



UNIVERSIDAD
CATÓLICA
DE CUENCA

UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**UNIDAD ACADÉMICA DE INFORMÁTICA, CIENCIAS DE
LA COMPUTACIÓN E INNOVACIÓN TECNOLÓGICA**

CARRERA DE INGENIERÍA EN SISTEMA DE INFORMACIÓN

**DESARROLLO DE UN PROTOTIPO DE SOFTWARE PARA LA
GESTIÓN DE INVENTARIOS Y FACTURACIÓN PARA EMPRESAS
DEL SECTOR DE VENTAS Y SERVICIOS DE COMPUTACIÓN CON
ALMACENAMIENTO EN LA NUBE - CASO DE APLICACIÓN
TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN SISTEMAS DE INFORMACIÓN**

AUTOR: SILVIO DAVID PACHECO HERRERA

DIRECTOR: ING. BLANCA LUCIA ÁVILA CORREA, MSC

AZOGUES - ECUADOR

2023

DIOS, PATRIA, CULTURA Y DESARROLLO



UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**UNIDAD ACADÉMICA DE INFORMÁTICA, CIENCIAS DE
LA COMPUTACIÓN E INNOVACIÓN TECNOLÓGICA**

CARRERA DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN

**TEMA: DESARROLLO DE UN PROTOTIPO DE SOFTWARE PARA LA
GESTIÓN DE INVENTARIOS Y FACTURACIÓN PARA EMPRESAS DEL
SECTOR DE VENTAS Y SERVICIOS DE COMPUTACIÓN CON
ALMACENAMIENTO EN LA NUBE - CASO DE APLICACIÓN**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN SISTEMAS DE INFORMACIÓN**

AUTOR: SILVIO DAVID PACHECO HERRERA

DIRECTOR: ING. BLANCA LUCIA ÁVILA CORREA, MSC

AZOGUES- ECUADOR

2023

DIOS, PATRIA, CULTURA Y DESARROLLO

Declaratoria de Autoría y Responsabilidad

Silvio David Pacheco Herrera portador de la cédula de ciudadanía N.º **0150041572**. Declaro ser el autor de la obra: “**Desarrollo de un Prototipo de Software para la Gestión de Inventarios y Facturación para Empresas del Sector de Ventas y Servicios de Computación con Almacenamiento en la Nube - Caso de Aplicación**”, sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaro que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Azogues, **24 de enero de 2023**

F: 

Silvio David Pacheco Herrera

C.I. 0150041572

CERTIFICACIÓN

Yo, Blanca Lucía Ávila Correa Docente de la Unidad Académica de Informática, Ciencias de la Computación e Innovación Tecnológica a petición de parte interesada:

Certifico que el estudiante **Silvio David Pacheco Herrera** con C.I. **0150041572** ha realizado el trabajo de titulación “**DESARROLLO DE UN PROTOTIPO DE SOFTWARE PARA LA GESTIÓN DE INVENTARIOS Y FACTURACIÓN PARA EMPRESAS DEL SECTOR DE VENTAS Y SERVICIOS DE COMPUTACIÓN CON ALMACENAMIENTO EN LA NUBE - CASO DE APLICACIÓN**”, previo a la obtención del título de Ingeniero de Sistemas, de la Unidad Académica de Informática Ciencias de la Computación en Innovación Tecnológica, el mismo que ha estado bajo mi supervisión y cumple con todas las observaciones realizadas por el tribunal evaluador; por lo que las ideas u opiniones vertidas en el mismo son de exclusiva responsabilidad del autor

Es todo cuanto puedo certificar en honor a la verdad, autorizando al peticionario, hacer del presente documento, el uso que creyere conveniente

F:

Ing. Blanca Lucia Ávila Correa MSC.

DOCENTE DE LA U. A. ICCIT

UNIVERSIDAD CATÓLICA DE CUENCA.

DEDICATORIA

El presente trabajo de titulación, con mucha gratitud dedico primeramente a Dios quien me ha guiado en este camino dándome la fuerza y capacidad para seguir siempre adelante.

A mi hijo, Cristian David Pacheco Gavilanes, quien me enseñó a jamás darme por vencido siendo mi principal fuente de inspiración en todo momento.

A mi esposa, Alicia Daniela Gavilanes Pinos, quien siempre me apoyó en todo momento y por estar siempre conmigo siendo parte fundamental de mi vida.

A mis padres, Silvio Enrique Pacheco Narváez y Giovanna Katerine Herrera Hugo, por creer en mí siempre, por ser los promotores de mis sueños permitiéndome cumplirlos y por ser el pilar fundamental en mi vida.

A mi hermana, Galilea Katerine Pacheco Herrera, por siempre apoyarme y quererme de manera incondicional.

AGRADECIMIENTO

Agradezco primeramente a Dios, a mis padres, mi esposa, mi hijo, mi hermana y todos mis familiares quienes me apoyaron en este camino de mi vida universitaria.

A mi tutor de tesis, Ing. Blanca Lucia Ávila Correa, por sus enseñanzas transmitidas, por su orientación, paciencia y colaboración para este trabajo de titulación.

A los docentes de la carrera de Ingeniería en Sistemas de Información de la Universidad Católica de Cuenca Campus Azogues por haber compartido sus conocimientos a lo largo de la carrera y actuar con rectitud y respeto siempre.

RESUMEN

La empresa Servcomp ubicada en los cantones Cañar y El Tambo de la provincia de Cañar brinda servicios de venta, actualización, reparación y mantenimiento de equipos informáticos.

El proceso de ventas de la empresa es llevado a cabo mediante un sistema legado de facturación que lleva sus registros en una base de datos local, además la situación actual de la facturación en Ecuador obliga a la empresa a adaptarse al proceso de facturación electrónica.

Dichas situaciones hacen que no se disponga de información del inventario y del proceso de ventas y facturación de manera centralizada. De otra parte, trabajar con un sistema que no tiene soporte es impide la implementación de emisión de comprobantes electrónicos.

Es por ello que se decide desarrollar una aplicación web para la facturación y gestión de inventario que almacene su información en la nube en una sola fuente de datos y que permita a la empresa emitir comprobantes electrónicos a sus clientes.

Palabras Clave: almacenamiento en la nube, SCRUM, facturación electrónica, desarrollo web

ABSTRACT

Servcomp company, located in Cañar and EI Tambo cantons in the province of Cañar, provides sales services, updating, repair, and maintenance services.

The company's sales process is conducted through a legacy invoicing system that keeps its records in a local database. In addition, the current situation of invoicing in Ecuador forces the company to adapt to the electronic invoicing process.

These situations mean that information on inventory and the sales and invoicing process is not available centrally. On the other hand, working with an unsupported system prevents the implementation of the electronic procedure.

For this, it was decided to develop a web-based billing and inventory management application that stores information in the Cloud in a single data source and allows the company to issue electronic receipts to its customers.

Keywords: cloud storage, SCRUM, electronic invoicing, web development

ÍNDICE DE CONTENIDO

CERTIFICACIÓN	8
DEDICATORIA	9
AGRADECIMIENTO	10
RESUMEN	11
ABSTRACT	12
ÍNDICE DE CONTENIDO.....	13
ÍNDICE DE FIGURAS	16
ÍNDICE DE TABLAS	17
INTRODUCCIÓN	18
CAPÍTULO I.....	19
1.1. Planteamiento del Problema.....	19
1.2. Justificación	20
1.3. Objetivos	21
1.3.1. Objetivo General	21
1.3.2. Objetivos Específicos.....	21
1.4. Alcance	21
CAPÍTULO II.....	23
2.1. Desarrollo de Software.....	23
2.1.1. Metodologías Tradicionales	24
2.1.2. Metodologías Ágiles	24
2.1.3. Metodologías Híbridas	24
2.2. SCRUM.....	25
2.2.1. Proceso de SCRUM	26
2.2.2. Actividades de SCRUM	26
2.2.3. Herramientas de SCRUM.....	27
2.2.4. Roles y Responsabilidades de SCRUM.....	27
2.3. Facturación Electrónica	28
2.3.1. Facturación Electrónica en Ecuador	29
2.3.1.1. Proceso	29
2.4. Cloud Computing.....	32
2.4.1. Modelos de Cloud Computing.....	32
2.4.2. Servicios de Cloud Computing.....	33
2.5. Almacenamiento en la nube	34
2.6. Aplicaciones Web	35

2.7.	Herramientas de Desarrollo Web.....	36
2.7.1.	Editor de código	37
2.7.1.1.	Visual Studio Code.....	38
2.7.2.	Frontend	38
2.7.2.1.	Angular	39
2.7.2.1.1.	Arquitectura	39
2.7.3.	Backend	40
2.7.3.1.	Node.js.....	41
2.7.3.2.	Firebase	43
2.7.3.2.1.	Firestore Cloud Database	43
2.7.3.2.2.	Firebase Authentication	44
2.7.3.2.3.	Firebase Hosting.....	45
CAPÍTULO III.....		46
3.1.	Desarrollo	46
3.2.	Planificación del Sprint	46
3.2.1.	Definición de los Sprint.....	48
3.3.	Ejecución del Sprint.....	51
3.3.1.	Lista de Tareas y Burndown Chart.....	52
3.4.	Incremento	53
3.4.1.	Sprint 1	53
3.4.1.1.	Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal	54
3.4.1.2.	Gestión de Proveedores	57
3.4.1.3.	Gestión de Clientes.....	59
3.4.2.	Sprint 2	61
3.4.2.1.	Gestión de Inventario.....	63
3.4.2.2.	Gestión de Kárdex	65
3.4.3.	Sprint 3	68
3.4.3.1.	Facturación Electrónica	69
3.4.3.2.	Crear el Dashboard.....	73
3.4.4.	Sprint 4	75
3.4.4.1.	Crear el Login.....	76
3.4.4.2.	Autenticación y Roles de Usuario.....	79
3.4.4.3.	Publicar la Aplicación Web	82
3.5.	Revisión y Retroalimentación.....	84
3.6.	Cierre	85

CAPÍTULO IV	87
4.1. Conclusiones.....	87
4.2. Recomendaciones	87
REFERENCIAS BIBLIOGRÁFICAS	89
ANEXOS	92
Anexo 1: Componente “Clientes”	92
Anexo 2: Servicio “Clientes”	94
Anexo 3: Componente “Proveedores”	94
Anexo 4: Servicio “Proveedores”	96
Anexo 5: Componente “Productos”	97
Anexo 6: Servicio “Productos”	98
Anexo 7: Componente “Kárdex”	99
Anexo 8: Servicio “Kárdex”	102
Anexo 9: Componente “Facturación”	103
Anexo 10: Servicio “Facturación”	108
Anexo 11: Componente “Main”	109
Anexo 12: Servicio “Dashboard”	113
Anexo 13: Estructura XML.....	113
Anexo 14: Roles de usuario	117
Anexo 15: Emisión del comprobante electrónico	118
Anexo 16: Carta de aceptación	120

ÍNDICE DE FIGURAS

FIG. 1 PROCESO DE SCRUM.....	26
FIG. 2 ARQUITECTURA DE UNA APLICACIÓN WEB.....	35
FIG. 3 ARQUITECTURA DE ANGULAR.....	40
FIG. 4 ENCUESTA DE STACKOVERFLOW SOBRE LOS FRAMEWORKS Y TECNOLOGÍAS WEB MÁS USADAS EN EL AÑO 2022.....	41
FIG. 5 ENCUESTA DE STACKOVERFLOW SOBRE PLATAFORMAS DE ALMACENAMIENTO EN LA NUBE.....	42
FIG. 6 BURNDOWN CHART INICIAL.....	52
FIG. 7 BURNDOWN CHART DEL SPRINT 1.....	54
FIG. 8 MODELO DE DATOS INICIAL.....	55
FIG. 9 COLECCIONES EN FIRESTORE DATABASE.....	56
FIG. 10 AVANCE DEL BURNDOWN CHART DEL SPRINT 1.....	57
FIG. 11 INTERFAZ "PROVEEDORES".....	57
FIG. 12 INTERFAZ PARA REGISTRAR UN NUEVO PROVEEDOR.....	58
FIG. 13 AVANCE DEL BURNDOWN CHART DEL SPRINT 1.....	59
FIG. 14 INTERFAZ "CLIENTES".....	59
FIG. 15 INTERFAZ PARA REGISTRAR UN NUEVO CLIENTE.....	60
FIG. 16 BURNDOWN CHART FINAL DEL SPRINT 1.....	61
FIG. 17 BURNDOWN CHART SPRINT 2.....	62
FIG. 18 INTERFAZ "PRODUCTOS".....	63
FIG. 19 INTERFAZ PARA AGREGAR UN NUEVO PRODUCTO.....	63
FIG. 20 AVANCE DEL BURNDOWN CHART DEL SPRINT 2.....	65
FIG. 21 INTERFAZ "KÁRDEX".....	65
FIG. 22 INTERFAZ PARA RECIBIR O RETIRAR UNIDADES EN EL STOCK DE UN PRODUCTO.....	66
FIG. 23 INTERFAZ DEL KÁRDEX DE UN PRODUCTO.....	66
FIG. 24 BURNDOWN CHART FINAL DEL SPRINT 2.....	68
FIG. 25 BURNDOWN CHART SPRINT 3.....	69
FIG. 26 INTERFAZ INICIAL "FACTURACIÓN".....	70
FIG. 27 INTERFAZ PARA SELECCIONAR UN CLIENTE Y PRODUCTOS A FACTURAR.....	70
FIG. 28 INTERFAZ DE FACTURACIÓN.....	71
FIG. 29 INTERFAZ DE LA LISTA DE VENTAS REALIZADAS.....	71
FIG. 30 XML.....	72
FIG. 31 AVANCE DEL BURNDOWN CHART DEL SPRINT 3.....	73
FIG. 32 INTERFAZ DEL DASHBOARD.....	73
FIG. 33 BURNDOWN CHART FINAL DEL SPRINT 3.....	75
FIG. 34 BURNDOWN CHART SPRINT 4.....	76
FIG. 35 INTERFAZ INICIO DE SESIÓN.....	77
FIG. 36 PANEL DE MENÚ.....	78
FIG. 37 AVANCE DEL BURNDOWN CHART DEL SPRINT 4.....	79
FIG. 38 CREACIÓN DE USUARIOS.....	80
FIG. 39 LISTA DE USUARIOS.....	80
FIG. 40 USUARIOS REGISTRADOS EN LA BASE DE DATOS.....	80
FIG. 41 AVANCE DEL BURNDOWN CHART DEL SPRINT 4.....	82
FIG. 42 PRODUCCIÓN DE LA APLICACIÓN.....	82
FIG. 43 DESPLIEGUE DE LA APLICACIÓN.....	83
FIG. 44 PÁGINA WEB.....	83
FIG. 45 BURNDOWN CHART FINAL DEL SPRINT 4.....	84

ÍNDICE DE TABLAS

TABLA I METODOLOGÍAS TRADICIONALES VS ÁGILES. FUENTE: AUTOR	25
TABLA II DIGITOS PARA LA CLAVE DE ACCESO.	31
TABLA III TIPO DE EMISIÓN.	31
TABLA IV TIPOS DE COMPROBANTE.....	31
TABLA V HERRAMIENTAS PARA EL DESARROLLO WEB.	37
TABLA VI PRODUCT BACKLOG.....	47
TABLA VII SPRINT 1.....	48
TABLA VIII SPRINT 2.....	49
TABLA IX SPRINT 3.....	49
TABLA X SPRINT 4.....	50
TABLA XI PLANIFICACIÓN SPRINT 1.....	50
TABLA XII PLANIFICACIÓN SPRINT 2.....	50
TABLA XIII PLANIFICACIÓN SPRINT 3.....	51
TABLA XIV PLANIFICACIÓN SPRINT 4.	51
TABLA XV LISTA DE TAREAS INICIAL.	52
TABLA XVI LISTA DE TAREAS SPRINT 1.	53
TABLA XVII AVANCE DE LA LISTA DE TAREAS SPRINT 1.....	56
TABLA XVIII AVANCE DE LA LISTA DE TAREAS SPRINT 1.....	58
TABLA XIX LISTA DE TAREAS FINAL DEL SPRINT 1.	60
TABLA XX LISTA DE TAREAS SPRINT 2.	62
TABLA XXI AVANCE DE LA LISTA DE TAREAS SPRINT 2.	64
TABLA XXII LISTA DE TAREAS FINAL SPRINT 2:	67
TABLA XXIII LISTA DE TAREAS SPRINT 3.	68
TABLA XXIV AVANCE DE LA LISTA DE TAREAS SPRINT 3.....	72
TABLA XXV LISTA DE TAREAS FINAL SPRINT 3.	74
TABLA XXVI LISTA DE TAREAS SPRINT 4.	75
TABLA XXVII AVANCE DE LA LISTA DE TAREAS SPRINT 4.....	78
TABLA XXVIII AVANCE DE LA LISTA DE TAREAS SPRINT 4.....	81
TABLA XXIX LISTA DE TAREAS FINAL SPRINT 4.	83
TABLA XXX INFORME DE ENTREGABLE FINAL.....	85

INTRODUCCIÓN

El uso de las tecnologías de la información dentro de las organizaciones, entidades públicas o privadas, centros comerciales, entre otros, cumplen un papel fundamental por la cantidad de herramientas existentes como los sistemas de información o aplicaciones que cuentan con ciertas características que permiten a los usuarios dar solución a problemas y de alguna forma mejorar la actividad laboral y productiva dentro de una organización.

Con el paso del tiempo han aparecido nuevas tecnologías y nuevas formas de llevar a cabo un proceso. En el caso de la facturación, se realizaba este proceso mediante la emisión de comprobantes físicos. Hoy en día la mayoría de las empresas deben cumplir con la obligación de adaptar su negocio al proceso de facturación electrónica.

Para las entidades o negocios que se dedican a la venta de un servicio al público es importante cumplir con los requisitos necesarios, sobre todo si es una empresa que lleva varios años en el mercado. De igual manera es importante llevar un registro de las ventas y adquisiciones que se hagan.

La presente investigación se enfoca en el desarrollo de una aplicación web de facturación y gestión de inventarios con almacenamiento en la nube para empresas del medio dedicadas a la venta de servicios de computación, en este caso la empresa Servcomp.

El desarrollo de este trabajo de titulación está dividido en cuatro capítulos: El Capítulo I describe el planteamiento del problema, la justificación, los objetivos y el alcance de este trabajo. El Capítulo II describe las bases teóricas que describen el contexto del trabajo de titulación como el almacenamiento en la nube, facturación electrónica, y la metodología de desarrollo de software a usar. En el Capítulo III se aplica la metodología de desarrollo descrita en el Capítulo II para el desarrollo de la solución. Finalmente, el Capítulo IV se presentan las conclusiones y recomendaciones relacionadas al trabajo de titulación.

CAPÍTULO I

MARCO REFERENCIAL

1.1. Planteamiento del Problema

Según la división política actual la provincia de Cañar está dividida en 7 cantones: Azogues, Biblián, Cañar, La Troncal, El Tambo, Déleg y Suscal, cada cantón tiene sus diferentes parroquias.

Dentro de los cantones de Cañar y El Tambo existen diferentes actividades que representan el sustento de las familias como el establecimiento de una microempresa que en su mayoría son empresas locales que surgen de la actividad familiar.

Las microempresas que brindan productos o servicios al público, en este caso de venta de servicios de computación, en su mayoría no hacen uso de un producto de software y trabajan de manera tradicional llevando registros, contabilidad y gestión de sus ventas en archivos planos o cuadernos. Esto genera problemas como la poca efectividad en las operaciones empresariales, disponibilidad nula de la información, desventaja competitiva, etc.

Los sistemas de información están diseñados a partir de las necesidades de un negocio lo que permite que una empresa mejore de manera operativa y organizacional en sus labores.

Algunas de las empresas del medio se encuentran en proceso de madurez a punto de que cuentan con una o varias sucursales en diferentes cantones o ciudades, un caso de ello es la empresa Servcomp que se encuentra establecida en los cantones Cañar y El Tambo y que se dedica a la venta, actualización, reparación y mantenimiento de equipos informáticos. Por lo que es necesario tener una fuente de datos centralizada por medio del almacenamiento en la nube para poseer una mayor disponibilidad de información.

De acuerdo con lo mencionado anteriormente, surge el problema a resolverse: ¿Es factible desarrollar un sistema que realice gestión de inventarios, ventas y facturación en la empresa Servcomp utilizando el almacenamiento en la nube para conseguir la integridad de la información y mejorar la eficiencia de la empresa?

1.2. Justificación

Los sistemas de información son herramientas que son consideradas de gran importancia y que han permitido gestionar, los datos e información del mismo. Cuentan con diferentes características o funciones que trabajan de manera conjunta para recoger, guardar, juntar y gestionar información, registros históricos, listados, reportes y otros dependiendo de las necesidades del usuario.

La implementación del almacenamiento en la nube es inmediata a través de conexión a internet y por medio de un proveedor de servicios de la nube, por ejemplo, Amazon Web Services que se encarga de brindar diferentes servicios de Cloud Computing como servidores virtuales en la nube, servicios de bases de datos relacionales y NoSQL, almacenamiento en la nube, y otros. Los beneficios de usar el almacenamiento en la nube es que existe una mayor seguridad, se pueden crear y configurar instancias de los servicios que se requieran, proporciona escalabilidad y flexibilidad en los servicios, todo esto dependiendo de las necesidades y el presupuesto del usuario.

Para el presente proyecto se desarrollará un prototipo de software de gestión de inventarios y facturación que permitirá a las empresas del sector guardar información de clientes, productos, proveedores, ventas, y otros, emitir comprobantes electrónicos, presentar dashboards simples que servirán de información al usuario y mejorar la integridad de la información con una base de datos única en la nube permitiendo tener disponibilidad de información y escalabilidad dependiendo de la información que se almacene. Además, se

diseñará una interfaz amigable con el usuario e intuitiva para que su uso no presente ninguna dificultad.

Esto se realizará debido a que en el cantón Cañar y El Tambo la empresa Servcomp la cual se dedica a la venta de servicios de computación no cuenta con una solución para mejorar su eficiencia e integridad de la información y tampoco maneja una fuente de datos única.

1.3. Objetivos

1.3.1. Objetivo General

Construir un prototipo de software de inventario y facturación para la empresa Servcomp con almacenamiento en la nube - caso de aplicación.

1.3.2. Objetivos Específicos

- Realizar la fundamentación teórica sobre desarrollo de software y almacenamiento en la nube.
- Desarrollar un prototipo funcional que permita realizar facturación, ventas y gestión de inventarios.
- Validar el prototipo de software.

1.4. Alcance

El presente proyecto consiste en desarrollar un prototipo de software de gestión de inventarios y facturación para las empresas del medio dedicadas a la venta de servicios de computación la cual hará uso de almacenamiento en la nube para guardar, administrar y centralizar la información que se genere, con el objetivo de solventar las necesidades existentes que se definieron anteriormente, para ello se pretende como primera instancia hacer un cuadro comparativo de las metodologías de desarrollo de software para determinar la que mejor se adapte al proyecto, posteriormente se diseñará y construirá el prototipo de software y se

levantará la instancia en la nube para el almacenamiento, y finalmente se validará el prototipo final por medio de la interacción del usuario en la empresa Servcomp del cantón Cañar y El Tambo.

Para el proceso de desarrollo de la interfaz de usuario, los módulos y el funcionamiento del prototipo de software se hará uso de metodologías ágiles de desarrollo las cuales se basan en la construcción y presentación en poco tiempo de un producto de software mediante un enfoque flexible y buscando la mejor organización y distribución del tiempo posible para alcanzar el objetivo propuesto.

CAPÍTULO II

MARCO TEÓRICO

2.1. Desarrollo de Software

El desarrollo de software es un proceso enfocado en adquirir las necesidades o requerimientos de un usuario y plasmarlos en una aplicación o sistema con el objetivo de solventar sus problemas o dificultades.

La ingeniería de software consiste en la aplicación del saber científico al diseño y composición de programas de computadora y el expediente que se necesita para desarrollarlos, manipularlos y conservarlos independientemente de la metodología que se implemente con el objetivo de cumplir cada una de sus etapas.[1]

El objetivo principal del desarrollo de software es alcanzar la calidad, que hace referencia a conseguir características como velocidad, estabilidad, escalabilidad, flexibilidad, seguridad, entre otras. Es por ello que surge la necesidad de implementar las metodologías de desarrollo de software, las cuales a lo largo de los años han evolucionado de manera constante y presentan un conjunto de reglas, procesos y buenas prácticas como guías de desarrollo.[2]

Las metodologías de desarrollo de software incluyen fases y procedimientos que se deben seguir para el correcto desarrollo de un sistema o aplicación dependiendo del alcance o las necesidades dentro de un proyecto. Actualmente existen diferentes alternativas con respecto a métodos de trabajo dentro del desarrollo de software, cada una de las cuales tiene particularidades y características que permitan adaptarse a un proyecto específico.

Actualmente, existen diferentes tipos de metodologías para el desarrollo de software, cada una cuenta con ciertas características que se adaptan a diferentes tipos de proyectos dependiendo de las necesidades existentes, dichas metodologías se clasifican en metodologías tradicionales, metodologías ágiles y metodologías híbridas.

2.1.1. Metodologías Tradicionales

También conocidas como metodologías pesadas por su proceso de llevar una documentación extensa de todo lo que sucede en el proyecto como procesos de planificación, control, especificaciones, cumplimiento, etc. Se basan en la disciplina en el proceso de desarrollo de software con el objetivo de obtener un producto más eficiente, mantener controlados todos los procesos, definir roles, actividades, artefactos, herramientas para contar con una documentación detallada del proyecto.[3]

2.1.2. Metodologías Ágiles

Surgen debido a la necesidad de solventar los problemas que puedan existir en las metodologías tradicionales. Se basan principalmente en la planificación adaptativa y en la adaptabilidad de los procesos del proyecto. Las principales características de estas metodologías son la entrega de manera incremental de la solución, el trabajo colaborativo, la adaptabilidad dependiendo de las necesidades existentes, la correcta comunicación entre clientes y desarrolladores, entre otras.[3]

2.1.3. Metodologías Híbridas

Como bien sabemos las principales metodologías existentes son las metodologías tradicionales y ágiles, de ellas surgen las metodologías híbridas las cuales se basan en la combinación de metodologías y sus principales características y ventajas para el desarrollo de un proyecto de software mediante el uso de las buenas prácticas de cada una de ellas.[4]

A continuación, se presenta un cuadro comparativo entre las metodologías tradicionales y ágiles.

Tabla I Metodologías tradicionales vs ágiles. Fuente: Autor

TIPOS DE METODOLOGÍAS		
	METODOLOGÍAS TRADICIONALES	METODOLOGÍAS ÁGILES
Comunicación	Poca comunicación entre el cliente y el desarrollador	Constante comunicación entre el cliente y el desarrollador
Entregables	Una sola entrega al final del proyecto	Entregas incrementales de la solución
Documentación	Documentación extensa	Poca documentación
Enfoque	Está orientado a procesos	Está orientado a las necesidades del usuario
Adaptabilidad	Puede llegar a tener resistencia al cambio	Está preparado para cualquier cambio que se requiera ya sea durante o después del desarrollo del proyecto

En base a la Tabla I y en función a las características del proyecto a desarrollar, dentro del cual se pretende mantener una constante comunicación con el usuario final para determinar sus necesidades para el desarrollo de la aplicación y adaptar la aplicación y sus diferentes módulos de acuerdo a sus requerimientos mediante la realización de varias entregas en tiempos determinados, se decide usar las metodologías ágiles para el desarrollo de la solución, específicamente la metodología SCRUM siendo esta la más usada actualmente en el desarrollo de proyectos de software y la más popular entre las metodologías ágiles.

2.2. SCRUM

SCRUM es una metodología que divide el proyecto en pequeñas iteraciones o sprints que se planifican y se revisan de manera seguida. La productividad, innovación y flexibilidad son esenciales en esta metodología. Además, hace uso de las buenas prácticas y el trabajo colaborativo para alcanzar un mejor resultado final dentro de un entorno donde se necesitan obtener resultados lo antes posible, basándose en entregas parciales y regulares del producto final.

2.2.1. Proceso de SCRUM

El proceso de SCRUM se define en base al Product Backlog que actúa como el plan del proyecto y en el cual se encuentran la lista de objetivos definidos por el cliente, se ejecuta en procesos cortos y fijos de entre 2 a 4 semanas como máximo.

Se divide en cinco fases:

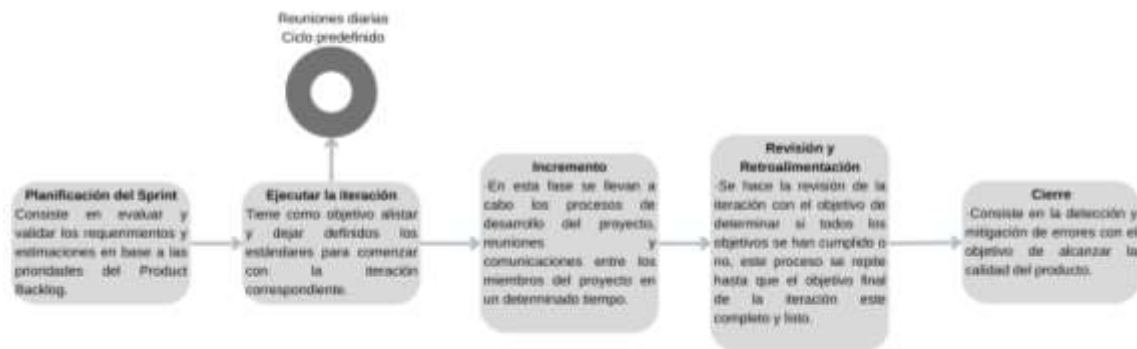


Fig. 1 Proceso de SCRUM.

Adaptado de: <https://revistas.uniminuto.edu/index.php/Inventum/article/view/9/9>

2.2.2. Actividades de SCRUM

Son todas las tareas que se deben realizar o que se realizarán a lo largo de todo el sprint, entre ellas tenemos:

- **Sprint Planning:** Es la planificación de las tareas a realizar para el desarrollo de software mediante reuniones con el cliente y el equipo de trabajo.[3]
- **Ejecución del Sprint:** Llamado también iteración la cual se encarga de facilitar un resultado completo o un incremento del producto.[3]
- **Daily Meeting:** Básicamente es una reunión entre el equipo de trabajo para transmitir información necesaria con respecto al proyecto lo que aumenta la productividad al poner en visibilidad los puntos más importantes.[3]

- **Sprint Review:** Es una reunión entre el equipo de trabajo y el cliente para presentar el incremento del proyecto con el objetivo de determinar si el cliente quiere implementar nuevas adaptaciones al proyecto.[3]
- **Sprint Retrospective:** Se hace una revisión general de todo lo realizado en un incremento con el propósito de determinar si se han conseguido o no los objetivos propuestos y mejorar la productividad o calidad del producto.[3]
- **Product Backlog Refinement:** Consiste en redefinir de mejor manera las actividades del proyecto de acuerdo a las necesidades actuales para que se adapten mejor al entorno.[3]

2.2.3. Herramientas de SCRUM

- **Product Owner:** Es la persona que mejor conoce el ámbito empresarial del cliente y que se encarga de definir los objetivos del producto de manera prioritaria.[3]
- **Scrum Master:** Es el encargado de liderar al equipo de desarrollo y mantener un estado de orden mediante el cumplimiento de ciertas responsabilidades.[3]
- **Equipo de Desarrollo:** El promedio de un equipo de desarrollo es de entre 5 a 9 personas los cuales son los encargados de desarrollar el producto del proyecto. Un equipo debe ser autoorganizado y cumplir las actividades propuestas en el Backlog.[3]

2.2.4. Roles y Responsabilidades de SCRUM

- **Product Owner:** Es la persona que mejor conoce el ámbito empresarial del cliente y que se encarga de definir los objetivos del producto de manera prioritaria.[3]

- **Scrum Master:** Es el encargado de liderar al equipo de desarrollo y mantener un estado de orden mediante el cumplimiento de ciertas responsabilidades.[3]
- **Equipo de Desarrollo:** El promedio de un equipo de desarrollo es de entre 5 a 9 personas los cuales son los encargados de desarrollar el producto del proyecto. Un equipo debe ser autoorganizado y cumplir las actividades propuestas en el Backlog.[3]

2.3. Facturación Electrónica

La facturación electrónica es un proceso de generación de un documento tributario digital en lugar de facturas físicas en papel por medio de una plataforma o sistema informático el cual contiene información referente a la compra y venta de productos o servicios dentro de una entidad específica, contiene los mismos datos de una factura tradicional, adicionalmente cuenta con una firma electrónica la cual proporciona cierta validez fiscal y legal al documento.[5]

Actualmente el uso de la facturación electrónica ha crecido debido a la modernización de las empresas y a que los entes reguladores exigen que se implemente este proceso en cada una de ellas para evitar la evasión tributaria.

Es importante conocer las ventajas que brinda la facturación electrónica frente a la facturación tradicional:

- **Gastos más bajos:** Ya no se gasta en papel, archivadores, impresoras, facturas preimpresas, etc.[5]
- **Mayor facilidad en la realización de trámites:** Ya que es un proceso que toma poco tiempo y no hace falta hacer trámites administrativos.[5]
- **Ahorro de espacio físico:** Los documentos emitidos pueden ser almacenados en la nube o en un dispositivo de almacenamiento.[5]

- **Proceso más seguro:** Ya que se usan firmas digitales y la validación de los entes reguladores para saber que es un documento auténtico.[5]

2.3.1. Facturación Electrónica en Ecuador

Este proceso surgió en el año 2009 primero como una resolución publicada en el Registro Oficial y que se validó en el año 2012 en el cual se definen las normas para la emisión de comprobantes, documentos, notas de crédito, entre otros. Inicialmente empezó como un proyecto piloto o de pruebas enfocado a un cierto grupo de contribuyentes los cuales contaban con ciertos recursos para ajustarse a los procesos tecnológicos correspondientes.[6]

El Servicio de Rentas Internas (SRI) estableció en el año 2013 que los contribuyentes debían unirse al sistema de facturación electrónica el cual ayudaría a disminuir en cierto grado el margen de error en el registro y velocidad con la que las Administraciones Tributarias atienden una transacción.[6]

Para empezar adecuadamente el proceso de facturación electrónica se requieren los siguientes requisitos:

- Firma electrónica.
- Aplicación o software para generar comprobantes electrónicos.
- Conexión a Internet.
- Clave de acceso a SRI en Línea.
- Convenio de débito registrado.

2.3.1.1. Proceso

Es necesario que los contribuyentes que decidan empezar a emitir comprobantes electrónicos conozcan la existencia de un ambiente de pruebas y otro de producción.

En el ambiente de pruebas el usuario podrá realizar los ajustes necesarios en el o los sistemas de emisión de comprobantes electrónicos haciendo uso de todos los recursos proporcionados por el SRI sin preocuparse por la validez tributaria de los comprobantes que se generen.[7]

El ambiente de producción está más enfocado a los contribuyentes que ya han realizado ajustes y modificaciones correspondientes en el ambiente de pruebas, por lo tanto, los comprobantes electrónicos emitidos ya tendrán la respectiva validez tributaria.[7]

Los pasos que se deben seguir para solicitar al SRI la autorización para la emisión de comprobantes electrónicos son los siguientes:

- **Registrar una firma electrónica:** Puede ser adquirido en una entidad de certificación autorizada como el Banco Central del Ecuador o en entidades privadas encargadas de proporcionar una firma electrónica con la autorización proporcionada por el ARCOTEL.[7]
- **Realizar la solicitud autorización de emisión de comprobantes electrónicos:** Mediante la página web del SRI en el apartado SRI en línea y en la parte de facturación electrónica, se realiza la solicitud para la emisión de comprobantes con los datos del contribuyente (RUC y contraseña de la plataforma del SRI) y seleccionando el ambiente en el que se desea trabajar (pruebas o producción).[7]
- **Configurar la generación de comprobantes electrónicos:** Dichos comprobantes se generarán en formato .xml los cuales contienen una clave de acceso especial de 49 dígitos que están comprendidos de la siguiente manera:

Tabla II Dígitos para la clave de acceso.

Adaptado de: <https://www.sri.gob.ec/facturacion-electronica>

DÍGITOS QUE CONFORMAN LA CLAVE DE ACCESO		
CAMPO	FORMATO	LONGITUD
Fecha de emisión	ddmmaaaa	8
Tipo de comprobante	Tabla IV	2
RUC	Númerico	13
Ambiente	Pruebas (1) – Producción (2)	1
Serie	001001	6
Número de comprobante (Secuencial)	000000001	9
Código numérico	Numérico	8
Tipo de emisión	Tabla III	1
Dígito verificador (módulo 11)	Numérico	1

Tabla III Tipo de emisión.

Adaptado de: <https://www.sri.gob.ec/facturacion-electronica>

TIPO DE EMISIÓN	
EMISIÓN	CÓDIGO
Emisión normal	1

Tabla IV Tipos de comprobante.

Adaptado de: <https://www.sri.gob.ec/facturacion-electronica>

TIPOS DE COMPROBANTE	
COMPROBANTE	CÓDIGO
Factura	01
Liquidación de compra de bienes y prestación de servicios	03
Nota de crédito	04
Nota de débito	05
Guía de remisión	06
Comprobante de retención	07

En el caso del dígito verificador actúa como mecanismo de detección de errores, se debe generar haciendo uso de un algoritmo que se basa en recorrer cada dígito de derecha a izquierda de la clave de acceso que inicialmente está comprendido por 48 dígitos y multiplicar cada dígito por 2, 3, 4, 5, 6 y 7 de manera sucesiva,

luego de dicha operación se deben sumar todos los valores obtenidos para luego sacar el resto de su división para 11.

- **Contar con un software para la emisión de comprobantes electrónicos:** Este proceso se puede realizar ya sea mediante la misma plataforma del SRI la cual brinda un facturador electrónico gratuito o mediante un software propio el cual permitiría al usuario tener un proceso más personalizado en lo que respecta a facturación electrónica y emisión de comprobantes electrónicos.[7]

2.4. Cloud Computing

La computación en la nube se puede definir como una concepción tecnológica cuyo objetivo es la prestación de servicios y recursos informáticos que se encuentran dentro de la red. Por medio de la computación en la nube se puede tener acceso a diferentes recursos que se pueden configurar, por ejemplo, el almacenamiento en la nube, los servidores, redes, entre otros. El principal beneficio que ofrece cloud computing es hacer uso y operar los servicios según sea sus necesidades y conveniencia del usuario.

2.4.1. Modelos de Cloud Computing

Los modelos de computación en la nube son conocidos también como modelos de implementación y se establecen de acuerdo al enfoque en el que vaya a ser usado, dichos modelos se dividen en:

- **Nube privada:** Puede existir dentro o fuera de las instalaciones de una organización. Hace referencia al uso especial por parte de una sola entidad la cual cuenta con varios clientes o unidades de negocio. Puede ser de propiedad, manejado y puesto en marcha por la organización, por un tercero, o una combinación de ellos. [8]

- **Nube comunitaria:** Su enfoque va para el uso exclusivo de una comunidad específica. Puede ser de propiedad, manejado y puesto en marcha por una o más organizaciones que forman parte de la comunidad, un tercero o una combinación de ellos.[8]
- **Nube pública:** Se encuentra solamente en el lugar del proveedor del servicio. Está enfocada para el uso del público en general. Puede ser de propiedad, manejado y puesto en marcha por una empresa, academia, instituciones del gobierno, o un tipo de combinación de ellos.[8]
- **Nube híbrida:** Es una combinación de dos o más tipos de las nubes mencionadas anteriormente las cuales siguen siendo entidades únicas, pero están enlazadas por tecnología estandarizada que consiente la portabilidad de datos y aplicaciones.[8]

2.4.2. Servicios de Cloud Computing

Los servicios que ofrece la computación en la nube se basan en los requerimientos del usuario, es por ellos que tenemos:

- **Infraestructura como servicio:** Da a conocer al usuario recursos de infraestructura como almacenamiento, virtualización, redes, etc. La limitación de infraestructura en la nube que un proveedor pueda ofrecer depende de las necesidades y la capacidad económica de contratación del servicio que el cliente posea.[8]
- **Plataforma como servicio:** Brinda al usuario las herramientas necesarias para implementar distintos proyectos informáticos, sin tener la necesidad de adquirir o implementar equipos locales. Las herramientas que el usuario adquiera deben

ser supervisadas y mantenidas por los propios proveedores de servicios en la nube.[8]

- **Software como servicio:** Hace referencia a productos que se encuentran en su fase de producción, es decir aplicaciones o productos de software que brindan cierto tipo de servicios, de igual manera consiste en la utilización de los servicios ofrecidos por dichas aplicaciones por parte del usuario.[8]

2.5. Almacenamiento en la nube

Es una herramienta que forma parte de los servicios que ofrece cloud computing o computación en la nube. El almacenamiento en la nube elimina la necesidad de adquirir equipos físicos, configurarlos y montarlos ya que se ofrece bajo demanda lo que genera a la persona que adquiera este servicio cuenta con buena escalabilidad, agilidad y durabilidad con respecto al acceso a datos. Es un modelo que guarda información o datos a través de internet por medio de un proveedor de informática en la nube el cual se encarga de gestionar y manejar el almacenamiento en la nube como un servicio.

Algunos de sus beneficios son:

- **Costo total de la propiedad:** Permite añadir o quitar recursos bajo demanda y contar y pagar por la capacidad de almacenamiento necesaria, elimina la necesidad de adquirir herramientas de hardware físico.[9]
- **Tiempo de implementación:** Proporciona rapidez al momento de montar una infraestructura de almacenamiento con la cantidad y recursos necesarios.[9]
- **Gestión de la información:** Consiste en la centralización del almacenamiento en la nube con el objetivo de realizar tareas de administración de información mediante las políticas del ciclo de vida del almacenamiento en la nube.[9]

2.6. Aplicaciones Web

Todo lo que conocemos actualmente como internet y la red nace aproximadamente en la década de los setenta por medio de la Agencia de Proyectos Avanzados para la defensa de los Estados Unidos. Por otro lado, la web nace en el año 1989 por medio de Tim Berners-Lee el cual diseño un sistema con acceso rápido para el Centro Europeo de Investigación Nuclear.[10]

Una aplicación web es una herramienta por la cual un usuario accede desde un navegador a los recursos existentes en un servidor por medio de una red de internet. Las aplicaciones web nacen de la necesidad de contar con páginas dinámicas para los usuarios ya que hace varios años atrás la web era un conjunto de documentos o paginas estáticas las cuales solamente se podían descargar y consultar una vez.[11]

El éxito actual que tiene la web es gracias a dos factores importantes que en su inicio fueron fundamentales:

- **Protocolo HTTP:** Hypertext Transfer Protocol que sirve para realizar la comunicación entre clientes y servidores por medio de peticiones.[10]
- **Lenguaje HTML:** Hypertext Mark-Up Language es un lenguaje de marcado que permite mostrar de mejor manera el contenido y hacer referencia a otro tipo de recursos.[10]

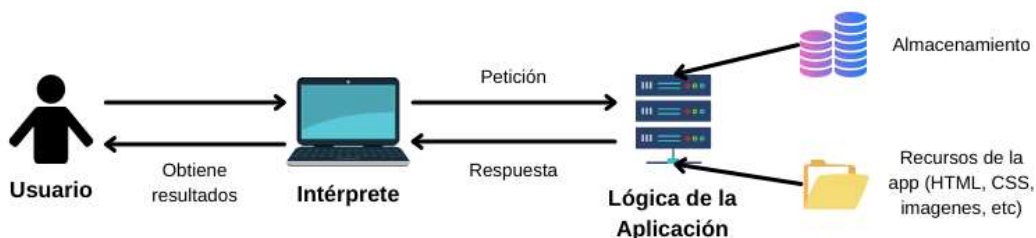


Fig. 2 Arquitectura de una Aplicación Web.

Fuente: Autor

2.7. Herramientas de Desarrollo Web

Son todas aquellas que permiten el correcto diseño, desarrollo, funcionamiento e implementación de aplicaciones web que van desde sitios simples y estáticos hasta aplicaciones dinámicas que hacen uso de las tecnologías web más recientes. Actualmente, existen varias herramientas para el desarrollo web y cada una con ciertas características que pueden beneficiar en casos concretos dependiendo de las necesidades o requerimientos que existan, es por ello que a continuación se presenta un esquema con las herramientas de desarrollo web más populares y más usadas.

Tabla V Herramientas para el desarrollo web.

Fuente: Autor

HERRAMIENTAS PARA DESARROLLO WEB		
DESCRIPCIÓN	HERRAMIENTAS	CARACTERÍSTICAS
Editores de código	Visual Studio Code	<i>Open source</i> , multiplataforma, eficiente, terminal integrado, compatible con varios lenguajes de programación.
	Atom	<i>Open source</i> , multiplataforma, compatible con varios lenguajes de programación.
Diseño (Frontend)	Angular	<i>Open source</i> , código reutilizable, modelo MVC, enlace de datos bidireccional.
	React	<i>Open source</i> , código reutilizable solo CSS, modelo DOM virtual, enlace de datos unidireccional.
	Vue.js	<i>Open source</i> , código reutilizable solo CSS y HTML, modelo DOM virtual, enlace de datos bidireccional.
Funcionamiento y almacenamiento (Backend)	Node.js	Código reutilizable, multiplataforma, escalable, asíncrono, uso del formato <i>JSON</i> y módulos, mejor integración con <i>Angular</i> .
	.NET	Multiplataforma, escalable, asíncrono, uso del formato <i>JSON</i> y módulos.
	AWS	Escalable, importación y exportación de datos, disponibilidad, alto rendimiento, se debe contratar y pagar el servicio de acuerdo a los requerimientos del usuario.
	Firebase	Proporciona un servicio de autenticación de usuarios, analítica de datos, escalable, disponibilidad, importación y exportación de datos, cuenta con dos tipos de licencia (libre y de pago)

2.7.1. Editor de código

Existe una gran variedad de editores de código hoy en día, cada uno con ciertas características y funciones que permiten al usuario trabajar y adaptarse de mejor manera a un

entorno. Un editor de código nos ayuda de cierta forma a aumentar la productividad y facilitar el trabajo ya que cuentan con diferentes extensiones para diferentes tipos de proyectos y la función de autocompletado o la identificación de posibles errores de tipeo o de referencias.

Entre los editores de código más usados actualmente se encuentran Visual Studio Code, Atom y Sublime Text, por lo que a continuación se detalla un cuadro comparativo con las características más importantes de un editor de código para el desarrollo de software.

2.7.1.1. Visual Studio Code

Es un editor de código fuente que ha sido desarrollado por Microsoft y está disponible tanto para Windows como para macOS y Linux, se identifica por ser un editor de código ligero, pero con grandes características lo cual hace que sea una herramienta potente, cuenta con soporte integrado para JavaScript, TypeScript y Node.js, de igual manera permite al usuario trabajar en lenguajes como C#, C, Java, Python, PHP, etc.[12]

2.7.2. Frontend

Frontend que hace referencia a la interfaz de usuario es un apartado del desarrollo de proyectos web o móviles que se encarga del diseño de toda la parte visual de un aplicativo, su estructura, colores, imágenes, fuentes de texto, animaciones, etc.

Para realizar el proceso de Frontend se usan el lenguaje de marcado HTML para formar la estructura de la interfaz, CSS para modificar el diseño de la estructura creada en base a HTML y JavaScript para el dinamismo de la aplicación. De igual manera se pueden usar los distintos frameworks como Angular, React, Vue.js que permite desarrollar aplicaciones de manera dinámica en base a la reutilización de código, uso de bibliotecas según los requerimientos, mejor organización en la estructura de la aplicación, etc.

2.7.2.1. Angular

Angular es una plataforma para el desarrollo de software enfocada al diseño de aplicaciones web y móviles, sienta sus bases en el lenguaje Typescript. Hace uso de componentes para crear aplicaciones escalables, bibliotecas integradas con una amplia variedad de funciones como el enrutamiento, manejo de formularios y la comunicación cliente – servidor.[13]

2.7.2.1.1. Arquitectura

Angular está basado en el uso de módulos y componentes para la comunicación entre las distintas partes de una aplicación, de igual manera hace uso del enrutamiento para acceder a los diferentes componentes existentes en una app.[14]

- **Módulos:** Es el código que se encarga únicamente a un solo objetivo. También puede hacer referencia a ciertas librerías usadas.[14]
- **Componentes:** Es la parte que controla la vista en conjunto con la lógica.
- **Template:** Es lo que permite modificar la estructura de la vista la cual se encuentra en formato HTML.[14]
- **Data Binding:** Se encarga de obtener las respuestas de los usuarios en acciones concretas, es decir pasar datos entre componentes.[14]
- **Directivas:** Es una clase que permite agregar, manipular, o borrar los elementos del DOM del HTML.
- **Servicios:** Es un componente en el cual está relacionado la lógica y provee funcionalidad adicional a los demás componentes.[14]
- **Dependency Injection:** Es un método que permite agregar nuevas dependencias a una clase.[14]

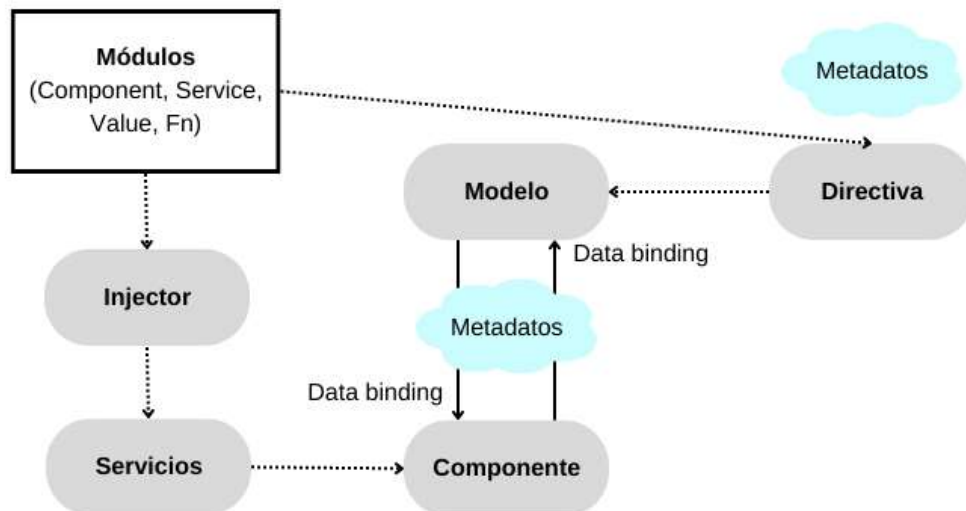


Fig. 3 Arquitectura de Angular.

Adaptado de: <https://angular.io/generated/images/guide/architecture/overview2.png>

2.7.3. Backend

Podemos decir que la parte de backend es todo lo relacionado con la lógica de programación usada para el desarrollo de un proyecto de software, es un proceso que el usuario como tal no puede apreciar de manera visual ya que es un proceso interno a nivel de servidor. En este apartado es esencial la correcta funcionalidad de una aplicación, optimización de los recursos y seguridad.

La Figura 4 hace referencia a una encuesta realizada por StackOverflow sobre los Frameworks y tecnologías web más usadas en el año 2022 podemos apreciar que Node.js se posiciona en el primer lugar como la más usada lo que permite reconocer la importancia de trabajar con Node.js, el cual es seleccionado para el presente trabajo.



Fig. 4 Encuesta de StackOverflow sobre los Frameworks y Tecnologías Web más usadas en el año 2022.

Fuente: <https://survey.stackoverflow.co/2022/#technology>

2.7.3.1. Node.js

Node.js es conocido como un entorno de ejecución de JavaScript que está dirigido a eventos asíncronos el cual tiene como objetivo la creación y diseño de aplicaciones escalables en la web. Este entorno es compatible con Windows, Linux y macOS y sus bases están construidas en base al motor V8 que inicialmente se pensaba su uso en Google Chrome.[15]

Algunas ventajas de Node.js:

- Permite generar páginas con contenidos dinámicos.[16]

- Puede agregar, abrir, visualizar, editar, borrar y cerrar archivos en el servidor.[16]
- Actualmente es la mejor opción al momento de usar API REST.[16]
- Baja manutención de código.[16]
- Hace uso de NPM (Node Package Manager) para realizar empaquetamiento.[16]
- Mejor integración con bases de datos No SQL.[16]
- Implementación fácil.[16]

Se debe tener en cuenta la parte de almacenamiento ya que toda aplicación web genera información la cual debe ser almacenada de alguna manera, en este caso se pretende trabajar con bases de datos No SQL.

La Figura 5 se basa en la encuesta proporcionada por StackOverflow realizada cada año se aprecia que Firebase se encuentra en el Top 4 de la lista de plataformas de almacenamiento en la nube, esta información permite determinar el servicio de almacenamiento para el sistema web.

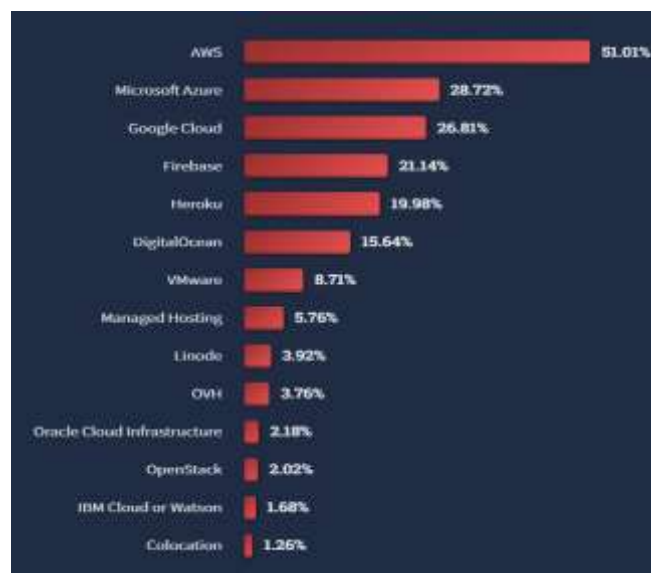


Fig. 5 Encuesta de StackOverflow sobre plataformas de almacenamiento en la nube.

Fuente: <https://survey.stackoverflow.co/2022/#technology>

Debido a que Firebase brinda las mismas funcionalidades que otras herramientas del mercado de forma gratuita, se ha decidido usarlo en el presente trabajo.

2.7.3.2. Firebase

Firestore es una base de datos No SQL en tiempo real que a diferencia de las demás hace uso de Websockets para obtener datos, esto hace que sea más rápido ya que no necesita hacer conexiones simultaneas, esta base de datos se enfoca en el manejo de información a través de una aplicación y al crecimiento que esta pueda tener.[17]

Características:

- **Desarrollo:** Son las herramientas usadas para el desarrollo de proyectos de aplicaciones web o móviles.[17]
- **Analítica:** Son todas las herramientas relacionadas a la analítica de datos que están asociadas a las estrategias de marketing como la toma de decisiones.[17]
- **Lanzamiento y supervisión:** Son herramientas enfocadas a mejorar la calidad de una o varias aplicaciones, la rápida corrección de errores y la simplificación de pruebas de la app.[18]

Además, Firestore proporciona una licencia gratuita denominada Spark para hacer uso de sus recursos.

2.7.3.2.1. Firestore Cloud Database

Es una base de datos usada en el desarrollo de aplicaciones web y móviles usando también Google Cloud, maneja SDK nativos de Node.js, Java, Python, C++, entre otras. Proporciona flexibilidad y escalabilidad, conserva los datos sincronizados entre aplicaciones mediante objetos de escucha en tiempo real y brinda soporte sin conexión para apps móviles y

web, esto último permite compilar aplicaciones con una buena calidad de respuesta sin importar la calidad de internet.[19]

Características:

- **Flexibilidad:** Almacenamiento por medio de documentos de manera jerárquica que van organizados en colecciones o sub colecciones.[19]
- **Consultas expresivas:** Uso de consultas para obtener uno o varios documentos de una colección por medio de ciertos parámetros de consulta o filtros de cadena, hace uso también de índices haciendo que el rendimiento de las consultas sea proporcional a la cantidad de resultados.[19]
- **Actualizaciones en tiempo real:** Sincroniza la información almacenada en cualquier dispositivo o cliente conectado a la base de datos.[19]
- **Asistencia sin conexión:** Permite la lectura y escritura de datos, aunque el dispositivo no tenga conexión, esto se logra ya que se almacena en cache los datos que la aplicación usa.[19]
- **Escalable:** Integrada a Google Cloud proporciona la adaptabilidad de los datos de acuerdo al tamaño y complejidad de una aplicación.[19]

2.7.3.2.2. Firebase Authentication

Es un servicio de Firebase que sirve para manejar la autenticación de usuarios dentro de una aplicación, brinda servicios de backend, SDK's y bibliotecas de IU predeterminadas.[20]

Los métodos de autenticación varían de acuerdo a las necesidades del usuario, algunas de ellas son:

- **Correo electrónico y contraseña:** Mediante correo electrónico que puede ser o no verificado por el mismo proveedor y contraseña, Además Firebase proporciona su propio servicio para restablecer contraseñas.[20]

- **Proveedores de identidad federada:** Mediante cuentas de plataformas como Google, Facebook, Twitter, GitHub, Apple.[20]
- **Número telefónico:** Se hace mediante el envío de mensajes de texto a un teléfono para realizar la autenticación.[20]

2.7.3.2.3. Firebase Hosting

Es un servicio de alojamiento en la web para aplicaciones web en producción proporcionado por Firebase que permite al usuario alojamiento rápido y seguro para aplicaciones web, contenido estático y dinámico y microservicios. Es un servicio fácil de implementar y posee excelente adaptabilidad con Angular.[21]

Sus características son: Alojamiento de contenido estático y dinámico a través de conexión segura, entrega el contenido de manera óptima, permite al usuario implementar o actualizar nuevas versiones de manera fácil con un solo comando.[21]

CAPÍTULO III

MARCO METODOLÓGICO

3.1. Desarrollo

Este capítulo describe el desarrollo de la aplicación web, el cual tiene como objetivo principal solventar los inconvenientes existentes en el proceso de facturación de la empresa Servcomp de la ciudad de Cañar y El Tambo, además brindar un sistema agradable e intuitivo para mejorar la experiencia del usuario final (Gerente y Empleados de la empresa).

En este apartado se describen las fases y procesos necesarios para el cumplimiento del proyecto de desarrollo de software en base a la metodología propuesta en el capítulo anterior basados en la metodología Scrum.

3.2. Planificación del Sprint

Los requerimientos iniciales se determinaron mediante una reunión con el Ing. Silvio Pacheco, gerente de la empresa Servcomp que se dedica a la venta, actualización y reparación de equipos informáticos en los cantones de Cañar y el Tambo de la Provincia de Cañar. Esta reunión se llevó a cabo con el objetivo de recolectar información sobre los requerimientos funcionales y no funcionales desde la perspectiva del usuario para generar el Product Backlog (ver Tabla VI)

También se planteó conjuntamente con el gerente y personal de la empresa en realizar reuniones semanales con el objetivo de verificar el avance del proyecto en cada sprint y el grado de aceptación de los mismos con respecto al aplicativo.

Con respecto a la metodología Scrum y previo al desarrollo del proyecto es importante definir el Product Backlog en el cual se especifican los requerimientos del cliente mediante historias de usuario la cuales fueron elaboradas conjuntamente con los usuarios implicados y priorizadas de acuerdo a su importancia de ejecución.

Cada historia de usuario sigue el formato:

- Id
- Nombre
- Importancia (intervalos de 1 a 5, siendo 5 muy importante y 1 poco importante)
- Estimación (en semanas, de 1 a 4 semanas como máximo)
- Descripción

Ejemplo de una historia de usuario:

1. Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal.

Como Gerente

Quiero que la información de la empresa se guarde en un solo lugar

Para poder acceder a ella desde cualquier sitio.

En la Tabla VI se presentan los requerimientos del proyecto.

Tabla VI Product Backlog.

Fuente: Autor

PRODUCT BACKLOG				
ID	HISTORIA DE USUARIO	IMPORTANCIA	ESTIMACIÓN (SEMANAS)	DESCRIPCIÓN
1	Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal.	5	1	Crear la base de datos en la nube para almacenar toda la información que se genere en el sistema.
2	Gestión de Proveedores	5	1	Crear un módulo para gestionar (Insertar, Leer, Editar, Eliminar) los Proveedores.
3	Gestión de Clientes	5	1	Crear un módulo para gestionar (Insertar, Leer, Editar, Eliminar) los Clientes.

4	Gestión de Inventario	5	1	Crear un módulo para gestionar (Insertar, Leer, Editar, Eliminar) los Productos.
5	Gestión de Kárdex	5	2	Crear un módulo para gestionar el inventario.
6	Facturación electrónica	5	3	Crear un módulo que permita realizar facturación y emisión de comprobantes electrónicos
7	Crear el Dashboard	4	2	Crear un módulo que permita visualizar información importante para el usuario.
8	Crear el Login	4	1	Crear un login para el acceso de los usuarios a la aplicación.
9	Autenticación y roles de usuario	4	1	Asignar roles a los usuarios para el acceso a los recursos e información del sistema.
10	Publicar la aplicación web	4	1	Publicar la aplicación web.

3.2.1. Definición de los Sprint

Luego de haber establecido el Product Backlog se determinan los sprint y sus estimaciones de tiempo tomando en cuenta que a lo largo del proyecto pueden existir contratiempos que de alguna manera retrasen el desarrollo de cada sprint.

Cada historia de usuario cuenta con un rango de estimación que está comprendido entre 1 a 4 semanas de esfuerzo y un rango de importancia comprendido entre intervalos de 1 a 5.

La Tabla VII contiene la lista de requerimientos de usuario del primer sprint.

Tabla VII Sprint 1.

Fuente: Autor

SPRINT 1				
ID	HISTORIA DE USUARIO	IMPORTANCIA	ESTIMACIÓN	DESCRIPCION
1	Almacenar y centralizar la información referente al	5	1	Crear la base de datos en la nube para almacenar toda la

	proceso de facturación de la matriz y sucursal.			información que se genere en el sistema.
2	Gestión de Proveedores	5	1	Crear un módulo para gestionar (Insertar, Leer, Editar, Eliminar) los Proveedores.
3	Gestión de Clientes	5	1	Crear un módulo para gestionar (Insertar, Leer, Editar, Eliminar) los Clientes.

La Tabla VIII contiene la lista de requerimientos de usuario del segundo sprint.

Tabla VIII Sprint 2.

Fuente: Autor

SPRINT 2				
ID	HISTORIA DE USUARIO	IMPORTANCIA	ESTIMACIÓN	DESCRIPCION
4	Gestión de Inventario	5	1	Crear un módulo para gestionar (Insertar, Leer, Editar, Eliminar) los Productos.
5	Gestión de Kárdex	5	2	Crear un módulo para gestionar el inventario.

La Tabla IX contiene la lista de requerimientos de usuario del tercer sprint.

Tabla IX Sprint 3.

Fuente: Autor

SPRINT 3				
ID	HISTORIA DE USUARIO	IMPORTANCIA	ESTIMACIÓN	DESCRIPCION
6	Facturación electrónica	5	3	Crear un módulo que permita realizar facturación y emisión de comprobantes electrónicos
7	Crear el Dashboard	4	2	Crear un módulo que permita visualizar información importante para el usuario.

La Tabla X contiene la lista de requerimientos de usuario del cuarto sprint.

Tabla X Sprint 4.

Fuente: Autor

SPRINT 4				
ID	HISTORIA DE USUARIO	IMPORTANCIA	ESTIMACIÓN	DESCRIPCION
8	Crear el Login	4	1	Crear un login para el acceso de los usuarios a la aplicación.
9	Autenticación y roles de usuario	4	1	Asignar roles a los usuarios para el acceso a los recursos e información del sistema.
10	Publicar la aplicación web	4	1	Publicar la aplicación web.

Para llevar a cabo cada sprint se mantuvieron revisiones periódicas con el gerente de la empresa para verificar los avances del proyecto y así contar con una especie de feedback y mejorar la calidad de los entregables para el final del proyecto.

La Tabla XI el detalle de la planificación del primer sprint.

Tabla XI Planificación Sprint 1.

Fuente: Autor

PLANIFICACIÓN DEL SPRINT 1	
Delegado / Autor: Silvio David Pacheco Herrera	
Fecha de Inicio	15/Agosto/2022
Fecha de Fin	05/Septiembre/2022
Duración	3 semanas
Tareas a Desarrollar	Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal Gestión de Proveedores Gestión de Clientes

La Tabla XII el detalle de la planificación del primer sprint.

Tabla XII Planificación Sprint 2.

Fuente: Autor

PLANIFICACIÓN DEL SPRINT 2	
Delegado / Autor: Silvio David Pacheco Herrera	
Fecha de Inicio	05/Septiembre/2022

Fecha de Fin	26/Septiembre/2022
Duración	3 semanas
Tareas a Desarrollar	Gestión de Inventario Gestión de Kárdex

La Tabla XIII el detalle de la planificación del primer sprint.

Tabla XIII Planificación Sprint 3.

Fuente: Autor

PLANIFICACIÓN DEL SPRINT 3	
Delegado / Autor: Silvio David Pacheco Herrera	
Fecha de Inicio	26/Septiembre/2022
Fecha de Fin	31/Octubre/2022
Duración	5 semanas
Tareas a Desarrollar	Facturación electrónica Crear el Dashboard

La Tabla XIV el detalle de la planificación del primer sprint.

Tabla XIV Planificación Sprint 4.

Fuente: Autor

PLANIFICACIÓN DEL SPRINT 4	
Delegado / Autor: Silvio David Pacheco Herrera	
Fecha de Inicio	31/Octubre/2022
Fecha de Fin	21/Noviembre/2022
Duración	3 semanas
Tareas a Desarrollar	Crear el Login Autenticación y roles de usuario Publicar la aplicación web

3.3. Ejecución del Sprint

La ejecución de cada sprint fue monitoreada mediante diagramas Burndown Chart para determinar su avance. Cada Burndown Chart contiene una lista de las tareas pendientes de desarrollar y de aquellas que se implementaron tal como se detalla a continuación.

3.3.1. Lista de Tareas y Burndown Chart

La lista de tareas iniciales y el estado inicial de cada uno de los sprints se presenta en la Tabla XV.

Tabla XV Lista de tareas inicial.

Fuente: Autor

LISTA DE TAREAS INICIAL				
Inicio: 15/Agosto/2022		Autor: Silvio David Pacheco Herrera		
Fin: 21/Noviembre/2022				
SPRINT	HISTORIA DE USUARIO	PENDIENTE	EN DESARROLLO	TERMINADO
1	Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal.	X		
	Gestión de Proveedores	X		
	Gestión de Clientes	X		
2	Gestión de Inventario	X		
	Gestión de Kárdex	X		
3	Facturación electrónica	X		
	Crear el Dashboard	X		
4	Crear el Login	X		
	Autenticación y roles de usuario	X		
	Publicar la aplicación web	X		

En la Figura 6 se presenta el Burndown Chart inicial.

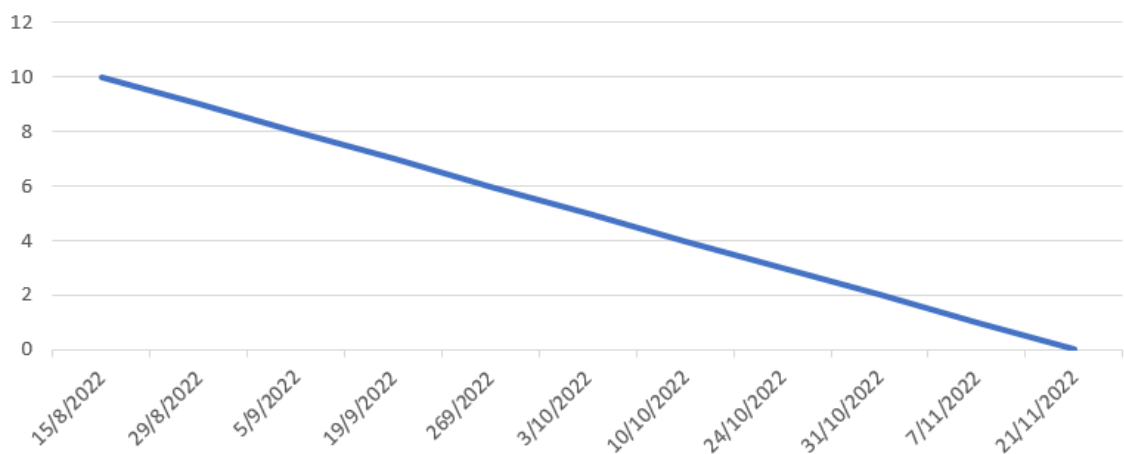


Fig. 6 Burndown Chart inicial.

3.4. Incremento

Inicialmente se pretendía desarrollar un aplicativo de escritorio para la solución del proyecto, pero de acuerdo a los requerimientos del usuario el aplicativo no iba a tener una buena adaptación a los servicios en la nube. Por lo tanto, se acordó desarrollar una aplicación web la cual tiene como objetivo principal cumplir con las necesidades del usuario y ser intuitiva y agradable en su diseño e interacción.

3.4.1. Sprint 1

En el Sprint 1 las tareas a desarrollar son las siguientes:

- Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal
- Gestión de Proveedores
- Gestión de Clientes

Se detalla a continuación la Tabla XVI donde se aprecia la lista de tareas correspondiente al Sprint 1 y la tarea “Desplegar la base de datos” se encuentra en desarrollo.

Tabla XVI Lista de tareas Sprint 1.

Fuente: Autor

LISTA DE TAREAS DEL SPRINT 1				
Inicio: 15/Agosto/2022		Autor: Silvio David Pacheco Herrera		
Fin: 05/Septiembre/2022				
SPRINT	HISTORIA DE USUARIO	PENDIENTE	EN DESARROLLO	TERMINADO
1	Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal.		X	
	Gestión de Proveedores	X		
	Gestión de Clientes	X		
2	Gestión de Inventario	X		
	Gestión de Kárdex	X		
3	Facturación electrónica	X		

	Crear el Dashboard	X		
4	Crear el Login	X		
	Autenticación y roles de usuario	X		
	Publicar la aplicación web	X		

En la Figura 7 se presenta el avance que se tiene en el Sprint 1 mediante el Burndown Chart.

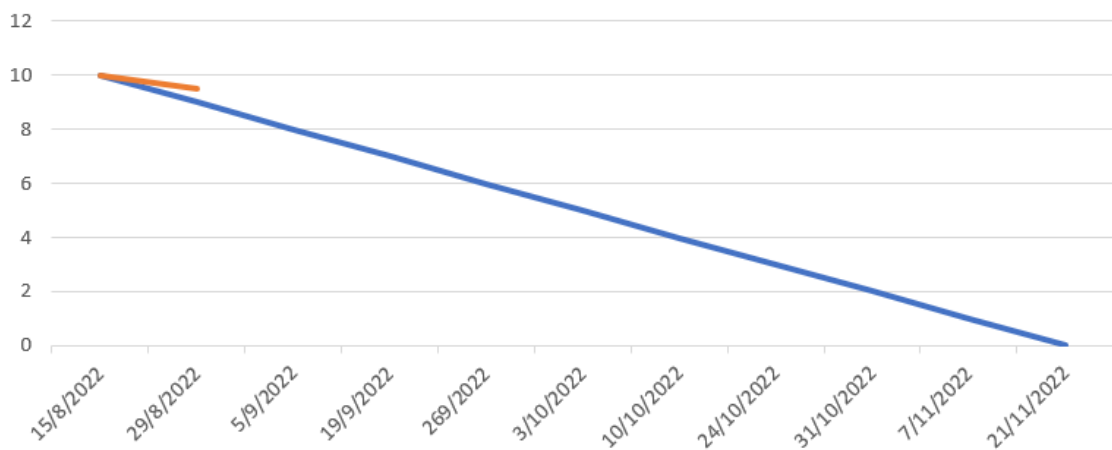


Fig. 7 Burndown Chart del Sprint 1.

Fuente: Autor

3.4.1.1. Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal

Inicialmente se planteó un modelo entidad relación para el almacenamiento de datos haciendo uso de una base de datos de PostgreSQL.

La Figura 8 presenta el modelo de datos inicial.

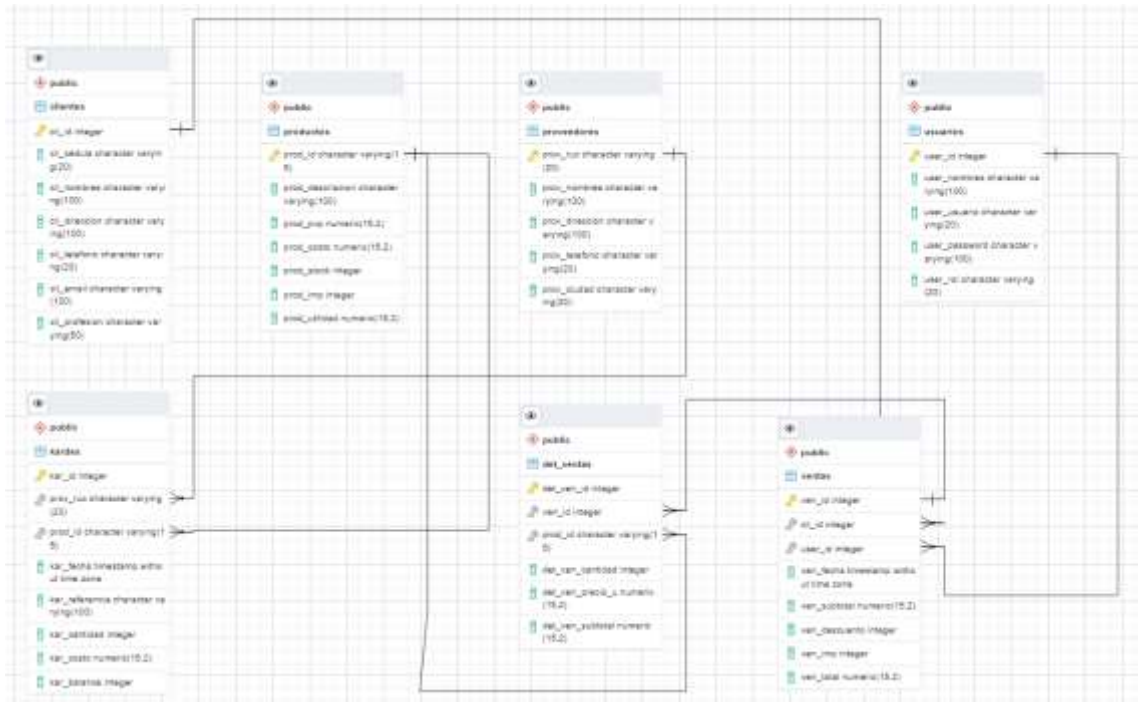


Fig. 8 Modelo de datos inicial.

Fuente: Autor

Debido a que el gerente de la empresa pretende unificar de alguna manera la información de la sucursal y la matriz para que la información que se genere en el proceso de facturación en ambos puntos se encuentre en una sola fuente de datos se requería del despliegue de un servidor físico que se encargue del almacenamiento de información.

Por este motivo se llegó a un acuerdo con el gerente para hacer uso de almacenamiento en la nube por medio de Firebase Firestore, dicho servicio proporciona el uso de manera gratuita de los recursos de almacenamiento lo cual es beneficioso para el desarrollo del proyecto ya que previo al despliegue del aplicativo web se necesitan realizar pruebas y peticiones a la base de datos para verificar que la información se almacene correctamente y su funcionalidad sea adecuada. Igualmente, el costo por almacenamiento que ofrece Firebase Firestore se adapta a las posibilidades económicas del gerente y la empresa.

En la Figura 9 se muestran las colecciones creadas en la base de datos de Firebase Firestore.

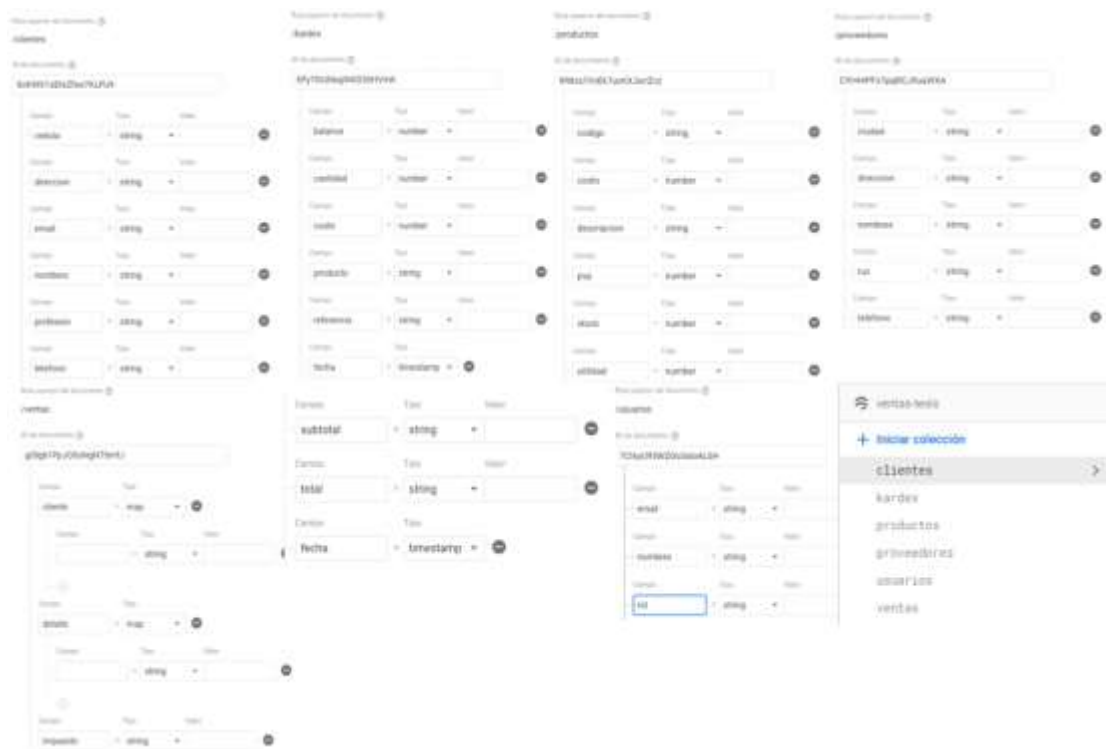


Fig. 9 Colecciones en Firestore Database.

Fuente: Autor

Se detalla a continuación la Tabla XVII donde se aprecia la lista de tareas correspondiente al Sprint 1 y la tarea “Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal.” se encuentra terminada.

Tabla XVII Avance de la lista de tareas Sprint 1.

Fuente: Autor

LISTA DE TAREAS DEL SPRINT 1				
Inicio: 15/Agosto/2022		Autor: Silvio David Pacheco Herrera		
Fin: 05/Septiembre/2022				
SPRINT	HISTORIA DE USUARIO	PENDIENTE	EN DESARROLLO	TERMINADO
1	Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal.			X
	Gestión de Proveedores		X	
	Gestión de Clientes	X		
2	Gestión de Inventario	X		
	Gestión de Kárdex	X		
3	Facturación electrónica	X		
	Crear el Dashboard	X		

4	Crear el Login	X		
	Autenticación y roles de usuario	X		
	Publicar la aplicación web	X		

En la Figura 10 se presenta el avance que se tiene en el Sprint 1 mediante el Burndown Chart, en el que se aprecia que los tiempos esperados para culminar el desarrollo del proyecto se mantienen.

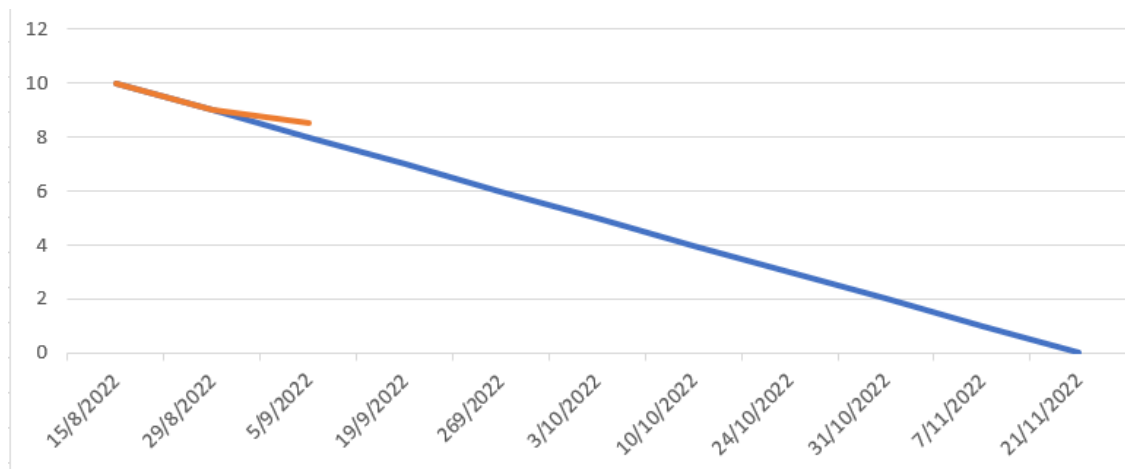


Fig. 10 Avance del Burndown Chart del Sprint 1.

Fuente: Autor

3.4.1.2. Gestión de Proveedores

En la Figura 11 se muestra la pantalla para la gestión de proveedores.



Fig. 11 Interfaz "Proveedores".

Fuente: Autor

En la Figura 12 se aprecian los campos necesarios para el registro de proveedores.

The image shows a mobile application interface for adding a new provider. The form is titled 'Detalle del Proveedor' and includes the following fields: Ruc, Nombres, Dirección, Teléfono, and Ciudad. At the bottom right, there is a blue 'Guardar' button with a checkmark icon. A red box highlights the text 'Botón para guardar un Proveedor' with an arrow pointing to the 'Guardar' button.

Fig. 12 Interfaz para registrar un nuevo Proveedor.

Fuente: Autor

Se detalla a continuación la Tabla XVIII donde se aprecia la lista de tareas correspondiente al Sprint 1 y la tarea “Gestión de Proveedores” se encuentra terminada.

Tabla XVIII Avance de la lista de tareas Sprint 1.

Fuente: Autor

LISTA DE TAREAS DEL SPRINT 1				
Inicio: 15/Agosto/2022		Autor: Silvio David Pacheco Herrera		
Fin: 05/Septiembre/2022				
SPRINT	HISTORIA DE USUARIO	PENDIENTE	EN DESARROLLO	TERMINADO
1	Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal.			X
	Gestión de Proveedores			X
	Gestión de Clientes		X	
2	Gestión de Inventario	X		
	Gestión de Kárdex	X		
3	Facturación electrónica	X		
	Crear el Dashboard	X		

4	Crear el Login	X		
	Autenticación y roles de usuario	X		
	Publicar la aplicación web	X		

En la Figura 13 se presenta el avance que se tiene en el Sprint 1 mediante el Burndown Chart, en el que se aprecia que los tiempos esperados para culminar el desarrollo del proyecto se mantienen.

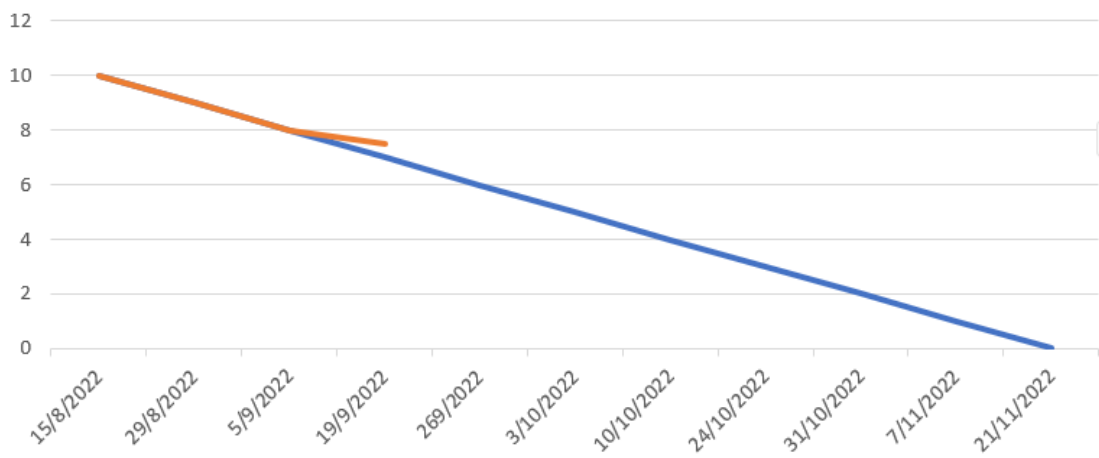


Fig. 13 Avance del Burndown Chart del Sprint 1.

Fuente: Autor

3.4.1.3. Gestión de Clientes

En la Figura 14 se muestra la pantalla para la gestión de clientes.



Fig. 14 Interfaz "Clientes".

Fuente: Autor

En la Figura 15 se aprecian los campos necesarios para el registro de clientes.

Detalle del Cliente

Cédula

Nombres

Dirección

Teléfono

Email

Profesión

Botón para guardar un Cliente → Guardar

Fig. 15 Interfaz para registrar un nuevo Cliente.

Fuente: Autor

Se detalla a continuación la Tabla XIX donde se aprecia la lista de tareas correspondiente al Sprint 1 y la tarea “Gestión de Clientes” se encuentra terminada.

Tabla XIX Lista de tareas final del Sprint 1.

Fuente: Autor

LISTA DE TAREAS DEL SPRINT 1				
Inicio: 15/Agosto/2022		Autor: Silvio David Pacheco Herrera		
Fin: 05/Septiembre/2022				
SPRINT	HISTORIA DE USUARIO	PENDIENTE	EN DESARROLLO	TERMINADO
1	Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal.			X
	Gestión de Proveedores			X
	Gestión de Clientes			X

2	Gestión de Inventario	X		
	Gestión de Kárdex	X		
3	Facturación electrónica	X		
	Crear el Dashboard	X		
4	Crear el Login	X		
	Autenticación y roles de usuario	X		
	Publicar la aplicación web	X		

En la Figura 16 se presenta el avance que se tiene en el Sprint 1 mediante el Burndown Chart, en el que se aprecia que el Sprint 1 ha finalizado y que los tiempos esperados para culminar el desarrollo del proyecto se mantienen.

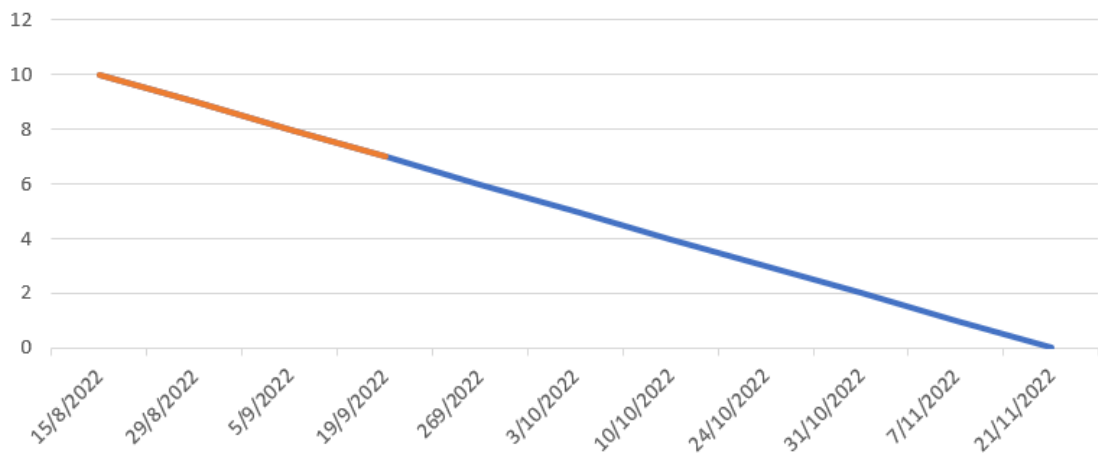


Fig. 16 Burndown Chart final del Sprint 1.

Fuente: Autor

3.4.2. Sprint 2

En el Sprint 2 las tareas a desarrollar son las siguientes:

- Gestión de Inventario
- Gestión de Kárdex

Se detalla a continuación la Tabla XX donde se aprecia la lista de tareas correspondiente al Sprint 2 y la tarea “Gestión de Inventario” se encuentra en desarrollo.

Tabla XX Lista de tareas Sprint 2.

Fuente: Autor

LISTA DE TAREAS DEL SPRINT 2				
Inicio: 05/Septiembre/2022		Autor: Silvio David Pacheco Herrera		
Fin: 26/Septiembre/2022				
SPRINT	HISTORIA DE USUARIO	PENDIENTE	EN DESARROLLO	TERMINADO
1	Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal.			X
	Gestión de Proveedores			X
	Gestión de Clientes			X
2	Gestión de Inventario		X	
	Gestión de Kárdex	X		
3	Facturación electrónica	X		
	Crear el Dashboard	X		
4	Crear el Login	X		
	Autenticación y roles de usuario	X		
	Publicar la aplicación web	X		

Tabla XX. Fuente: Autor

En la Figura 17 se presenta el avance que se tiene en el Sprint 2 mediante el Burndown Chart.

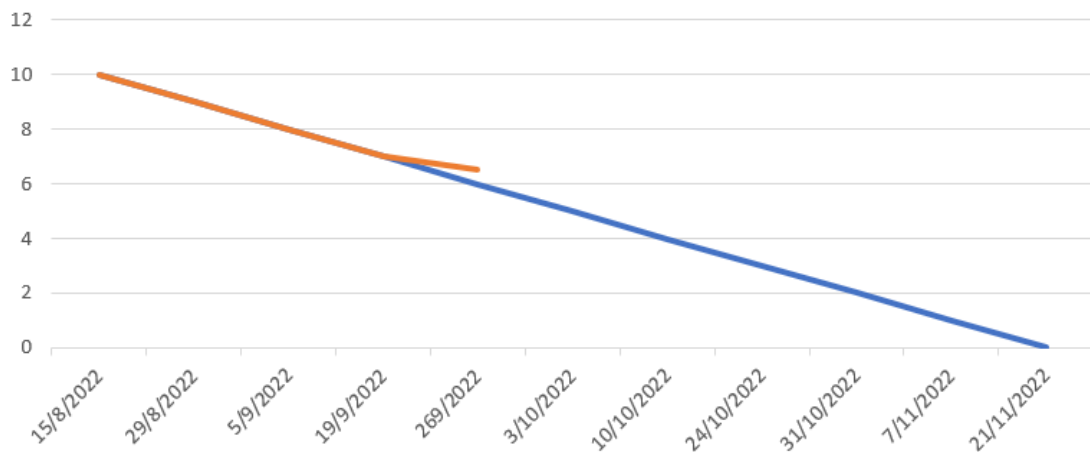


Fig. 17 Burndown Chart Sprint 2.

Fuente: Autor

3.4.2.1. Gestión de Inventario

En la Figura 18 se muestra la pantalla para la gestión de inventario, permitiendo al usuario insertar, editar y eliminar de la base de datos a los productos existentes en la empresa.



Fig. 18 Interfaz "Productos".

Fuente: Autor

En la Figura 19 se aprecian los campos necesarios para el ingreso de nuevos productos.

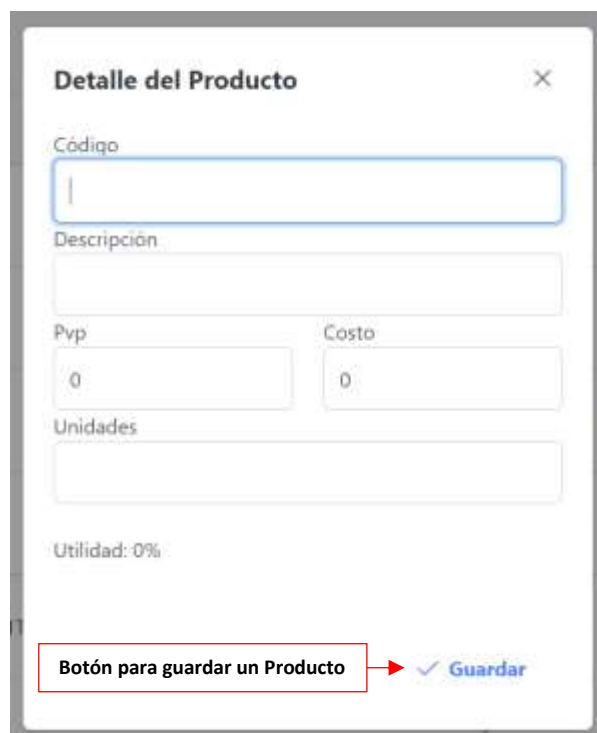


Fig. 19 Interfaz para agregar un nuevo Producto.

Fuente: Autor

Se detalla a continuación la Tabla XXI donde se aprecia la lista de tareas correspondiente al Sprint 2 y la tarea “Gestión de Inventario” se encuentra terminada.

Tabla XXI Avance de la lista de tareas Sprint 2.

Fuente: Autor

LISTA DE TAREAS DEL SPRINT 2				
Inicio: 05/Septiembre/2022		Autor: Silvio David Pacheco Herrera		
Fin: 26/Septiembre/2022				
SPRINT	HISTORIA DE USUARIO	PENDIENTE	EN DESARROLLO	TERMINADO
1	Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal.			X
	Gestión de Proveedores			X
	Gestión de Clientes			X
2	Gestión de Inventario			X
	Gestión de Kárdex		X	
3	Facturación electrónica	X		
	Crear el Dashboard	X		
4	Crear el Login	X		
	Autenticación y roles de usuario	X		
	Publicar la aplicación web	X		

En la Figura 20 se presenta el avance que se tiene en el Sprint 2 mediante el Burndown Chart, en el que se aprecia que los tiempos esperados para culminar el desarrollo del proyecto se mantienen.

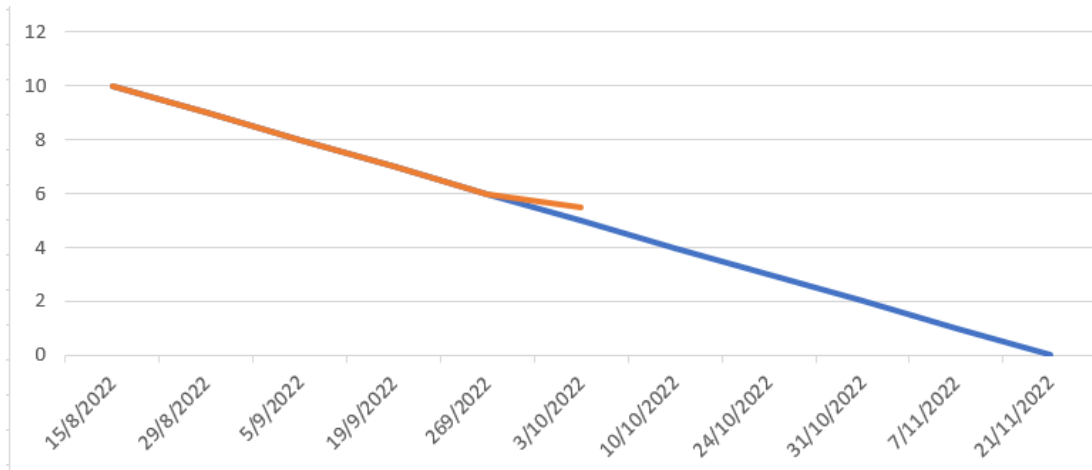


Fig. 20 Avance del Burndown Chart del Sprint 2.

Fuente: Autor

3.4.2.2. Gestión de Kárdex

En la Figura 21 se muestra la pantalla para la gestión de kárdex, permitiendo al usuario recibir y retirar productos de acuerdo a las transacciones que se realicen en la empresa con respecto a los productos de la misma, de igual manera un listado por producto de los movimientos realizados en el stock.



Fig. 21 Interfaz "Kárdex".

Fuente: Autor

En la Figura 22 se aprecian los campos requeridos para registrar si se reciben o retiran unidades en el stock de un producto en específico.

Recibir Unidades

Producto: CAB-IPH

Proveedor: Proveedor...

Fecha: dd/mm/aaaa

Referencia:

Cantidad:

Costo: 9

Balance: 0

Botón para guardar el detalle de un Kárdex → Guardar

Fig. 22 Interfaz para recibir o retirar unidades en el stock de un Producto.

Fuente: Autor

En la Figura 23 se muestra un ejemplo de cómo funciona el listado de los movimientos realizados en el stock de un producto en específico.

Kárdex - CARG-SAMTC

Tabla para mostrar el detalle de Kárdex de un producto

Referencia	Fecha	Costo	Cantidad	Balance
Eliminación de Venta	Sep 7, 2022	\$10.00	1	10
Venta de Productos	Sep 7, 2022	\$10.00	1	8
Venta de Productos	Sep 14, 2022	\$10.00	-1	9
Venta de Productos	Sep 22, 2022	\$10.00	-4	5
Venta de Productos	Oct 5, 2022	\$10.00	-1	4

Fig. 23 Interfaz del Kárdex de un Producto.

Fuente: Autor

Se detalla a continuación la Tabla XXII donde se aprecia la lista de tareas correspondiente al Sprint 2 y la tarea “Gestión de Kárdex” se encuentra terminada.

Tabla XXII Lista de tareas final Sprint 2:

Fuente: Autor

LISTA DE TAREAS DEL SPRINT 2				
Inicio: 05/Septiembre/2022		Autor: Silvio David Pacheco Herrera		
Fin: 26/Septiembre/2022				
SPRINT	HISTORIA DE USUARIO	PENDIENTE	EN DESARROLLO	TERMINADO
1	Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal.			X
	Gestión de Proveedores			X
	Gestión de Clientes			X
2	Gestión de Inventario			X
	Gestión de Kárdex			X
3	Facturación electrónica	X		
	Crear el Dashboard	X		
4	Crear el Login	X		
	Autenticación y roles de usuario	X		
	Publicar la aplicación web	X		

En la Figura 24 se presenta el avance que se tiene en el Sprint 2 mediante el Burndown Chart, en el que se aprecia que el Sprint 2 ha finalizado y que los tiempos esperados para culminar el desarrollo del proyecto se mantienen.

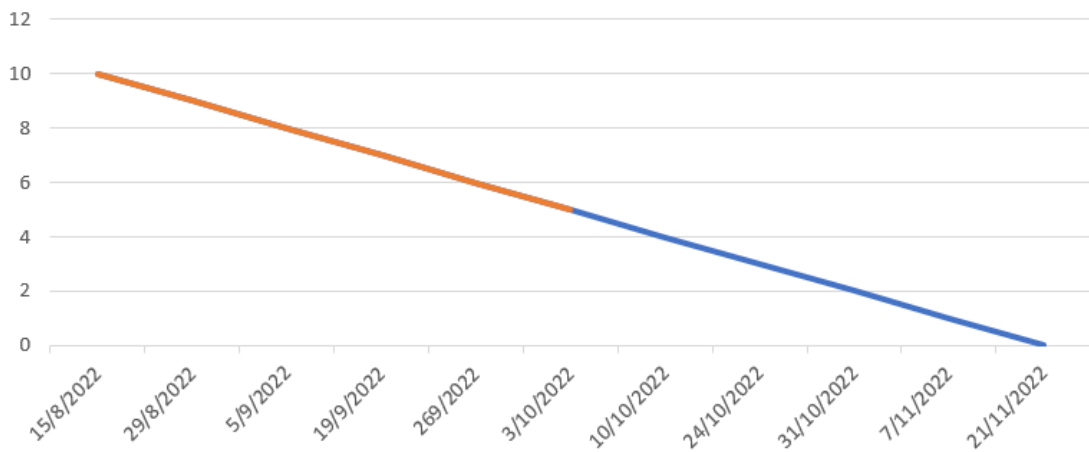


Fig. 24 Burndown Chart final del Sprint 2.

Fuente: Autor

3.4.3. Sprint 3

En el Sprint 3 las tareas a desarrollar son las siguientes:

- Facturación electrónica
- Crear el Dashboard

Se detalla a continuación la Tabla XXIII donde se aprecia la lista de tareas correspondiente al Sprint 3 y la tarea “Facturación electrónica” se encuentra en desarrollo.

Tabla XXIII Lista de tareas Sprint 3.

Fuente: Autor

LISTA DE TAREAS DEL SPRINT 3				
Inicio: 26/Septiembre/2022		Autor: Silvio David Pacheco Herrera		
Fin: 31/Octubre/2022				
SPRINT	HISTORIA DE USUARIO	PENDIENTE	EN DESARROLLO	TERMINADO
1	Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal.			X
	Gestión de Proveedores			X
	Gestión de Clientes			X
2	Gestión de Inventario			X
	Gestión de Kárdex			X
3	Facturación electrónica		X	
	Crear el Dashboard	X		

4	Crear el Login	X		
	Autenticación y roles de usuario	X		
	Publicar la aplicación web	X		

En la Figura 25 se presenta el avance que se tiene en el Sprint 3 mediante el Burndown Chart.

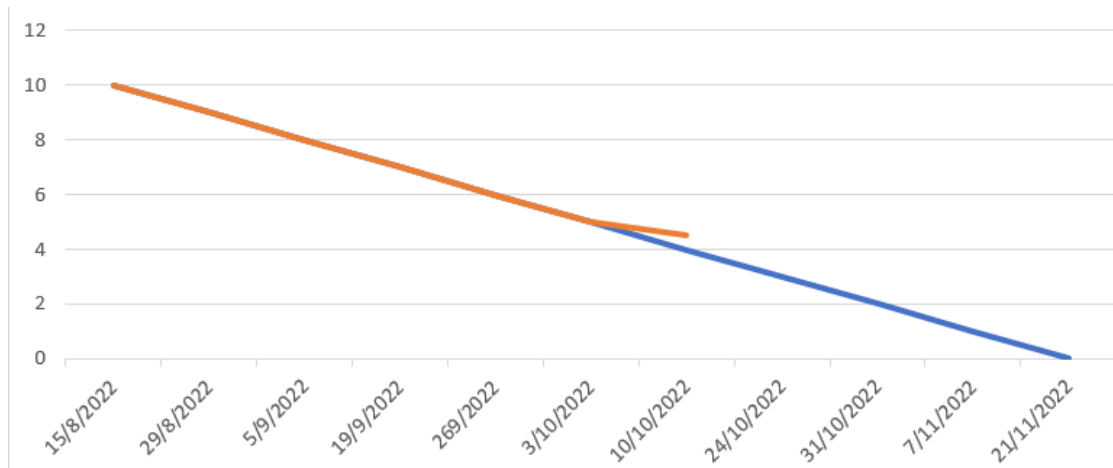


Fig. 25 Burndown Chart Sprint 3.

Fuente: Autor

3.4.3.1. Facturación Electrónica

La idea inicial fue desarrollar un sistema que permita emitir comprobantes físicos a los clientes por medio de la impresión de facturas. Sin embargo, la situación actual de la empresa requiere la emisión de comprobantes electrónicos debido a lo establecido por el SRI.

Por tal motivo se acordó la implementación de la emisión de comprobantes electrónicos dentro del aplicativo web, para este proceso la empresa DigitalSoft de la ciudad de Cañar que cuenta con un servicio activo de facturación electrónica en ambiente de pruebas y producción brindará a la empresa Servcomp los recursos necesarios para la emisión de comprobantes electrónicos.

Los gerentes de ambas empresas llegaron a un acuerdo para inicialmente hacer uso del servicio de facturación electrónica en ambiente de pruebas de manera gratuita y posteriormente pasar al ambiente de producción por un bajo costo.

En la Figura 26 se muestra la pantalla principal para realizar una factura, donde el botón crear permite empezar con el proceso de realizar una factura y el botón listado permite visualizar las ventas realizadas en la empresa y sus detalles.

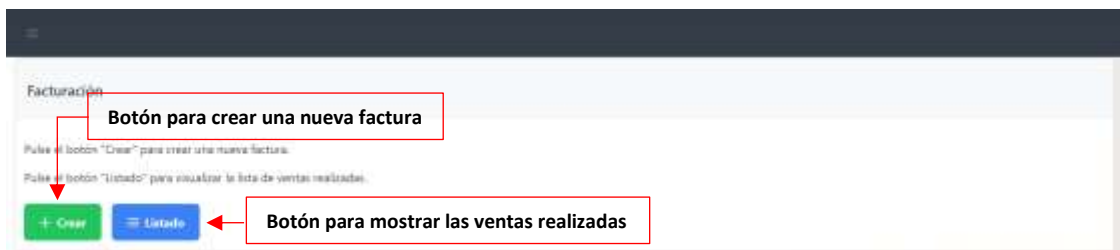


Fig. 26 Interfaz inicial "Facturación".

Fuente: Autor

En la Figura 27 se aprecia una pantalla para seleccionar el cliente a facturar y la lista de productos a facturar.

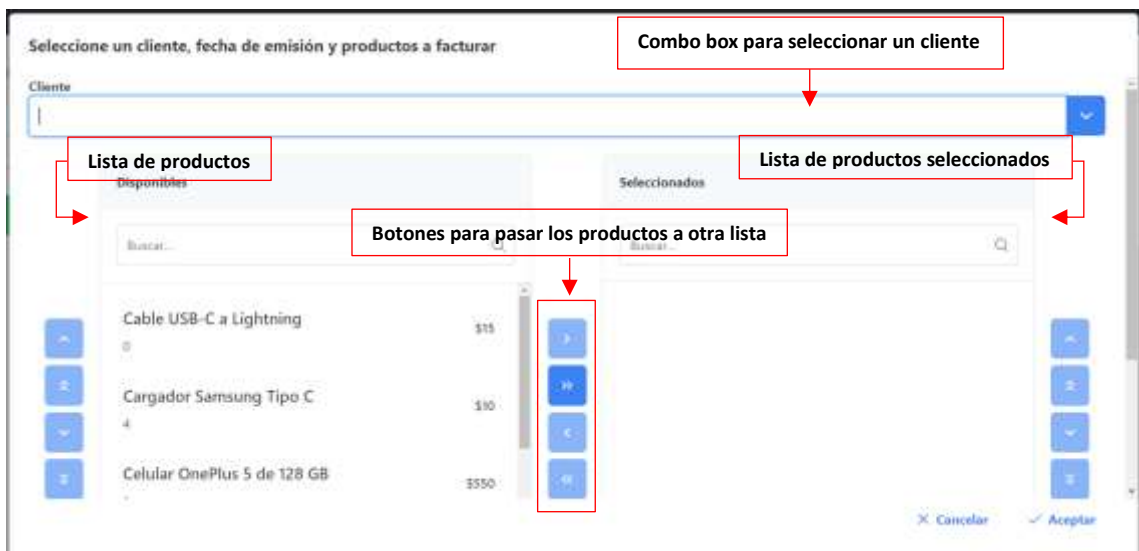


Fig. 27 Interfaz para seleccionar un Cliente y Productos a facturar.

Fuente: Autor

En la Figura 28 se presenta la pantalla de facturación con los productos seleccionados a facturar.



Fig. 28 Interfaz de Facturación.

Fuente: Autor

En la Figura 29 se muestra la pantalla del listado de las ventas realizadas.

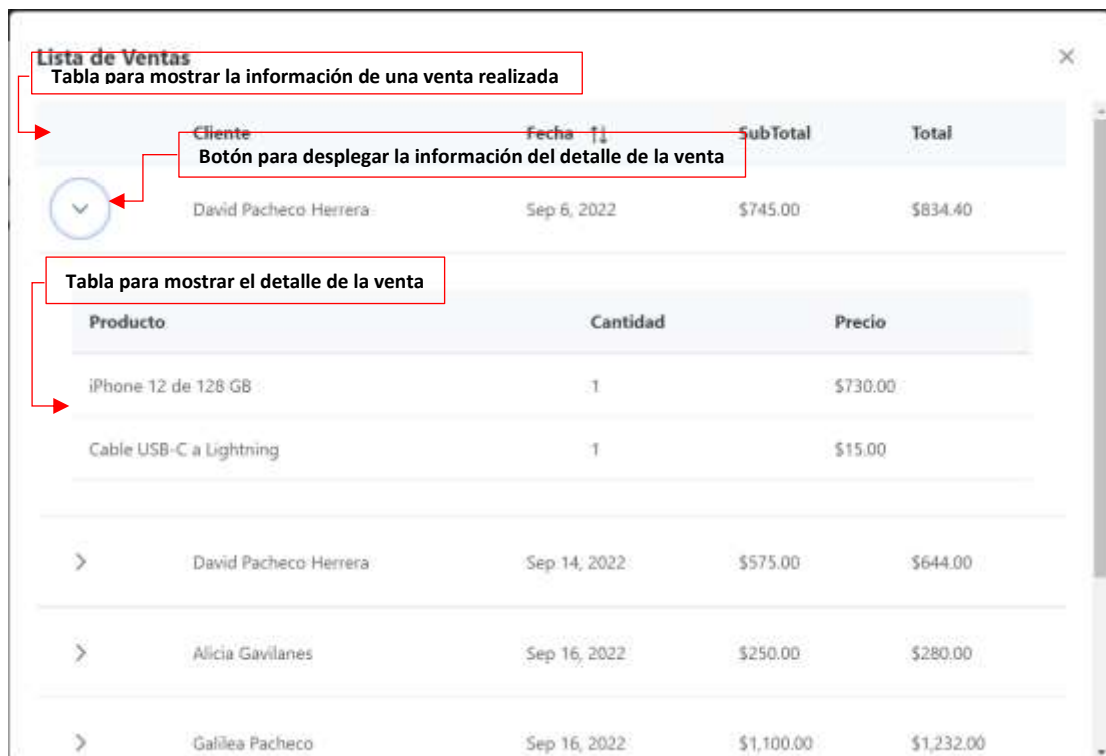


Fig. 29 Interfaz de la lista de ventas realizadas.

Fuente: Autor

En la Figura 30 se presenta el formato de los XML para el envío a los Web Services del SRI.

En la Figura 31 se presenta el avance que se tiene en el Sprint 3 mediante el Burndown Chart.

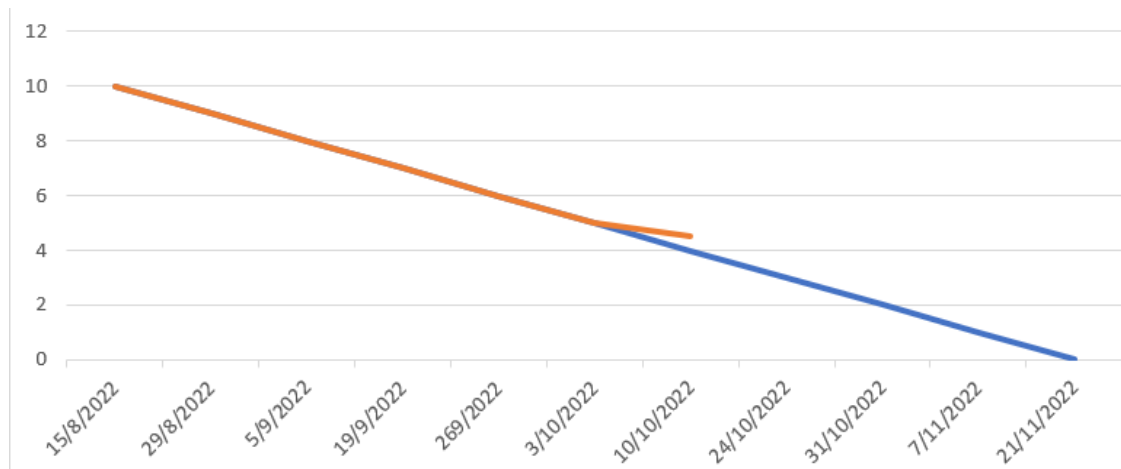


Fig. 31 Avance del Burndown Chart del Sprint 3.

Fuente: Autor

3.4.3.2. Crear el Dashboard

En la Figura 32 se muestra la pantalla del Dashboard la cual a su vez es la página principal de la aplicación.

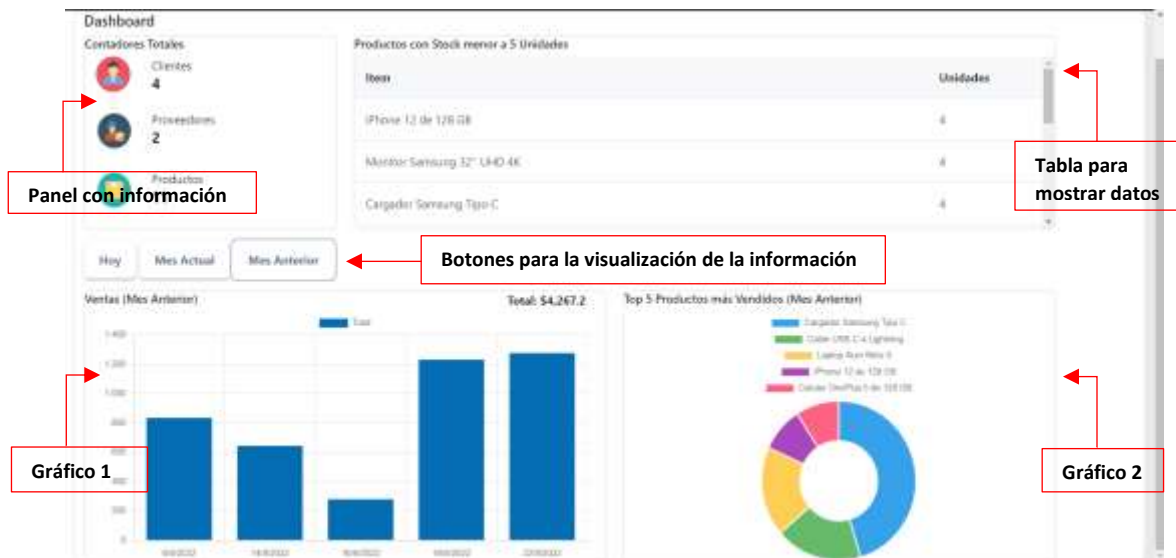


Fig. 32 Interfaz del Dashboard.

Fuente: Autor

Se detalla a continuación la Tabla XXV donde se aprecia la lista de tareas correspondiente al Sprint 3 y la tarea “Crear el Dashboard” se encuentra en desarrollo debido a ciertos contratiempos existentes en el desarrollo de esta actividad.

Tabla XXV Lista de tareas final Sprint 3.

Fuente: Autor

LISTA DE TAREAS DEL SPRINT 3				
Inicio: 26/Septiembre/2022		Autor: Silvio David Pacheco Herrera		
Fin: 31/Octubre/2022				
SPRINT	HISTORIA DE USUARIO	PENDIENTE	EN DESARROLLO	TERMINADO
1	Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal.			X
	Gestión de Proveedores			X
	Gestión de Clientes			X
2	Gestión de Inventario			X
	Gestión de Kárdex			X
3	Facturación electrónica			X
	Crear el Dashboard		X	X
4	Crear el Login	X		
	Autenticación y roles de usuario	X		
	Publicar la aplicación web	X		

En la Figura 33 se presenta el avance que se tiene en el Sprint 3 mediante el Burndown Chart, en el que se aprecia que el sprint actual ha finalizado y que los tiempos esperados para culminar el desarrollo del proyecto se mantienen.

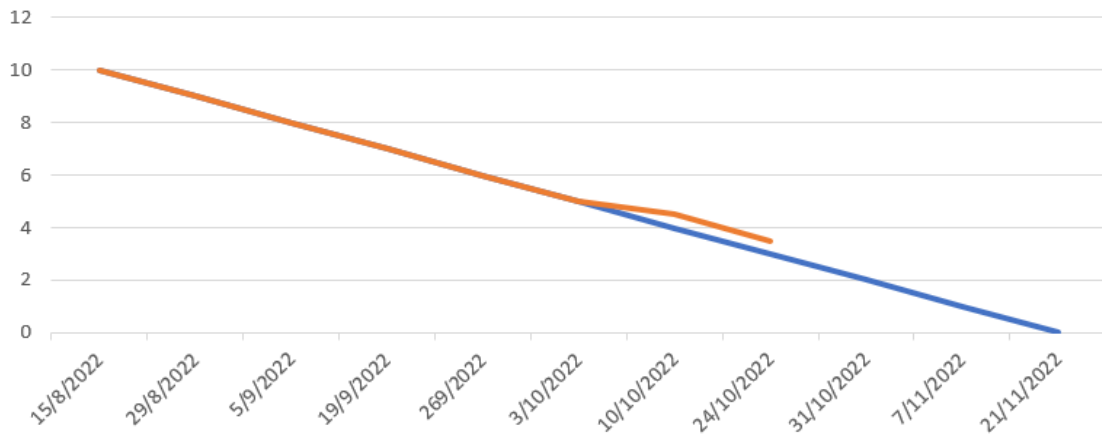


Fig. 33 Burndown Chart final del Sprint 3.

Fuente: Autor

3.4.4. Sprint 4

En el Sprint 4 las tareas a desarrollar son las siguientes:

- Crear el Login
- Autenticación y roles de usuario
- Publicar la aplicación web

Se detalla a continuación la Tabla XXVI donde se aprecia la lista de tareas correspondiente al Sprint 4 y la tarea “Crear el Login” se encuentra en desarrollo.

Tabla XXVI Lista de tareas Sprint 4.

Fuente: Autor

LISTA DE TAREAS DEL SPRINT 4				
Inicio: 31/Octubre/2022		Autor: Silvio David Pacheco Herrera		
Fin: 21/Noviembre/2022				
SPRINT	HISTORIA DE USUARIO	PENDIENTE	EN DESARROLLO	TERMINADO
1	Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal.			X
	Gestión de Proveedores			X
	Gestión de Clientes			X
2	Gestión de Inventario			X
	Gestión de Kárdex			X
3	Facturación electrónica			X

	Crear el Dashboard		X	
4	Crear el Login		X	
	Autenticación y roles de usuario	X		
	Publicar la aplicación web	X		

En la Figura 34 se presenta el avance que se tiene en el Sprint 4 mediante el Burndown Chart.

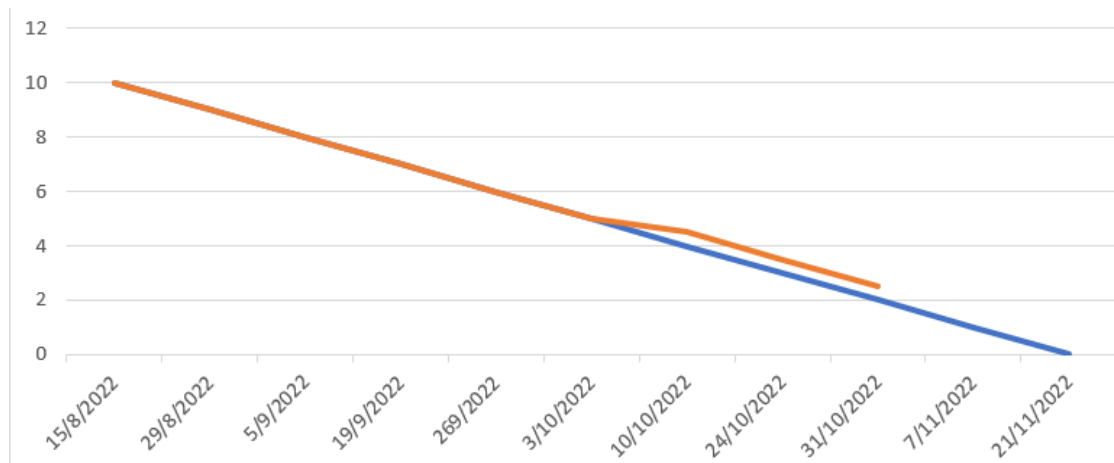


Fig. 34 Burndown Chart Sprint 4.

Fuente: Autor

3.4.4.1. Crear el Login

En la Figura 35 se muestra la página para el inicio de sesión de los usuarios.



Fig. 35 Interfaz Inicio de Sesión.

Fuente: Autor

Adicionalmente, se diseñó un panel de menú el cual se presenta en la Figura 36 y que sirve para navegar por las diferentes páginas de la aplicación web.

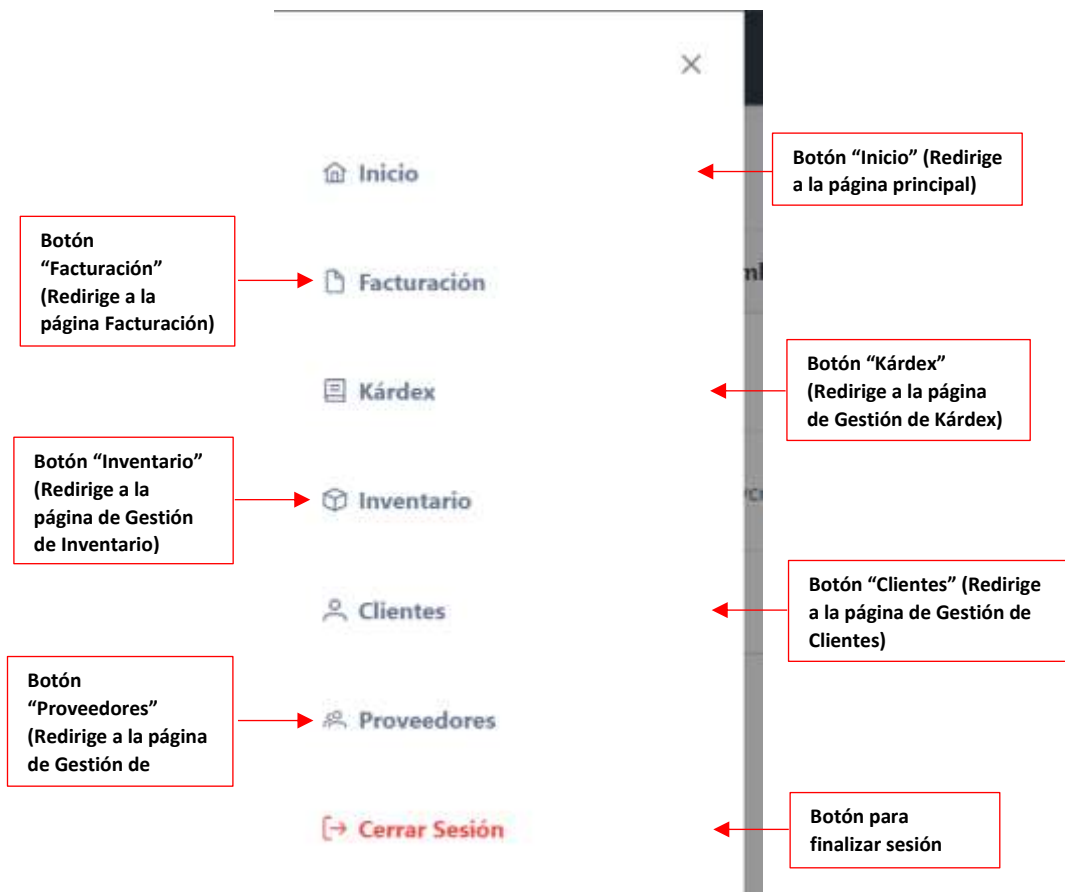


Fig. 36 Panel de menú.

Fuente: Autor

Se detalla a continuación la Tabla XXVII donde se aprecia la lista de tareas correspondiente al Sprint 4 y la tarea “Crear el Login” se encuentra terminada. De igual manera la tarea “Crear el Dashboard” correspondiente al Sprint 3 y que seguía en desarrollo ha concluido.

Tabla XXVII Avance de la lista de tareas Sprint 4.

Fuente: Autor

LISTA DE TAREAS DEL SPRINT 4				
Inicio: 31/Octubre/2022		Autor: Silvio David Pacheco Herrera		
Fin: 21/Noviembre/2022				
SPRINT	HISTORIA DE USUARIO	PENDIENTE	EN DESARROLLO	TERMINADO
1	Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal.			X
	Gestión de Proveedores			X

	Gestión de Clientes			X
2	Gestión de Inventario			X
	Gestión de Kárdex			X
3	Facturación electrónica			X
	Crear el Dashboard			X
4	Crear el Login			X
	Autenticación y roles de usuario		X	
	Publicar la aplicación web	X		

En la Figura 37 se presenta el avance que se tiene en el Sprint 4 mediante el Burndown Chart, en el que se aprecia que los tiempos esperados para culminar el desarrollo del proyecto se mantienen.

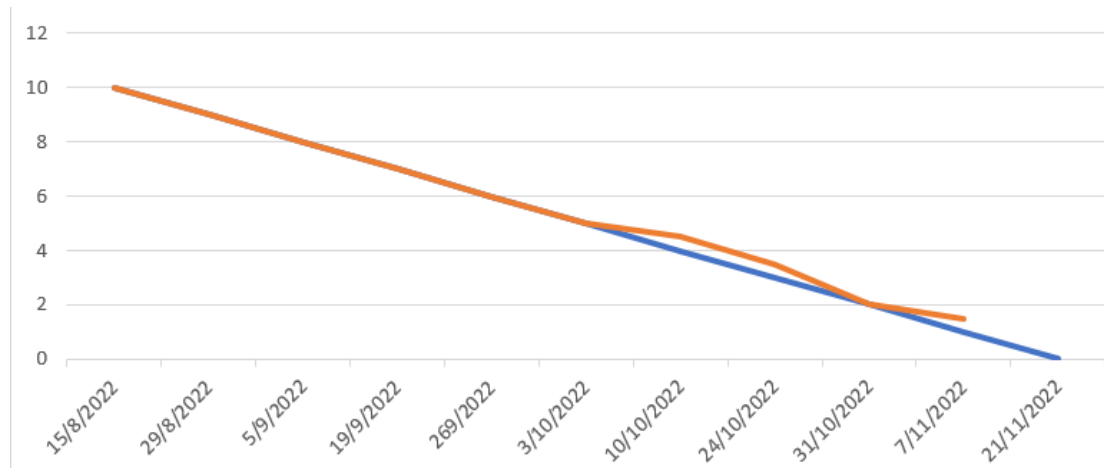


Fig. 37 Avance del Burndown Chart del Sprint 4.

Fuente: Autor

3.4.4.2. Autenticación y Roles de Usuario

En la Figura 38 se muestra la creación de los usuarios mediante la herramienta Firebase Authentication haciendo uso de su servicio de autenticación de correo electrónico y contraseña.

Agrega un usuario con correo electrónico y contraseña

Correo electrónico:

Contraseña:

Cancelar Agregar usuario

Fig. 38 Creación de Usuarios.

Fuente: Autor

En la Figura 39 se presentan los usuarios registrados y que podrán acceder a la aplicación web.

Buscar por dirección de correo electrónico, número de teléfono o UID de usuario Agregar usuario

Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario
galypachecoh@gmail.com	✉	8 nov 2022		6IWzok8G5tY6as8KWLADgVQFV3...
rinayya@hotmail.com	✉	8 nov 2022		ELuhCu91kcRKFezJeyWzfrU88vJ3
efversp@hotmail.es	✉	26 sept 2022	8 nov 2022	Rj7K7VA9qZa5BCgcV2gsJJ4Z6HK2
davidph5469@gmail.com	✉	18 jul 2022	26 sept 2022	hVC4FpaetqdJw7DDswwNAvQIRJj1

Filas por página: 50 1 - 4 of 4

Fig. 39 Lista de usuarios.

Fuente: Autor

En la Figura 40 se muestra el ingreso de los usuarios en la base de datos para posteriormente autenticarlos mediante su rol de usuario.

ventas-tesis	usuarios	hVC4FpaetqdJw7DDswwNAvQIRJj1
+ Iniciar colección	+ Agregar documento	+ Iniciar colección
clientes	6IWzok8G5tY6as8KWLADgVQFV3p1	+ Agregar campo
comprobantes	ELuhCu91kcRKFezJeyWzfrU88vJ3	administrador: true
kardex	Rj7K7VA9qZa5BCgcV2gsJJ4Z6HK2	email: "davidph5469@gmail.com"
productos	hVC4FpaetqdJw7DDswwNAvQIRJj1 >	nombres: "David Pacheco"
proveedores		
usuarios >		
ventas		

Fig. 40 Usuarios registrados en la base de datos.

Fuente: Autor

Se detalla a continuación la Tabla XXVIII donde se aprecia la lista de tareas correspondiente al Sprint 4 y la tarea “Autenticación y roles de usuario” se encuentra terminada.

Tabla XXVIII Avance de la lista de tareas Sprint 4.

Fuente: Autor

LISTA DE TAREAS DEL SPRINT 4				
Inicio: 31/Octubre/2022		Autor: Silvio David Pacheco Herrera		
Fin: 21/Noviembre/2022				
SPRINT	HISTORIA DE USUARIO	PENDIENTE	EN DESARROLLO	TERMINADO
1	Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal.			X
	Gestión de Proveedores			X
	Gestión de Clientes			X
2	Gestión de Inventario			X
	Gestión de Kárdex			X
3	Facturación electrónica			X
	Crear el Dashboard			X
4	Crear el Login			X
	Autenticación y roles de usuario			X
	Publicar la aplicación web		X	

En la Figura 41 se presenta el avance que se tiene en el Sprint 4 mediante el Burndown Chart, en el que se aprecia que los tiempos esperados para culminar el desarrollo del proyecto se mantienen.

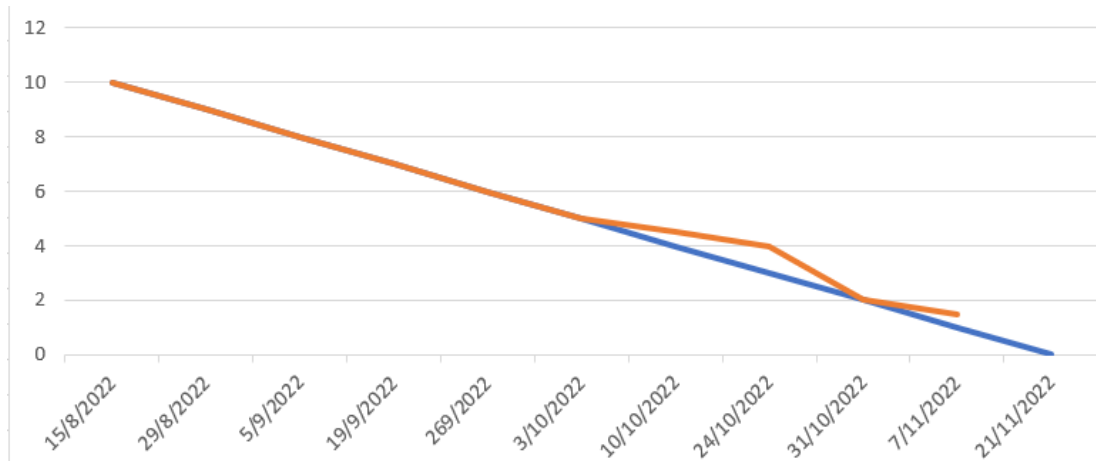


Fig. 41 Avance del Burndown Chart del Sprint 4.

Fuente: Autor

3.4.4.3. Publicar la Aplicación Web

En la Figura 42 se procede a poner la aplicación web en producción mediante Firebase Hosting.

```

Firebase Hosting and (optionally) set up GitHub Action deploys

=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Please select an option: Use an existing project
? Select a default Firebase project for this directory: (Use arrow keys)
> ventas-tesis (ventas-tesis)

```

Fig. 42 Producción de la aplicación.

Fuente: Autor

En la Figura 43 se presenta el despliegue de la aplicación.

```
=== Deploying to 'ventas-tesis'...  
  
i deploying hosting  
i hosting[ventas-tesis]: beginning deploy...  
i hosting[ventas-tesis]: found 1 files in public  
+ hosting[ventas-tesis]: file upload complete  
i hosting[ventas-tesis]: finalizing version...  
+ hosting[ventas-tesis]: version finalized  
i hosting[ventas-tesis]: releasing new version...  
+ hosting[ventas-tesis]: release complete  
  
+ Deploy complete!
```

Fig. 43 Despliegue de la aplicación.

Fuente: Autor

En la Figura 44 se visualiza la pantalla de inicio de sesión para acceder a la aplicación a través del siguiente link: (<https://ventas-tesis.web.app/>).



Fig. 44 Página web.

Fuente: Autor

Se detalla a continuación la Tabla XXIX donde se aprecia la lista de tareas correspondiente al Sprint 3 y la tarea “Publicar la aplicación web” se encuentra terminada.

Tabla XXIX Lista de tareas final Sprint 4.

Fuente: Autor

LISTA DE TAREAS DEL SPRINT 4	
Inicio: 31/Octubre/2022	Autor: Silvio David Pacheco Herrera

Fin: 21/Noviembre/2022				
SPRINT	HISTORIA DE USUARIO	PENDIENTE	EN DESARROLLO	TERMINADO
1	Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal.			X
	Gestión de Proveedores			X
	Gestión de Clientes			X
2	Gestión de Inventario			X
	Gestión de Kárdex			X
3	Facturación electrónica			X
	Crear el Dashboard			X
4	Crear el Login			X
	Autenticación y roles de usuario			X
	Publicar la aplicación web			X

En la Figura 45 se presenta el avance que se tiene en el Sprint 4 mediante el Burndown Chart, en el que se aprecia que el sprint actual ha finalizado, que los tiempos esperados para culminar el desarrollo del proyecto se han cumplido y que la ejecución del proyecto ha llegado a su fin.

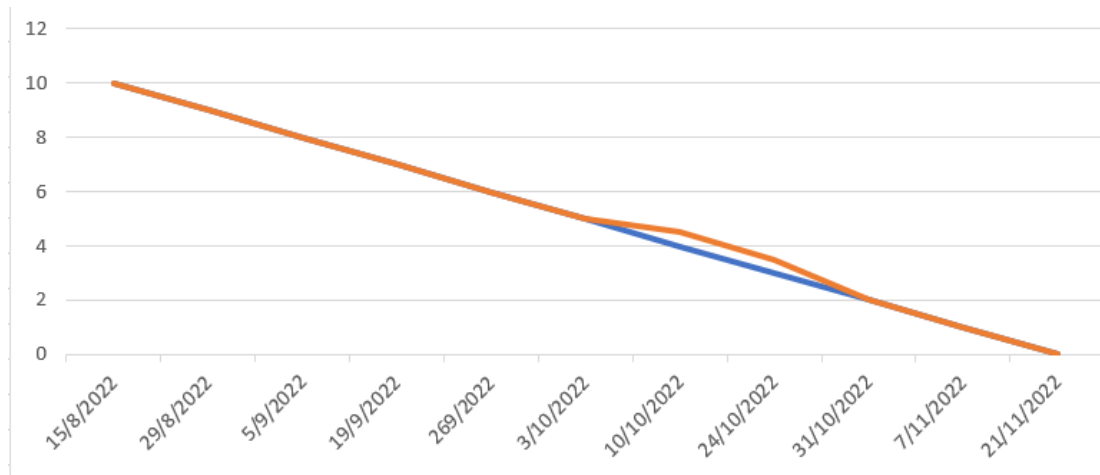


Fig. 45 Burndown Chart final del Sprint 4.

Fuente: Autor

3.5. Revisión y Retroalimentación

Como se planteó en la Planificación del Sprint se llevaron a cabo las respectivas reuniones con los miembros de la empresa las cuales sirvieron para determinar el avance del

proyecto, verificar si el aplicativo funcionaba correctamente, determinar si el diseño era adecuado y agradable para el personal y principalmente realizar modificaciones si ese fuera el caso.

Este proceso permitió obtener un producto final funcional y de calidad, en el Anexo ... se presenta una carta de aceptación firmada por el gerente de la empresa Servcomp.

3.6. Cierre

Luego de haber concluido con el último sprint se da por concluido el desarrollo del proyecto, cabe recalcar que la existencia de contratiempos en la ejecución del mismo no causó un impacto significativo para la entrega del producto final ya que el tiempo restante en el desarrollo de ciertos sprint permitieron balancear la ejecución de aquellos que requerían más tiempo de esfuerzo.

En la Tabla XXX se aprecia el informe de entregable final, valorando la funcionalidad de la aplicación web luego de haber concluido los sprint.

Tabla XXX Informe de entregable final.

Fuente: Autor

INFORME DE ENTREGABLE FINAL				
HISTORIA DE USUARIO	EXPECTATIVA	RESPUESTA DE LA APLICACIÓN	AVANCE (%)	OBSERVACIÓN
Almacenar y centralizar la información referente al proceso de facturación de la matriz y sucursal.	Permite ingresar, eliminar, editar, recuperar y visualizar la información de la base de datos desde cualquier sitio.	Recupera la información en formato JSON para el posterior manejo de la misma.	10%	Ninguna
Gestión de Proveedores	Insertar, editar, listar y eliminar Proveedores.	Las transacciones se reflejan en la base de datos.	20%	Ninguna
Gestión de Clientes	Insertar, editar, listar y eliminar Clientes.	Las transacciones se reflejan en la base de datos.	30%	Ninguna

Gestión de Inventario	Insertar, editar, listar y eliminar Productos.	Las transacciones se reflejan en la base de datos.	40%	Ninguna
Gestión de Kárdex	Recibir y retirar unidades de un producto.	Aumenta o disminuye stock de un producto. Lista el detalle por producto de las transacciones realizadas.	50%	Necesita información cargada en los anteriores.
Facturación electrónica	Guardar las ventas realizadas y emitir comprobantes electrónicos a los clientes.	La información se muestra en la base de datos y los comprobantes se envían al correo electrónico del cliente.	70%	Necesita información cargada en los anteriores; Es necesario que el cliente a facturar cuente con una cuenta de correo electrónico activa.
Crear el Dashboard	Carga la información solicitada.	Presenta los gráficos correspondientes al usuario.	80%	Necesita información relacionada a ventas.
Crear el Login	Iniciar sesión mediante correo electrónico y contraseña.	Acceso a la aplicación luego de iniciar sesión.	90%	Los usuarios deben ser creados posteriormente.
Autenticación y roles de usuario	Mostrar ciertos módulos, recursos e información del sistema a un usuario en concreto.	Bloquea ciertos elementos de la aplicación a los usuarios "Empleados".	95%	Ninguna
Publicar la aplicación web	Permite acceder a la aplicación mediante conexión a internet.	Permite visualizar el sistema haciendo uso de un navegador web.	100%	Ninguna

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

Posterior al desarrollo e implantación de la solución en la empresa Servcomp se comprueba que por medio del almacenamiento en la nube la empresa puede contar con una sola fuente de información lo cual permite al gerente administrar de mejor manera el inventario, las ventas realizadas y los comprobantes emitidos en la matriz y sucursal.

La implementación de la facturación electrónica en el sistema permitió a la empresa adaptarse a los requisitos establecidos por el SRI, a largo plazo ayudará a que la empresa ahorre capital ya que los comprobantes son emitidos por medio de correo electrónico a los clientes, eliminando la necesidad de adquirir facturas preimpresas o equipos físicos para imprimir recibos en papel.

Aplicar la metodología Scrum permitió que existiera una mejor comunicación con el cliente para saber si el avance que se tenía en el proyecto cumplía con sus expectativas y ayudó a entregar el producto final de manera oportuna y eficaz, puesto que esta metodología genera versiones del producto que van siendo evaluadas por el usuario, que en este caso corresponde al gerente de la empresa Servcomp.

4.2. Recomendaciones

A la Universidad Católica de Cuenca Campus Azogues:

- Promover proyectos en el área de desarrollo de software en general ya que es una actividad que se encuentra en constante evolución, existiendo cada vez nuevos lenguajes de programación, nuevas tecnologías de desarrollo como hosting, almacenamiento, analítica, etc.

A la empresa Servcomp:

- Se recomienda contratar los servicios de pago de Firebase ya que la licencia Spark que proporciona funciones gratuitamente no es suficiente para la cantidad de datos que se van a almacenar en la nube.
- Se recomienda darle seguimiento a la aplicación implementada para mantener adecuadamente la base de datos y su información y de igual manera pasar a la fase de producción en el proceso de facturación electrónica.

REFERENCIAS BIBLIOGRÁFICAS

- [1] W. Ruiz-Martínez and A. A. González-Gómez, “An Approach from Software Engineering to an IoT and Machine Learning Technological Solution that Allows Monitoring and Controlling Environmental Variables in a Coffee Crop,” *Ingeniería*, vol. 26, no. 3, pp. 465–478, Jan. 2021, doi: 10.14483/23448393.18495.
- [2] B. Montero-Molina, H. Cevallos-Vite, and J. Dávila-Cuesta, “Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software,” *Espirales*, 2018, Accessed: Nov. 10, 2022. [Online]. Available: https://www.researchgate.net/profile/Harry-Vite-Cevallos/publication/327537074_Metodologias_agiles_frente_a_las_tradicionales_en_el_proceso_de_desarrollo_de_software/links/5b942061a6fdccfd542a2b13/Metodologia-s-agiles-frente-a-las-tradicionales-en-el-proceso-de-desarrollo-de-software.pdf
- [3] E. Maida and J. Pacienza, “Metodologías de desarrollo de software,” PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA SANTA MARIA DE LOS BUENOS AIRES, 2015.
- [4] E. Jiménez-Hernández and S. Orantes-Jiménez, “Metodología Híbrida para Desarrollo de Software en México,” *Memorias la Segunda Conf. Iberoam. Complejidad, Informática y Cibernética (CICIC 2012)*, vol. 5, 2012.
- [5] D. Millet and A. Navarro, “Facturación electrónica: la búsqueda de la eficiencia y productividad,” pp. 88–95, 2008, Accessed: Nov. 10, 2022. [Online]. Available: www.partidadoble.es
- [6] J. Ramírez-Álvarez, N. Oliva, and M. Andino, “Facturación electrónica en Ecuador: Evaluación de impacto en el cumplimiento tributario,” 2018, Accessed: Nov. 10, 2022. [Online]. Available:

<https://publications.iadb.org/publications/spanish/document/Facturación-electrónica-en-Ecuador-Evaluación-de-impacto-en-el-cumplimiento-tributario.pdf>

- [7] Servicio de Rentas Internas, “Facturación Electrónica - intersri - Servicio de Rentas Internas,” 2022. <https://www.sri.gob.ec/facturacion-electronica#información> (accessed Nov. 10, 2022).
- [8] P. Palos-Sánchez, “Estudio organizacional del Cloud Computing en empresas emprendedoras.,” vol. 6, pp. 1–16, 2017, doi: 10.17993/3ctecno.2017.v6n2e22.
- [9] Amazon Web Services, “¿Qué es el almacenamiento en la nube?,” 2022. <https://aws.amazon.com/es/what-is/cloud-storage/> (accessed Nov. 10, 2022).
- [10] C. Mateu, *Desarrollo de aplicaciones web*. Catalunya: Eureca Media, SL, 2004.
- [11] M. R. Valarezo Pardo, J. A. Honores Tapia, A. S. Gómez Moreno, and L. F. Vínecs Sánchez, “Comparación de tendencias tecnológicas en aplicaciones web,” vol. 7, no. 3, pp. 28–49, 2018, doi: 10.17993/3ctecno.2018.v7n3e27.28-49/30.
- [12] Microsoft, “Documentation for Visual Studio Code,” 2022. <https://code.visualstudio.com/docs> (accessed Nov. 10, 2022).
- [13] Powered by Google, “Angular - What is Angular?,” 2022. <https://angular.io/guide/what-is-angular> (accessed Nov. 10, 2022).
- [14] C. F. Proaño Yépez, “DESARROLLO DE UN SISTEMA WEB PARA EL CONTROL DE LA PRODUCCIÓN DE UN INVERNADERO DE TOMATE EN EL CANTÓN COTACACHI, UTILIZANDO EL FRAMEWORK ANGULAR VERSIÓN 4,” UNIVERSIDAD TÉCNICA DEL NORTE, Ibarra, 2018. Accessed: Nov. 10, 2022. [Online]. Available: http://repositorio.utn.edu.ec/bitstream/123456789/9019/1/04_ISC_500_TRABAJO_DE_GRADO.pdf

- [15] OpenJS Foundation, “Acerca | Node.js,” 2022. <https://nodejs.org/es/about/> (accessed Nov. 10, 2022).
- [16] E. Haro, T. Guarda, A. O. Zambrano Peñaherrera, and G. Ninahualpa Quiña, “Desarrollo backend para aplicaciones web, Servicios Web Restful: Node.js vs Spring Boot,” *Rev. Ibérica Sist. e Tecnol. Informação*, pp. 309–321, 2019.
- [17] D. Villalón, “Crear y desarrollar una aplicación de alto rendimiento con bajo coste utilizando Flutter y Firebase,” Escola Tècnica Superior d’Enginyeria Informàtica Universitat Politècnica de València, 2021.
- [18] Google, “Documentación de Firebase,” 2022. <https://firebase.google.com/docs?hl=es-419> (accessed Nov. 10, 2022).
- [19] Google, “Firestore | Firebase,” 2022. <https://firebase.google.com/docs/firestore?hl=es-419> (accessed Nov. 10, 2022).
- [20] Google, “Firebase Authentication | Firebase Authentication,” 2022. <https://firebase.google.com/docs/auth?hl=es-419> (accessed Nov. 10, 2022).
- [21] Google, “Firebase Hosting,” 2022. <https://firebase.google.com/docs/hosting> (accessed Nov. 10, 2022).

ANEXOS

Anexo 1: Componente “Clientes”

```
applyFilterGlobal($event: any, stringVal: any) {
  this.dtl.filterGlobal({$event.target as HTMLInputElement}.value, stringVal);
}
openNew() {
  this.clienteExistente = false;
  this.manejarCliente = this.fb.group({
    cedula: ['', [Validators.required, Validators.maxLength(13)]],
    nombres: ['', Validators.required],
    direccion: ['', Validators.required],
    telefono: ['', Validators.required],
    email: ['', [Validators.required, Validators.email]],
    profesion: ['', Validators.required]
  });
  this.submitted = false;
  this.clienteDialog = true;
  this.editar = false;
}
```

```
hideDialog() {
  this.clienteDialog = false;
  this.submitted = false;
}
editarCliente(id: string) {
  this.clienteExistente = false;
  this._clientesService.getClientes(id).subscribe(data => {
    this.idCliente = id;
    this.manejarCliente = this.fb.group({
      cedula: [data.payload.data()['cedula'], [Validators.required, Validators.maxLength(13)]],
      nombres: [data.payload.data()['nombres'], Validators.required],
      direccion: [data.payload.data()['direccion'], Validators.required],
      telefono: [data.payload.data()['telefono'], Validators.required],
      email: [data.payload.data()['email'], [Validators.required, Validators.email]],
      profesion: [data.payload.data()['profesion'], Validators.required]
    });
  });
  this.clienteDialog = true;
  this.editar = true;
}
```

```

guardarCliente() {
  this.submitted = true;

  if (this.manejarCliente.invalid) {
    return;
  }
  const cliente: any = {
    cedula: this.manejarCliente.value.cedula,
    nombres: this.manejarCliente.value.nombres,
    direccion: this.manejarCliente.value.direccion,
    telefono: this.manejarCliente.value.telefono,
    email: this.manejarCliente.value.email,
    profesion: this.manejarCliente.value.profesion
  }

  if (this.editar == false) {
    for(let item in this.clientes){
      if(this.clientes[item].cedula == cliente.cedula){
        this.clienteExistente = true;
        return;
      }
    }
    this._clientesService.agregarCliente(cliente).then(() => {
      this.clienteDialog = false;
      this.toastr.success('El cliente fue registrado con éxito!', 'Cliente Registrado', {
        positionClass: 'toast-bottom-right'
      });
    }).catch(error => {
      this.toastr.error(this.firebError.firebaseError(error.code), 'Error', {
        positionClass: 'toast-bottom-right'
      });
    });
  }
  else {
    this._clientesService.editarCliente(this.idCliente, cliente).then(() =>{
      this.clienteDialog = false;
      this.toastr.info('El cliente fue modificado con éxito', 'Cliente Modificado', {
        positionClass: 'toast-bottom-right'
      });
    }).catch(error => {
      this.toastr.error(this.firebError.firebaseError(error.code), 'Error', {
        positionClass: 'toast-bottom-right'
      });
    });
  }
}
}
}

```

```

eliminarCliente(id: string, cliente: string) {
  this.confirmationService.confirm({
    message: '¿Está seguro de eliminar a ' + cliente + ' de la lista?',
    header: 'Atención',
    icon: 'pi pi-exclamation-triangle',
    acceptLabel: 'Sí',
    accept: () => {
      this._clientesService.eliminarCliente(id).then(() => {
        this.toastr.error('El cliente ' + cliente + ' ha sido eliminado', 'Cliente Eliminado', {
          positionClass: 'toast-bottom-right'
        });
      });
    }
  });
}

getClientes() {
  this._clientesService.getClientes().subscribe(data => {
    this.clientes = [];
    data.forEach(element: any) => {
      this.clientes.push({
        id: element.payload.doc.id,
        cedula: element.payload.doc.cedula,
        ...element.payload.doc.data()
      });
    });
  });
}
}
}

```

Anexo 2: Servicio “Clientes”

```
import { Injectable } from '@angular/core';
import { AngularFireStore, Transactable } from '@angular/fire/compat/firestore';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class ClientesService {

  constructor(private firestore: AngularFireStore) {}

  getClientes(): Observable<any> {
    return this.firestore.collection('clientes', ref => ref.orderBy('nombres', 'asc')).snapshotChanges();
  }

  agregarClientes(cliente: any): Promise<any> {
    return this.firestore.collection('clientes').add(cliente);
  }

  editarClientes(id: string, data: any) {
    return this.firestore.collection('clientes').doc(id).update(data);
  }

  eliminarClientes(id: string): Promise<any> {
    return this.firestore.collection('clientes').doc(id).delete();
  }

  getCliente(id: string): Observable<any> {
    return this.firestore.collection('clientes').doc(id).snapshotChanges();
  }
}
```

Anexo 3: Componente “Proveedores”

```
applyFilterGlobal($event: any, stringVal: any) {
  this.dt!.filterGlobal(($event.target as HTMLInputElement).value, stringVal);
}

openNew() {
  this.proveedorExistente = false;
  this.manejarProveedor = this.fb.group({
    ruc: ['', [Validators.required, Validators.maxLength(13)]],
    nombres: ['', Validators.required],
    direccion: ['', Validators.required],
    telefono: ['', Validators.required],
    ciudad: ['', Validators.required],
  });
  this.submitted = false;
  this.proveedoresDialog = true;
  this.editar = false;
}

hideDialog() {
  this.proveedoresDialog = false;
  this.submitted = false;
}
```

```

editarProveedor(id: string) {
  this.proveedorExistente = false;
  this._proveedoresService.getProveedor(id).subscribe((data) => {
    this.idProveedor = id;
    this.manejarProveedor = this.fb.group({
      ruc: [
        data.payload.data()['ruc'],
        [Validators.required, Validators.maxLength(13)],
      ],
      nombres: [data.payload.data()['nombres'], Validators.required],
      direccion: [data.payload.data()['direccion'], Validators.required],
      telefono: [data.payload.data()['telefono'], Validators.required],
      ciudad: [data.payload.data()['ciudad'], Validators.required],
    });
  });
  this.proveedoresDialog = true;
  this.editar = true;
}
guardarProveedor() {
  this.submitted = true;

  if (this.manejarProveedor.invalid) {
    return;
  }
  const proveedor: any = {
    ruc: this.manejarProveedor.value.ruc,
    nombres: this.manejarProveedor.value.nombres,
    direccion: this.manejarProveedor.value.direccion,
    telefono: this.manejarProveedor.value.telefono,
    ciudad: this.manejarProveedor.value.ciudad,
  };
  if (this.editar == false) {
    for (let item in this.proveedores) {
      if (this.proveedores[item].ruc == proveedor.ruc) {
        this.proveedorExistente = true;
        return;
      }
    }
  }
  this._proveedoresService
    .agregarProveedor(proveedor)
    .then(() => {
      this.proveedoresDialog = false;
      this.toastr.success(
        'El proveedor fue registrado con éxito!',
        'Proveedor Registrado',
        {
          positionClass: 'toast-bottom-right',
        }
      );
    });
}

```

```

eliminarProveedor(id: string, proveedor: string) {
  this.confirmationService.confirm({
    message: '¿Está seguro de eliminar a ' + proveedor + ' de la lista?',
    header: 'Atención',
    icon: 'pi pi-exclamation-triangle',
    acceptLabel: 'Si',
    accept: () => {
      this._proveedoresService.eliminarProveedor(id).then(() => {
        this.toastr.error(
          'El proveedor ' + proveedor + ' ha sido eliminado',
          'Proveedor Eliminado',
          {
            positionClass: 'toast-bottom-right',
          }
        );
      });
    }
  });
}

getProveedores() {
  this._proveedoresService.getProveedores().subscribe((data) => {
    this.proveedores = [];
    data.forEach((element: any) => {
      this.proveedores.push({
        id: element.payload.doc.id,
        ruc: element.payload.doc.ruc,
        ...element.payload.doc.data(),
      });
    });
  });
}
}
}

```

Anexo 4: Servicio “Proveedores”

```

import { Injectable } from '@angular/core';
import { AngularFireStore } from '@angular/fire/compat/firestore';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class ProveedoresService {

  constructor(private firestore: AngularFireStore) {}

  getProveedores(): Observable<any> {
    return this.firestore.collection('proveedores', ref => ref.orderBy('nombres', 'asc')).snapshotChanges();
  }

  agregarProveedor(proveedor: any): Promise<any> {
    return this.firestore.collection('proveedores').add(proveedor);
  }

  editarProveedor(id: string, data: any) {
    return this.firestore.collection('proveedores').doc(id).update(data);
  }

  eliminarProveedor(id: string): Promise<any> {
    return this.firestore.collection('proveedores').doc(id).delete();
  }

  getProveedor(id: string): Observable<any> {
    return this.firestore.collection('proveedores').doc(id).snapshotChanges();
  }
}

```

Anexo 5: Componente “Productos”

```
applyFilterGlobal($event: any, stringVal: any) {
  this.$t!.filterGlobal(($event.target as HTMLInputElement).value, stringVal);
}
openNew() {
  this.productoExistente = false;
  this.Hpvp = 0;
  this.Hcosto = 0;
  this.utilidad = 0;

  this.manejarProducto = this.fb.group({
    codigo: ['', [Validators.required, Validators.maxLength(13)]],
    descripcion: ['', Validators.required],
    pvp: ['', Validators.required],
    costo: ['', Validators.required],
    stock: ['', Validators.required],
    utilidad: ['']
  });
  this.submitted = false;
  this.productosDialog = true;
  this.editar = false;
}
hideDialog() {
  this.productosDialog = false;
  this.submitted = false;
}
```

```
editarProducto(id: string) {
  this.productoExistente = false;
  this._productosService.getProductos(id).subscribe(data => {
    this.idProducto = id;
    this.Hpvp = data.payload.data()['pvp'];
    this.Hcosto = data.payload.data()['costo'];
    this.utilidad = data.payload.data()['utilidad'];
    this.manejarProducto = this.fb.group({
      codigo: [data.payload.data()['codigo'], [Validators.required, Validators.maxLength(10)]],
      descripcion: [data.payload.data()['descripcion'], Validators.required],
      pvp: [data.payload.data()['pvp'], Validators.required],
      costo: [data.payload.data()['costo'], Validators.required],
      stock: [data.payload.data()['stock'], Validators.required],
      utilidad: [data.payload.data()['utilidad']]
    });
  });
  this.productosDialog = true;
  this.editar = true;
}
guardarProducto() {
  this.submitted = true;

  if (this.manejarProducto.invalid) {
    return;
  }
  const producto: any = {
    codigo: this.manejarProducto.value.codigo,
    descripcion: this.manejarProducto.value.descripcion,
    pvp: this.manejarProducto.value.pvp,
    costo: this.manejarProducto.value.costo,
    stock: this.manejarProducto.value.stock,
    utilidad: this.utilidad
  };
  if (this.editar == false) {
    for (let item in this.productos) {
      if (this.productos[item].codigo == producto.codigo) {
        this.productoExistente = true;
        return;
      }
    }
  }
  this._productosService.agregarProducto(producto).then(() => {
    this.productosDialog = false;
    this.toastr.success('El producto fue registrado con éxito!', 'Producto Registrado', {
      positionClass: 'toast-bottom-right'
    });
  }).catch(error => {
    this.toastr.error(this.fireError.firebaseError(error.code), 'Error', {

```

```

eliminarProducto(id: string, producto: string) {
  this.confirmationService.confirm({
    message: '¿Está seguro de eliminar ' + producto + ' del inventario?',
    header: 'Atención',
    icon: 'pi pi-exclamation-triangle',
    acceptLabel: 'Si',
    accept: () => {
      this._productosService.eliminarProducto(id).then(() => {
        this.toastr.error('El producto ' + producto + ' ha sido eliminado', 'Producto Eliminado', {
          positionClass: 'toast-bottom-right'
        });
      });
    }
  });
}

getProductos() {
  this._productosService.getProductos().subscribe(data => {
    this.productos = [];
    data.forEach(element: any) => {
      this.productos.push({
        id: element.payload.doc.id,
        codigo: element.payload.doc.codigo,
        ...element.payload.doc.data()
      });
    });
  });
}

calcularUtilidad(): number {
  this.utilidad = Math.round(((this.Hpvp - this.Hcosto) / (this.Hcosto)) * 100);
  return this.utilidad;
}
}

```

Anexo 6: Servicio “Productos”

```

export class ProductosService {

  constructor(private firestore: AngularFirestore) {}

  getProductos(): Observable<any> {
    return this.firestore.collection('productos', ref => ref.orderBy('descripcion', 'asc')).snapshotChanges();
  }

  agregarProducto(producto: any): Promise<any> {
    return this.firestore.collection('productos').add(producto);
  }

  editarProducto(id: string, data: any) {
    return this.firestore.collection('productos').doc(id).update(data);
  }

  eliminarProducto(id: string): Promise<any> {
    return this.firestore.collection('productos').doc(id).delete();
  }

  getProducto(id: string): Observable<any> {
    return this.firestore.collection('productos').doc(id).snapshotChanges();
  }
}

```

Anexo 7: Componente “Kárdex”

```
applyFilterGlobal($event: any, stringVal: any) {
  this.dtl.filterGlobal(($event.target as HTMLInputElement).value, stringVal);
}
pasarProveedor(value: any){
  this.manejarKardex.value.proveedor = value.nombres.toString();
}
guardarKardex(){
  this.submitted = true;
  if (this.manejarKardex.invalid || this.balNegativo) {
    return;
  }
  if(this.recibir == false){
    this.manejarKardex.value.cantidad = -this.manejarKardex.value.cantidad;
  }
  const kard: any = {
    proveedor: this.manejarKardex.value.proveedor,
    producto: this.manejarKardex.value.producto,
    fecha: this.manejarKardex.value.fecha,
    referencia: this.manejarKardex.value.referencia,
    cantidad: this.manejarKardex.value.cantidad,
    costo: this.manejarKardex.value.costo,
    balance: this.bal
  }
  if(this.recibir){
    this._kardexService.recibir(this.idProducto, kard.balance);
  }else{
    this._kardexService.retirar(this.idProducto, kard.balance);
  }
  this.kardexDialog = false;
  this._kardexService.guardarKardex(kard).then(() => {
    this.toastr.info('El stock fue modificado con éxito!', 'Kárdex Actualizado', {
      positionClass: 'toast-bottom-right'
    });
  }).catch(error => {
    this.toastr.info(this.firebError.firebaseError(error.code), 'Error', {
      positionClass: 'toast-bottom-right'
    });
  }).catch(error =>{
    this.toastr.info(this.firebError.firebaseError(error.code), 'Error', {
      positionClass: 'toast-bottom-right'
    });
  })
}
```

```

recibirProducto(id: string){
  this.recibir = true;
  this.encabezadoDialog = "Recibir Unidades";
  this.bal = 0;
  this._kardexService.getProducto(id).subscribe(data => {
    this.idProducto = id;
    this.stockProd = data.payload.data()['stock'];
    this.manejarKardex = this.fb.group({
      proveedor: ['', Validators.required],
      producto: [data.payload.data()['codigo'], Validators.required],
      fecha: ['', Validators.required],
      referencia: ['', Validators.required],
      cantidad: ['', Validators.required],
      costo: [data.payload.data()['costo'], Validators.required],
      balance: [''],
    })
  })
  this.kardexDialog = true;
}

retirarProducto(id: string){
  this.recibir = false;
  this.encabezadoDialog = "Retirar Unidades";
  this.bal = 0;
  this._kardexService.getProducto(id).subscribe(data => {
    this.idProducto = id;
    this.stockProd = data.payload.data()['stock'];
    this.manejarKardex = this.fb.group({
      proveedor: ['', Validators.required],
      producto: [data.payload.data()['codigo'], Validators.required],
      fecha: [new Date(), Validators.required],
      referencia: ['', Validators.required],
      cantidad: ['', Validators.required],
      costo: [data.payload.data()['costo'], Validators.required],
      balance: [''],
    })
  })
  this.kardexDialog = true;
}

hideDialog() {
  this.kardexDialog = false;
  this.submitted = false;
}

openReportes(prod: string){
  this.kardexRepDialog = true;
  this.idProducto = prod;
  this.getKardex(prod);
}

```

```

hideReportes(){
  this.kardexRepDialog = false;
}
getProductos() {
  this._kardexService.getProductos().subscribe(data => {
    this.productos = [];
    data.forEach((element: any) => {
      this.productos.push({
        id: element.payload.doc.id,
        codigo: element.payload.doc.codigo,
        ...element.payload.doc.data()
      })
    });
  })
}
getProveedores(){
  this._provService.getProveedores().subscribe(data => {
    this.proveedores = [];
    data.forEach((element: any) => {
      this.proveedores.push({
        id: element.payload.doc.id,
        nombres: element.payload.doc.nombres,
        ...element.payload.doc.data()
      })
    });
  })
}
getKardex(prod: string){
  this._kardexService.getKardex(prod).subscribe(data => {
    this.kardex = [];
    data.forEach((element: any) => {
      this.kardex.push({
        id: element.payload.doc.id,
        ...element.payload.doc.data()
      })
    });
  })
}
calcularBalance(): number{
  this.balNegativo = false;
  if(this.recibir){
    this.bal = Number(this.stockProd) + this.manejarKardex.value.cantidad;
  }else{
    this.bal = Number(this.stockProd) - this.manejarKardex.value.cantidad;
  }
  if(this.bal < 0){
    this.balNegativo = true;
  }
}

```

Anexo 8: Servicio “Kárdex”

```
import { Injectable } from '@angular/core';
import { AngularFireStore } from '@angular/fire/compat/firestore';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class KárdexSService {

  constructor(private firestore: AngularFireStore) {
  }

  getProductos(): Observable<any> {
    return this.firestore.collection('productos', ref => ref.orderBy('descripcion', 'asc')).snapshotChanges();
  }

  getProducto(id: string): Observable<any> {
    return this.firestore.collection('productos').doc(id).snapshotChanges();
  }

  guardarKárdex(kárdex: any): Promise<any> {
    return this.firestore.collection('kárdex').add(kárdex);
  }

  recibir(id: string, balance: number) {
    return this.firestore.collection('productos').doc(id).update({
      stock: balance
    });
  }

  retirar(id: string, balance: number) {
    return this.firestore.collection('productos').doc(id).update({
      stock: balance
    });
  }

  getKárdex(producto: string): Observable<any> {
    return this.firestore.collection('kárdex', ref => ref.where(
      ('producto', '==', producto).orderBy(
        ('fecha', 'asc')).snapshotChanges();
    )
  }
}
```

Anexo 9: Componente “Facturación”

```
filterClientes(event: any) {
  let filtered: any[] = [];
  let query = event.query;

  for (let i = 0; i < this.clientes.length; i++) {
    let nom = this.clientes[i];
    if (nom.nombres.toLowerCase().indexOf(query.toLowerCase()) == 0) {
      filtered.push(nom);
    }
  }
  this.filtroClientes = filtered;
}
agregarProd() {
  this.aggNuevo = true;
}
openDialog(nuevo: boolean) {
  if (nuevo == true) {
    this.aggNuevo = true;
    this.manejarVenta = this.fb.group({
      cliente: ['', Validators.required],
      fecha: [new Date(Date.now()), Validators.required],
      subtot: ['', Validators.required],
      impuesto: ['12'],
      total: ['', Validators.required],
      detalle: [''],
    });
    this.ventaDialog = true;
  } else {
    this.aggNuevo = false;
    this.ventaDialog = true;
  }
}
openRepDialog() {
  this.getVentas();
  this.repDialog = true;
}
hideDialog() {
  if (this.aggNuevo == true) {
    this.ventaDialog = false;
    this.productosSeleccionados = [];
    this.aggNuevo = false;
  } else {
    this.ventaDialog = false;
    this.aggNuevo = false;
  }
}
```

```

pasarSubyTot() {
  this.subt = 0;
  this.tot = 0;
  for (let i = 0; i < this.productosSeleccionados.length; i++) {
    this.subt +=
      Number(this.productosSeleccionados[i].pvp) *
      Number(this.productosSeleccionados[i].cantidad);
    this.tot = this.subt + Number(this.subt * (12 / 100));
  }
}
pasarCantidad() {
  for (let i = 0; i < this.productosSeleccionados.length; i++) {
    this.productosSeleccionados[i].cantidad = 1;
  }
}
pasarProductos() {
  this.sinStock = false;

  if (this.productosSeleccionados.length == 0) {
    return;
  }
  this.pasarCantidad();
  this.pasarSubyTot();
  this.pasarProd = true;
  if (this.manejarVenta.invalid) {
    return;
  }
  for (let item in this.productosSeleccionados) {
    if (this.productosSeleccionados[item].stock < 1) {
      this.sinStock = true;
    }
  }
  if (this.sinStock) {
    return;
  }
  this.ventaDialog = false;
  this.clienteSeleccionado = this.manejarVenta.value.cliente.nombres;
}

```

```

this._facturacionService
  .guardarVenta(venta)
  .then(() => {
    this.actualizarKardex();
    this.productosSeleccionados = [];
    this.toastr.info('Venta guardada con éxito!', 'Factura Emitida', {
      positionClass: 'toast-bottom-right',
    });
  })
  .catch((error) => {
    this.toastr.error(this.firebError.firebaseError(error.code), 'Error', {
      positionClass: 'toast-bottom-right',
    });
  });
this._facturacionService.guardarComprobante(comprobante);
}

```

```

secuencial(): string {
    var sec = '00000000';
    if (this.ultimaVenta[0].numero >= 99) {
        sec = '000000';
    }
    if (this.ultimaVenta[0].numero >= 9) {
        sec = '000000';
    } else {
        sec = '00000000';
    }
    return sec + (this.ultimaVenta[0].numero + 1);
}
generarClaveAcceso(): string {
    var clave =
        new DatePipe('en-US').transform(Date.now(), 'ddMMyyyy') +
        '01' +
        '0301450631001' +
        '1' +
        '001100' +
        this.secuencial() +
        '12345678' +
        '1';
    return clave;
}
modulo11(): string {
    var suma: number = 0;
    var factor: number = 2;
    var clave = [...this.generarClaveAcceso()];
    for (let i = clave.length - 1; i >= 0; i--) {
        suma += Number(clave[i]) * factor;
        if (factor == 7) {
            factor = 2;
        } else {
            factor++;
        }
    }
    var digitoVerificador: number = suma % 11;
    digitoVerificador = 11 - digitoVerificador;
    if (digitoVerificador == 11) {
        digitoVerificador = 0;
    }
    if (digitoVerificador == 10) {
        digitoVerificador = 1;
    }
    return this.generarClaveAcceso() + digitoVerificador;
}

```

```

descargarXml(response: any, fileName: string): void {
  const dataType = response.type;
  const binaryData: any[] = [];
  binaryData.push(response);

  const filePath = window.URL.createObjectURL(
    new Blob(binaryData, { type: dataType })
  );
  const downloadLink = document.createElement('a');
  downloadLink.href = filePath;
  downloadLink.setAttribute('download', fileName);
  document.body.appendChild(downloadLink);
  downloadLink.click();
}

quitarProducto(codProd: string) {
  for (let i = 0; i < this.productosSeleccionados.length; i++) {
    if (codProd == this.productosSeleccionados[i].codigo) {
      this.productos.push(this.productosSeleccionados[i]);
      this.productosSeleccionados.splice(i, 1);
    }
  }
  this.pasarSubyTot();
}

actualizarKardex() {
  for (let i = 0; i < this.productosSeleccionados.length; i++) {
    const kard: any = {
      producto: this.productosSeleccionados[i].codigo,
      fecha: this.manejarVenta.value.fecha,
      referencia: 'Venta de Productos',
      cantidad: -this.productosSeleccionados[i].cantidad,
      costo: this.productosSeleccionados[i].pvp,
      balance:
        this.productosSeleccionados[i].stock -
        this.productosSeleccionados[i].cantidad,
    };
    this._kardexService.guardarKardex(kard);
    this._kardexService.retirar(
      this.productosSeleccionados[i].id,
      kard.balance
    );
  }
}

```

```

getClientes() {
  this._clientesService.getClientes().subscribe((data) => {
    this.clientes = [];
    data.forEach((element: any) => {
      this.clientes.push({
        id: element.payload.doc.id,
        cedula: element.payload.doc.cedula,
        nombres: element.payload.doc.nombres,
        ...element.payload.doc.data(),
      });
    });
  });
}

getProductos() {
  this._productosService.getProductos().subscribe((data) => {
    this.productos = [];
    data.forEach((element: any) => {
      this.productos.push({
        id: element.payload.doc.id,
        codigo: element.payload.doc.codigo,
        ...element.payload.doc.data(),
      });
    });
  });
}

getVentas() {
  this._facturacionService.getVentas().subscribe((data) => {
    this.ventas = [];
    data.forEach((element: any) => {
      this.ventas.push({
        id: element.payload.doc.id,
        detalle: element.payload.doc.detalle,
        ...element.payload.doc.data(),
      });
    });
  });
}

getUltimaVenta() {
  this._facturacionService.getUltimaVenta().subscribe((data) => {
    this.ultimaVenta = [];
    data.forEach((element: any) => {
      this.ultimaVenta.push({
        ...element.payload.doc.data(),
      });
    });
  });
}

```

Anexo 10: Servicio “Facturación”

```
import { Injectable } from '@angular/core';
import { AngularFireStore } from '@angular/fire/compat/firestore';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class FacturacionService {

  constructor(private firestore: AngularFireStore) {}

  guardarVenta(venta: any): Promise<any> {
    return this.firestore.collection('ventas').add(venta);
  }

  getVentas(): Observable<any> {
    return this.firestore.collection('ventas', ref => ref.orderBy(
      'fecha', 'asc')).snapshotChanges();
  }

  getUltimaVenta(): Observable<any> {
    return this.firestore.collection('ventas', ref => ref.orderBy(
      'numero', 'desc').limit(1)).snapshotChanges();
  }

  guardarComprobante(comprobante: any): Promise<any> {
    return this.firestore.collection('comprobantes').doc(comprobante.ClaveAcceso).set(comprobante);
  }
}
```

Anexo 11: Componente “Main”

```
hoy() {
  this.startDate = new Date(Date.now());
  this.endDate = new Date(Date.now());

  this.label = 'Hoy';

  this.startDate.setHours(0);
  this.startDate.setMinutes(0);
  this.startDate.setSeconds(0);

  this.endDate.setHours(23);
  this.endDate.setMinutes(59);
  this.endDate.setSeconds(59);

  this.getVentasFecha(this.startDate, this.endDate);
  this.getTopProdFecha(this.startDate, this.endDate);
}
mesActual() {
  this.startDate = new Date();
  this.startDate.setDate(1);
  this.endDate = new Date(Date.now());
  this.label = 'Mes Actual';

  this.startDate.setHours(0);
  this.startDate.setMinutes(0);
  this.startDate.setSeconds(0);

  this.endDate.setHours(23);
  this.endDate.setMinutes(59);
  this.endDate.setSeconds(59);

  this.getVentasFecha(this.startDate, this.endDate);
  this.getTopProdFecha(this.startDate, this.endDate);
}
mesAnterior() {
  this.startDate = new Date();
  this.endDate = new Date();

  this.startDate.setDate(1);
  this.startDate.setMonth(this.startDate.getMonth() - 1);

  this.endDate.setDate(1);
  this.endDate.setDate(this.endDate.getDate() - 1);

  this.label = 'Mes Anterior';

  this.startDate.setHours(0);
```

```

getClientes() {
  this._dashboardService.getClientes().subscribe((data) => {
    this.clientes = [];
    data.forEach((element: any) => {
      this.clientes.push({
        ...element.payload.doc.data(),
      });
    });
    this.cliTotal = this.clientes.length;
  });
}
getProductos() {
  this._dashboardService.getProductos().subscribe((data) => {
    this.productos = [];
    data.forEach((element: any) => {
      this.productos.push({
        ...element.payload.doc.data(),
      });
    });
    this.prodTotal = this.productos.length;
  });
}
getProveedores() {
  this._dashboardService.getProveedores().subscribe((data) => {
    this.proveedores = [];
    data.forEach((element: any) => {
      this.proveedores.push({
        ...element.payload.doc.data(),
      });
    });
    this.provTotal = this.proveedores.length;
  });
}
prodBajoStock() {
  this._dashboardService.prodBajoStock().subscribe((data) => {
    this.prodStock = [];
    data.forEach((element: any) => {
      this.prodStock.push({
        ...element.payload.doc.data(),
      });
    });
  });
}
}

```

```
getVentasFecha(inicio: Date, fin: Date) {
  this._dashboardService.ventasPorFecha(inicio, fin).subscribe((data) => {
    this.ventasList = [];
    data.forEach((element: any) => {
      this.ventasList.push({
        ..element.payload.doc.data(),
      });
    });
    var totVentas: any[] = [];
    this.labelTotal = 0;
    for (let i = 0; i < this.ventasList.length; i++) {
      totVentas.push({
        total: this.ventasList[i].total,
        fecha: new Date(
          this.ventasList[i].fecha.seconds * 1000
        ).toLocaleDateString(),
      });
      this.labelTotal += this.ventasList[i].total;
    }
    this.ventasData = {
      labels: totVentas.map((totVentas) => totVentas.fecha),
      datasets: [
        {
          label: 'Total',
          backgroundColor: '#036cb2',
          data: totVentas.map((totVentas) => totVentas.total),
        },
      ],
    };
  });
}
```

```

getTopProdFecha(inicio: Date, fin: Date) {
  this._dashboardService
    .topProductosPorFecha(inicio, fin)
    .subscribe((data) => {
      this.topProdList = [];
      data.forEach((element: any) => {
        this.topProdList.push({
          ...element.payload.doc.data(),
        });
      });
      let agrupado: any[] = [];
      let topP: any[] = [];
      let topCinco: any[] = [];
      for (let i = 0; i < this.topProdList.length; i++) {
        for (let j = 0; j < this.topProdList[i].detalle.length; j++) {
          if (!agrupado[this.topProdList[i].detalle[j].descripcion])
            agrupado[this.topProdList[i].detalle[j].descripcion] = [];
          agrupado[this.topProdList[i].detalle[j].descripcion].push({
            descripcion: this.topProdList[i].detalle[j].descripcion,
            cantidad: this.topProdList[i].detalle[j].cantidad,
          });
        }
      }
      var desc = '';
      var cant = 0;
      for (let item in agrupado) {
        for (let i in agrupado[item]) {
          desc = agrupado[item][i].descripcion;
          cant += agrupado[item][i].cantidad;
        }
        topP.push({
          descripcion: desc,
          cantidad: cant,
        });
        cant = 0;
      }
      topP.sort((a, b) => b.cantidad - a.cantidad);

      if (topP.length < 5) {
        for (let i = 0; i < topP.length; i++) {
          topCinco.push(topP[i]);
        }
      } else {
        for (let i = 0; i < 5; i++) {
          topCinco.push(topP[i]);
        }
      }
    });
}

```

Anexo 12: Servicio “Dashboard”

```
import { Injectable } from '@angular/core';
import { AngularFireStore } from '@angular/fire/compat/firestore';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class DashboardService {

  constructor(private firestore: AngularFireStore) { }

  getClientes(): Observable<any> {
    return this.firestore.collection('clientes').snapshotChanges();
  }

  getProductos(): Observable<any> {
    return this.firestore.collection('productos').snapshotChanges();
  }

  getProveedores(): Observable<any> {
    return this.firestore.collection('proveedores').snapshotChanges();
  }

  prodBajoStock(): Observable<any> {
    return this.firestore.collection('productos', ref => ref.where(
      {'stock', '<', 5}).orderBy(
        {'stock', 'desc'}).snapshotChanges();
  }

  ventasPorFecha(startDate: Date, endDate: Date): Observable<any> {
    return this.firestore.collection('ventas', ref => ref.where(
      {'fecha', '>=', startDate}.where({'fecha', '<=', endDate}).orderBy({'fecha'})).snapshotChanges();
  }

  topProductosPorFecha(startDate: Date, endDate: Date): Observable<any> {
    return this.firestore.collection('ventas', ref => ref.where(
      {'fecha', '>=', startDate}.where({'fecha', '<=', endDate}).orderBy({'fecha'})).snapshotChanges();
  }
}
```

Anexo 13: Estructura XML

```
const { create } = require('xmlbuilder2');

var root = create({ version: '1.0', encoding: 'utf-8' })
  .ele('factura', { id: 'comprobante', version: '1.1.0' })
  .ele('infoTributaria')
  .ele('ambiente')
  .txt('1')
  .up()
  .ele('tipoMision')
  .txt('1')
  .up()
  .ele('razonSocial')
  .txt(this.infoEmpresa[0].razonSocial)
  .up()
  .ele('nombreComercial')
  .txt(this.infoEmpresa[0].nombreComercial)
  .up()
  .ele('ruc')
  .txt(this.infoEmpresa[0].ruc)
  .up()
  .ele('claveAcceso')
  .txt(this.modulo1())
  .up()
  .ele('codDoc')
  .txt('01')
  .up()
  .ele('estab')
  .txt('801')
  .up()
  .ele('ptoEmi')
  .txt('100')
  .up()
  .ele('secuencial')
  .txt(this.secuencial())
  .up()
  .ele('dirMatriz')
  .txt(this.infoEmpresa[0].dirMatriz)
  .up()
  .up()
```

```
.ele('introduccion')
.ele('fechaEmision')
.txt(new DatePipe('en-US').transform(venta.fecha, 'dd/MM/yyyy'))
.up()
.ele('dirEstablecimiento')
.txt(this.infoEmpresa[0].dirMatr1a)
.up()
.ele('obligadoContabilidad')
.txt('NO')
.up()
.ele('tipoIdentificacionComprador')
.txt('05')
.up()
.ele('razonSocialComprador')
.txt(venta.cliente.nombre)
.up()
.ele('identificacionComprador')
.txt(venta.cliente.cedula)
.up()
.ele('direccionComprador')
.txt(venta.cliente.direccion)
.up()
.ele('totalSinImpuestos')
.txt(
  new CurrencyPipe('en-US')
    .transform(venta.subtotal, '')
    ?.replace(",","")
)
.up()
.ele('totalDescuento')
.txt('0.00')
.up()
.ele('totalConImpuestos')
.ele('totalImpuesto')
.ele('codigo')
.txt('2')
.up()
.ele('codigoPorcentaje')
.txt('0')
```

```
.up()
.ele('baseImponible')
.txt(
  new CurrencyPipe('en-US')
    .transform(venta.subtotal, ' ')
    ?.replace(',',' ')
)
.up()
.ele('valor')
.txt('0.00')
.up()
.up()
.ele('totalImpuesto')
.ele('codigo')
.txt('2')
.up()
.ele('codigoPorcentaje')
.txt('2')
.up()
.ele('baseImponible')
.txt('0.00')
.up()
.ele('valor')
.txt('0.00')
.up()
.up()
.up()
.ele('propina')
.txt('0.00')
.up()
.ele('importeTotal')
.txt(
  new CurrencyPipe('en-US')
    .transform(venta.subtotal, ' ')
    ?.replace(',',' ')
)
.up()
.ele('moneda')
.txt('DOLAR')
```

```

.up()
.ele('pagos')
.ele('pago')
.ele('formaPago')
.txt('01')
.up()
.ele('total')
.txt(
  new CurrencyPipe('en-US')
    .transform(venta.subtotal, ' ')
    ?.replace(',',' ')
)
.up()
.up()
.up()
.up()
.ele('detalles');
for (let i in venta.detalle) {
  root
  .ele('detalle')
  .ele('codigoPrincipal')
  .txt(venta.detalle[i].codigo)
  .up()
  .ele('descripcion')
  .txt(venta.detalle[i].descripcion)
  .up()
  .ele('cantidad')
  .txt(venta.detalle[i].cantidad)
  .up()
  .ele('precioUnitario')
  .txt(
    new CurrencyPipe('en-US')
      .transform(venta.detalle[i].pvp, ' ')
      ?.replace(',',' ')
  )
  .up()
  .ele('descuento')
  .txt('0.00')
  .up()

```

```

    .txt(
      new CurrencyPipe('en-US')
        .transform(venta.detalle[i].pvp * venta.detalle[i].cantidad, ' ')
        ?.replace(',',' ')
    )
  .up()
  .ele('impuestos')
  .ele('impuesto')
  .ele('codigo')
  .txt('2')
  .up()
  .ele('codigoPorcentaje')
  .txt('0')
  .up()
  .ele('tarifa')
  .txt('0.00')
  .up()
  .ele('baseImponible')
  .txt(
    new CurrencyPipe('en-US')
      .transform(venta.detalle[i].pvp * venta.detalle[i].cantidad, ' ')
      ?.replace(',',' ')
  )
  .up()
  .ele('valor')
  .txt('0.00')
  .up();
}
root
.up()
.ele('infoAdicional')
.ele('campoAdicional', { nombre: 'Dirección cliente' })
.txt(venta.cliente.direccion)
.up()
.ele('campoAdicional', { nombre: 'Email' })
.txt(venta.cliente.email)
.up();

```

Anexo 14: Roles de usuario

```
ngOnInit(): void {
  this.afAuth.authState.subscribe((user =>{
    if (user) {
      this.dataUser = user;
      this.getUsuario(this.dataUser.uid);
    } else {
      this.router.navigate(['/login']);
    }
  }));
}

getUsuario(id: string) {
  this._usuariosService.getUsuario(id).subscribe((data) => {
    this.usuario = [];
    this.usuario.push(data.payload.data());
    this.administrador = this.usuario[0].administrador;
  });
}
```

Anexo 15: Emisión del comprobante electrónico

**SERVCOMP -
Servicios
Computarizados**

PACHECO NARVAEZ SILVIO ENRIQUE

**Estimad@ Silvio David
Pacheco Herrera**

Adjunto sírvase encontrar el documento electrónico
FACTURA Nro. davidph5469@gmail.com emitido
por el contribuyente **PACHECO NARVAEZ SILVIO
ENRIQUE**

Ambiente	PRUEBAS
Clave de	101120220103014506310011001100
Acceso:	0000000361234567810
Nro.	101120220103014506310011001100
Autorización:	0000000361234567810


Acceso: 0000000361234567810
Nro. 101120220103014506310011001100
Autorización: 0000000361234567810
Fecha 10/11/2022 11:46:59
Autorización:



10112022010301
4506310011001...

 PDF

10112022010301
4506310011001...

 xml

← Responder

→ Reenviar

Silvio David Pacheco Herrera portador de la cédula de ciudadanía N.º **0150041572**. En calidad de autor y titular de los derechos patrimoniales del trabajo de titulación “**Desarrollo de un Prototipo de Software para la Gestión de Inventarios y Facturación para Empresas del Sector de Ventas y Servicios de Computación con Almacenamiento en la Nube - Caso de Aplicación**” de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos y no comerciales. Autorizo además a la Universidad Católica de Cuenca, para que realice la publicación de este trabajo de titulación en el Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Azogues, **24 de enero de 2023**

F: 

Silvio David Pacheco Herrera

C.I. 0150041572