, E. (2019). *Testing Microservices: Understanding and Improving the Performance of Microservices*.