



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA DE COMPUTACIÓN

**CONSTRUCCIÓN DE UNA APLICACIÓN WEB DE FACTURACIÓN ELECTRÓNICA
PARA TIENDAS VIRTUALES IMPLEMENTADAS CON WOOCOMMERCE EN
ECUADOR**

Trabajo de titulación previo a la obtención del

Título de Ingeniera en Ciencias de la Computación

AUTORA: KARLA VANESSA MOYÓN RIVERA

TUTOR: RODRIGO EFRAÍN TUFÍÑO CÁRDENAS

Quito – Ecuador

2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, Karla Vanessa Moyón Rivera con documento de identificación N° 1724997950 manifiesto que:

Soy la autora responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 11 de marzo del año 2022

Atentamente,



Karla Vanessa Moyón Rivera

1724997950

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Karla Vanessa Moyón Rivera con documento de identificación No. 1724997950, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autora del Proyecto Técnico: “Construcción de una aplicación web de facturación electrónica para tiendas virtuales implementadas con WooCommerce en Ecuador”, el cual ha sido desarrollado para optar por el título de: Ingeniera en Ciencias de la Computación, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana

Quito, 11 de marzo del año 2022

Atentamente,



Karla Vanessa Moyón Rivera

1724997950

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Rodrigo Efraín Tufiño Cárdenas con documento de identificación N° 1717646390, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: CONSTRUCCIÓN DE UNA APLICACIÓN WEB DE FACTURACIÓN ELECTRÓNICA PARA TIENDAS VIRTUALES IMPLEMENTADAS CON WOOCOMMERCE EN ECUADOR, realizado por Karla Vanessa Moyón Rivera con documento de identificación N° 1724997950, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 11 de marzo del año 2022

Atentamente,



Ing. Rodrigo Efraín Tufiño Cárdenas, MSc.

1717646390

AGRADECIMIENTO

En primer lugar, agradezco a Dios por darme salud y una vida llena de aprendizaje, experiencias y felicidad junto a toda a mi familia y amigos. Ser lleno de luz y amor que ilumina mi camino y me ayuda a culminar con éxito cada meta planteada.

A mi madre, Laura Rivera, y mi hermano, Luis Moyón, por brindarme su apoyo incondicional y estar siempre a mi lado en cada paso que doy. Por sus consejos, los cuales me han ayudado a crecer y tomar las decisiones correctas a lo largo de mi vida.

A mi tutor, Rodrigo Tufiño, por el tiempo dedicado y compartir conmigo sus sabios conocimientos. Principalmente por confiar en mí y aceptar dirigir el presente proyecto.

Por último, a toda mi familia y amigos cercanos por darme ánimo durante este proceso.

DEDICATORIA

Dedico con todo mi corazón mi trabajo de titulación a mi madre, Laura Rivera, pues sin ella no lo habría logrado. Muchos de mis logros, incluyendo el presente trabajo, es gracias a ti, por tu amor y apoyo incondicional.

Tu bendición a lo largo de mi vida me protege y me lleva por el camino del bien, convirtiéndome en la mujer que soy.

ÍNDICE DE CONTENIDO

1. REVISIÓN DE LA LITERATURA O FUNDAMENTOS TEÓRICOS	19
1.1. Facturación electrónica.....	19
1.1.1. Consideraciones para la emisión de la factura electrónica.....	19
1.1.2. Emisión de comprobantes electrónicos	20
1.1.3. Requisitos para emitir facturas electrónicas	21
1.1.4. Ventajas y desventajas de la facturación electrónica	22
1.1.5. Proceso de Facturación electrónica	23
1.1.6. XML.....	24
1.2. Firma electrónica	25
1.2.1. Certificados digitales.....	26
1.2.2. Entidades Certificadoras en Ecuador	26
1.2.3. Tipos de firma	27
1.2.4. Funcionamiento de la firma electrónica	28
1.2.5. Estándar XadES_BES	29
1.3. Desarrollo de aplicaciones Web.....	30
1.3.1. WordPress	31
1.3.2. WooCommerce.....	31
1.3.3. Integración de aplicaciones	32
2. MARCO METODOLÓGICO	34
2.1. Eventos	34
2.2. Roles	35
2.3. Elementos	36
3. RESULTADOS	40
3.1. Análisis	40
3.1.1. Análisis del proceso de facturación electrónica	40
3.1.2. Análisis de Viabilidad	50
3.1.2.1. Viabilidad técnica.....	51
3.1.2.2. Viabilidad económica.....	52
3.1.2.3. Viabilidad operacional.....	53
3.1.3. Análisis de Requerimientos.....	54
3.1.3.1. Alcance.....	54
3.1.3.2. Definiciones, acrónimos y abreviaturas.....	55
3.1.3.3. Descripción general.....	56
3.1.3.4. Requerimientos específicos	59

3.1.3.5.	Requerimientos Funcionales	60
3.1.3.6.	Requerimientos No Funcionales.....	65
3.1.3.7.	Restricciones de diseño	65
3.2.	Diseño	66
3.2.2.	Casos de uso	70
3.2.2.1.	Caso de uso 01: Inicio de sesión.....	70
3.2.2.2.	Caso de uso 02: Menú	71
3.2.2.3.	Caso de uso 03: Mis facturas.....	72
3.2.3.	Diagrama de secuencia	73
3.2.3.1.	Autenticación de usuario	73
3.2.3.2.	Visualizar datos generales para la facturación	74
3.2.3.3.	Actualización de pedidos.....	75
3.2.3.4.	Visualizar los datos de la factura electrónica	76
3.2.3.5.	Búsquedas para filtrar pedidos	77
3.2.3.6.	Facturar pedido.....	78
3.2.3.7.	Generar reporte general	79
3.2.3.8.	Consultar datos usuario	80
3.2.4.	Diagrama de base de datos	81
3.2.5.	Diagrama de clases.....	87
3.3.	Construcción.....	90
3.3.1.	Configuraciones Principales.....	90
3.3.2.	Sprint 1	91
3.3.3.	Sprint 2	94
3.3.4.	Sprint 3	99
3.3.5.	Sprint 4	104
3.3.6.	Sprint 5	111
3.4.	Pruebas de funcionamiento.....	119
4.	CONCLUSIONES.....	138
5.	RECOMENDACIONES.....	139
6.	BIBLIOGRAFÍA.....	140
7.	ANEXOS.....	143
	ANEXO 1: SOLICITUD DE EMISIÓN PARA DOCUMENTOS ELECTRÓNICOS PARA PRUEBAS	143
	ANEXO 2: CÓDIGO FUENTE DEL PROYECTO EN REPOSITORIO EN GITHUB... ..	144

Resumen

El sistema desarrollado dentro de este proyecto es un prototipo de una aplicación Web que brinda el servicio de facturación electrónica de pedidos realizados a una tienda virtual implementada con WooCommerce en Ecuador, cumpliendo con especificaciones técnicas, operativas y administrativas determinadas por el Servicio de Rentas Internas (SRI) en el año 2021. Cabe mencionar, que el sistema de manera adicional muestra un reporte general con datos de los pedidos de la tienda virtual como el total en ventas, productos más vendidos, total de clientes, consumo promedio y el total IVA de ventas.

A lo largo del documento se presenta la construcción del sistema, llamado KV - Facturación, iniciando por las fases de análisis, donde se menciona la viabilidad: técnica, económica y operacional, además de sus requerimientos. Para luego continuar con el diseño mediante diagramas UML y prototipo de interfaces para la construcción de la aplicación utilizando metodología ágil de desarrollo Scrum. Finalmente se realiza pruebas de funcionamiento con conexión a una tienda virtual activa. Cabe mencionar que las herramientas y plataformas utilizadas en el desarrollo del presente proyecto son de Software Libre.

- **Palabras claves**

Sistema de facturación, aplicación web, firma electrónica, WooCommerce, WordPress, metodología Scrum, tienda virtual, comercio electrónico.

Abstract

The system developed within this project is a prototype of a Web application that provides the electronic invoicing service for orders made to a virtual store implemented with WooCommerce in Ecuador, complying with technical, operational and administrative specifications determined by the Internal Revenue Service (SRI). It is worth mentioning that the system additionally shows a general report with data on virtual store orders such as total sales, best-selling products, total customers, average consumption and total taxes.

Throughout the document, the construction of the system, called KV - Facturación, is presented, starting with the analysis phases, where the viability is mentioned: technical, economic and operational, in addition to its requirements. To then continue with the design using UML diagrams and prototype interfaces for building the application using agile Scrum development methodology. Finally, performance tests are performed with connection to an active virtual store. It is worth mentioning that the tools and platforms used in the development of this project are Free Software

- **Keys Words**

Billing system, web application, electronic signature, WooCommerce, WordPress, Scrum methodology, virtual store, electronic commerce

INTRODUCCIÓN

En la actualidad, el comercio electrónico se ha convertido en una herramienta muy importante para los negocios a nivel mundial, debido a la apertura y facilidad de acceso al Internet, teniendo como ventajas la expansión del mercado globalizado y la rapidez con la que se manejan. Por otro lado, las relaciones interpersonales con los clientes pueden causar complicaciones en los negocios, sin embargo, el uso de las nuevas tecnologías y su innovación han creado mejoras en la comunicación con el cliente y la empresa.

Para el planteamiento del presente tema, se analizó que existen plataformas que ya cuentan con el servicio de facturación electrónica como es el caso de Datil, una plataforma robusta, pero con planes que varían dependiendo de las características que necesite cada negocio y el número de transacciones que se realicen mensualmente o anualmente (Datil, 2021). Además, existe plugins integrados en la tienda de woocommerce con Datil, su instalación se necesita de conocimientos de desarrollo de software y además tiene un costo adicional, sin mencionar que de igual manera se considera la cantidad de comprobantes emitidos. Con el presente proyecto se quiere diferenciar de esta plataforma de manera que sea más dinámica y fácil de usar considerando al comerciante y consumidor que busca agilizar el proceso de facturación. Sintetizando, el proyecto proporciona una integración directa con WooCommerce, haciendo que sea una aplicación más integral ya que al haber menos capas de software entre la tienda virtual y su respectiva facturación, es mejor.

Por otra parte, la facturación electrónica se está volviendo indispensable, debido a que ayuda a tener un mejor control de las operaciones de los contribuyentes. A su vez disminuye los errores y los costos operativos, a través de la automatización y optimización de los procesos administrativos y contables. Además, de acuerdo a las estadísticas de emisión registradas en el SRI ayuda a salvar alrededor de 200.000 árboles diariamente. (SRI, Estadística de emisión de comprobantes electrónicos, 2021)

El comercio electrónico, también conocido como E-Commerce, ha demostrado ser un canal de ventas exitoso y efectivo ya que mejora sustancialmente los beneficios de las empresas, como por ejemplo la campaña de la empresa de supermercados Tesco y sus catálogos virtuales con códigos QR, que supuso un aumento del 76% en las ventas online. (Camarillo, 2008).

Por este motivo, se realiza la propuesta del desarrollo de la plataforma web para brindar el servicio de facturación electrónica. Además, se lo considerará como un proyecto de software libre para que la aplicación esté abierta a todo el mundo que desee estudiar, compartir, ejecutar y mejorar la plataforma. De esta manera, la aplicación estará abierta a todo el mundo, cooperando con la comunidad.

PROBLEMA

La competitividad que existe en el mundo globalizado y la rapidez para concretar negocios han impulsado el desarrollo del comercio electrónico, modificando la forma de vender y comprar productos o servicios en Internet. La crisis de la pandemia del COVID -19 ha tenido un impacto importante sobre las ventas en grandes y pequeños negocios, sus canales digitales se han dinamizado, optimizando el servicio no presencial, logrando una ventaja competitiva para aumentar el alcance hacia los clientes, crecer como empresa y sobre todo crear nuevos emprendimientos. Por tal motivo, la facturación electrónica tiene un rol importante al momento de realizar ventas en línea ya que según el Sistema de Rentas Internas (SRI) lo define como un documento, el cual debe obedecer a requisitos legales y reglamentarios, garantizando la autenticidad de su origen e integridad de su contenido. (SRI, ¿Qué es y cómo funciona la facturación electrónica?, s.f.). Por otro lado, una aplicación web es fundamental y se complementa al momento de la creación de tiendas virtuales o servicios en línea, debido a que no es necesario instalar en cada computador y se pueden utilizar en cualquier navegador web, además que trabaja con bases de datos que permiten procesar y mostrar información de forma dinámica para el usuario. (Admin, 2015)

A nivel mundial, la facturación electrónica está aumentando de forma acelerada, ya que se crearon plataformas más eficientes para generar documentos tributarios, esto sucedió debido a que los gobiernos buscan reducir la evasión de impuestos con la ayuda de comprobantes que se emitan y validen en tiempo real. De acuerdo a cifras de la investigación global Billentis, de los 500 billones de facturas emitidas a nivel mundial, el 8% fueron electrónicas, de este total, Latinoamérica contribuye con 25 billones, Norteamérica con 7 y Europa con 6 billones, estos números van en aumento, teniendo como estadísticas a nivel mundial que el número de facturas electrónicas crece un 3% anual. Por otro lado, la facturación en África es residual y algunos países como Marruecos y el Magreb están empezando a crecer paulatinamente. Se espera que

la facturación electrónica ya sea un modelo dominante para todo el mundo en el año 2050. (Capacitación Dfacture, 2017)

En Latinoamérica, la facturación electrónica tiene sus inicios en los años 90 y en el siglo XXI. En el año 2003, Chile fue el país pionero en implementar un sistema de facturación electrónica, con la finalidad de lograr una mejor competitividad dentro del mercado. Por otra parte, a mediados del año 2017, países como: Brasil, México, Argentina, Perú y Uruguay, comenzaron a utilizar diseños de facturación más desarrollados. (Barreix & Zambrano, 2018) Con el transcurso del tiempo más países, han comenzado a realizar exitosamente estos cambios, obteniendo como resultado la optimización de su gestión tributaria por la reducción de costos, el aumento de la seguridad, la simplificación de procesos, entre otros beneficios. (ekomercio, s.f.)

En el año 2014, Ecuador comenzó a implementar el sistema de facturación electrónica, donde el Servicio de Rentas Internas (SRI) ha emitido varias normas que establecen el calendario de integración a dicho sistema. En el caso de los contribuyentes existentes previo a la legislación actual, el calendario de implementación abarca desde enero del 2019 hasta el año 2023, cuando se requerirá al último grupo de contribuyentes cumplir con el mandato. (Sovos Compliance, 2020)

El presente proyecto de investigación responderá a la problemática ¿Existe una aplicación web de facturación electrónica que permita la integración con el plugin de WooCommerce y además cumpla con la legislación tributaria del SRI? Dicho problema radica en la ausencia de aplicaciones para brindar el servicio de facturación electrónica que esté al alcance de todas las personas, considerando los factores económico, social y tecnológico de la actualidad, es importante destacar que actualmente ya existen aplicaciones con el servicio de facturación, sin embargo, algunas son consideradas difíciles de manejar para el usuario, para lo cual se necesita de asesoramiento adicional, y otras aplicaciones son costosas. En consecuencia, debido a la

legislación tributaria del SRI es conveniente contar con una aplicación web de facturación electrónica, ya que todos los sujetos pasivos de impuestos están obligados a emitir y entregar comprobantes de venta, a pesar de que el adquiriente no los solicite, se debe emitir y entregar la factura en el momento que se efectúe las transferencias del bien o la prestación del servicio sin importar si la compra es realizada de manera presencial o virtual. (SRI, Facturación física, s.f.)

Entre las diferentes causas que originaron el problema se encuentra como principal la evasión de impuestos, debido a que los gobiernos buscan disminuir dicha evasión con el apoyo de comprobantes que se emitan y validen en tiempo real. En segundo lugar, costos elevados de servicio de facturación electrónica, dicha consecuencia tiene lugar por falta de conocimiento de los usuarios en la adquisición y manejo de programas de facturación electrónica. Por último, la pandemia del Covid -19, ya que el año 2020 marcó un antes y un después en el comercio en línea, obligando a la población a adaptarse a la nueva realidad y consumo.

Los efectos que pueden traer este problema son diversos, entre los que se ha detectado tenemos en primer lugar el cumplimiento de los contribuyentes, aumentando la seguridad y rapidez de emisión de los comprobantes, teniendo un mayor control documental, por otro lado, tenemos que pocos usuarios adquieren el servicio de facturación electrónica para sus tiendas virtuales debido al desconocimiento y a los costos lo que nos conlleva a la última consecuencia que es la baja situación económica y cierre de negocios.

Como una solución a la problemática se realiza la propuesta de la construcción del prototipo de una aplicación web para brindar el servicio de facturación electrónica de una manera fácil, segura y económica para tener un control de las facturas emitidas al cliente y además cumplir con la legislación actual del SRI. Como puede inferirse, es la nueva forma de operatividad 24/7 de los negocios que han optado por trabajar en línea, además, de brindar comodidad tanto a los comerciantes como a sus clientes.

A continuación, se describe trabajos relacionados con la facturación electrónica en Ecuador que han sido desarrollados por estudiantes de la Universidad Central del Ecuador, Escuela Politécnica Nacional y Universidad Politécnica Salesiana

(Cayambe Chicaiza, 2015) El trabajo de titulación “Diseño e implementación de un sistema de facturación electrónica para la Universidad Central del Ecuador” tuvo como objetivo desarrollar, diseñar e implementar un sistema web de facturación electrónica para la emisión de comprobantes electrónicos, que cumpla con las normativas, requisitos legales y reglamentos exigidos por el SRI. El presente proyecto fue desarrollado bajo la plataforma JEE, con herramientas de software libre y utilizando la metodología ágil Scrum. En los resultados de la investigación se obtuvo el módulo de facturación, el cual permite integrar sistemas externos convencionales para la emisión de comprobantes electrónicos, bajo las especificaciones técnicas, operativas y administrativas definidas por la institución, logrando la ejecución del sistema con éxito.

(Arango & Romero, 2019) El trabajo de titulación “Sistema de facturación electrónica para la web de la Distribuidora Mateito” publicada en la Universidad Politécnica Salesiana se propone integrar el módulo de facturación electrónica al sistema de inventario ya existente de la distribuidora. La metodología empleada para el desarrollo del proyecto fue Scrum, en el cual se pudo llevar a cabo el proceso de avance de una forma más organizada. Además, se empleó la arquitectura modelo, vista, controlador (MVC) utilizando el lenguaje de programación PHP. Como conclusión del proyecto se obtuvieron resultados exitosos, ya que se utilizó el mismo lenguaje de programación entre el sistema de facturación y el sistema de inventario ya existente, logrando mejorar el manejo de stock de forma más organizada, reduciendo tiempos de emisión.

(Gualli , 2019) El trabajo de titulación “Desarrollo de un Sistema web prototipo para facturación electrónica de una ferretería” publica en la Escuela Politécnica Nacional, se enfocó en la creación de un sistema que permite emitir comprobantes electrónicos como facturación,

retenciones, guías de remisión, notas de débito y crédito, autorizados a través de los servicios web proporcionados por el SRI, así mismo permitir un manejo centralizado de las sucursales de la ferretería y obtener un manejo eficiente del proceso de facturación. Para el desarrollo del proyecto se utilizó la metodología ágil RAD, utilizando la herramienta Entity Framework, lo cual permitió agilizar el proceso de consumo de la capa de datos. Se obtuvo como resultado el prototipo desarrollado altamente escalable, permitiendo que sea posible desarrollar en un corto plazo nuevas funcionalidades que automaticen otros procesos similares.

OBJETIVOS GENERALES Y ESPECÍFICOS

Objetivo General

Crear un prototipo de una aplicación web para brindar un servicio de facturación electrónica para usuarios u organizaciones que cuenten con una tienda online creada con WooCommerce.

Objetivos Específicos

- ✓ Analizar los principales componentes de una plataforma para facturación electrónica y definir los requerimientos de la aplicación.
- ✓ Diseñar la estructura y funcionamiento interno del prototipo para facturación electrónica y la integración con WooCommerce.
- ✓ Construir un prototipo de una aplicación web para brindar un servicio de facturación electrónica.
- ✓ Evaluar los resultados del funcionamiento del prototipo en base a métricas.

1. REVISIÓN DE LA LITERATURA O FUNDAMENTOS TEÓRICOS

Dentro de este capítulo se describirá y analizará los fundamentos teóricos necesarios para la construcción del sistema de facturación electrónica, el cual se basa en tres principales pilares: facturación electrónica, firma electrónica y el desarrollo de aplicaciones web.

1.1.Facturación electrónica

“La facturación electrónica es otra forma de emisión de comprobantes de venta que cumple con los requisitos legales y reglamentarios exigibles para su autorización por parte del Servicio de Rentas Internas SRI, garantizando la autenticidad de su origen e integridad de su contenido, ya que incluye en cada comprobante la firma electrónica del emisor.” (SRI, Guía para contribuyentes, s.f.)

El uso de la facturación electrónica tiene como objetivo principal reducir la evasión de impuestos con la ayuda de sistemas que emitan comprobantes y se validen en tiempo real. Además, tiene como propósito automatizar los procesos contables para evitar errores al momento de realizar manualmente la facturación.

1.1.1. Consideraciones para la emisión de la factura electrónica

El documento electrónico debe tener la firma electrónica con un certificado digital que corresponda a la entidad emisora del Ecuador. (Cayambe Chicaiza, 2015) De manera análoga, tanto el emisor como receptor deben tener en cuenta las siguientes consideraciones:

Consideraciones por parte del emisor

- ✓ Se debe tener el consentimiento previo del receptor del comprobante electrónico, caso contrario el emisor debe entregar el RIDE, el cual es una representación impresa del documento electrónico con igual validez tributaria. (SRI, Guía para contribuyentes, s.f.)

- ✓ El sistema debe garantizar de modo automático la autenticidad del origen y la integridad de las facturas.
- ✓ El emisor debe contar con un sistema que garantice la recuperación íntegra y el almacenamiento de las facturas electrónicas con la seguridad de que serán legibles, entendibles y confiables. (Cayambe Chicaiza, 2015)

Consideraciones por parte del receptor

Por otra parte, la consideración más importante que debe tener el receptor es disponer de una cuenta de correo electrónico para recibir la factura electrónica emitidos a su nombre, en caso de no contar con un correo puede recibir de manera impresa el RIDE por parte del emisor o revisar el portal del Sistema de Rentas Internas (SRI) para poder consultar y descargar las facturas que ha recibido electrónicamente.

Cabe mencionar que tanto el emisor como receptor deben almacenar la información de los comprobantes electrónicos por siete años. (SRI, Guía para contribuyentes, s.f.)

1.1.2. Emisión de comprobantes electrónicos

La emisión de comprobantes electrónicos, en este caso la factura electrónica, se requiere solicitar la autorización emitida por el SRI. El esquema de comprobantes electrónicos tiene dos ambientes: pruebas y producción.

Ambiente de pruebas

El ambiente de pruebas, también llamado Certificación, permite realizar pruebas y ajustes en los sistemas informáticos, con el fin de revisar el funcionamiento del esquema de emisión electrónica. Garantiza que la información que es enviada al SRI cumpla con los requerimientos establecidos, especialmente la información del contribuyente. Además, permite realizar los ajustes necesarios y corregir posibles errores. Es importante tener en cuenta que no tienen

validez tributaria los comprobantes emitidos en este ambiente. (SRI, Facturación electrónica, s.f.)

Ambiente de producción

Una vez que se concluya con el Ambiente de Pruebas, el contribuyente puede solicitar la autorización para que se habilite el Ambiente de Producción. La aprobación emitida por el SRI tendrá una vigencia indefinida, todos los comprobantes electrónicos autorizados en este ambiente tienen validez tributaria. (SRI, Facturación electrónica, s.f.)

Para ingresar la solicitud de ambos ambientes se debe ingresar a la página “*SRI en Línea*”, ingresando el RUC y clave de acceso, seleccionar la opción “Comprobantes electrónicos” y escoger el tipo de ambiente que se necesite.

Los comprobantes electrónicos emitidos en el ambiente de Pruebas así como en el ambiente de Producción, deberán cumplir con los requisitos establecidos en el Reglamento de Comprobantes de Venta, Retención y Documentos Complementarios y las resoluciones establecidos para el efecto. (SRI, Guía para contribuyentes, s.f.)

1.1.3. Requisitos para emitir facturas electrónicas

El contribuyente debe considerar los siguientes requisitos previo a la solicitud de autorización que debe realizar ante el SRI. Entre los más importantes e indispensables a considerar se tiene los siguientes: (SRI, Guía para contribuyentes, s.f.)

- ✓ Firma electrónica, puede ser adquirida en cualquier entidad de certificación autorizadas en el país.
- ✓ Software que genere comprobantes electrónicos, se puede utilizar herramientas gratuitas de emisión de comprobantes propias del SRI o un sistema propio del contribuyente.
- ✓ Conexión a Internet

- ✓ Clave de acceso a SRI en línea
- ✓ Convenio de débito registrado, para pagos mensuales o semestrales de las obligaciones que tiene cada establecimiento, como el pago del IVA, pago del impuesto a la renta, entre otros. Cabe resaltar que el SRI da autorizaciones para la emisión de facturas por el periodo de un año, siempre y cuando las obligaciones del contribuyente estén al día, caso contrario, la autorización será únicamente por tres meses.

1.1.4. Ventajas y desventajas de la facturación electrónica

Al momento de utilizar un sistema de facturación electrónica se tiene que tomar en cuenta que existen ventajas y desventajas que intervienen al momento de la emisión de documentos electrónicos. A continuación, se detallan los beneficios e inconvenientes más destacados:

Ventajas

Existen múltiples beneficios al momento de emitir una factura electrónica, los más importantes a considerar son:

- ✓ Reducción de errores humanos
- ✓ Menor costo en el cumplimiento de obligaciones tributarias
- ✓ Procesos administrativos más rápidos y eficientes
- ✓ Mayor seguridad en el resguardo de los documentos
- ✓ Eliminación de almacenamiento físico
- ✓ Cuidado del medio ambiente al disminuir el uso del papel, lo que supone una mejora en el servicio de atención al cliente y un importante efecto positivo en la naturaleza.
- ✓ Consulta en línea de la validez de las transacciones
- ✓ Tiene igual validez tributaria que la facturación física, por lo que se mantienen las dos modalidades. (SRI, Guía para contribuyentes, s.f.)

Desventajas

Por otro lado, en la actualidad aún existen negocios que no utilizan esta modalidad para la emisión de comprobantes de venta, debido a las siguientes desventajas: (tugesto, 2020)

- ✓ Puede ocurrir atrasos en el caso que los trabajadores no conocen el sistema, es decir, es indispensable tener una capacitación para la adaptación de la nueva metodología de emisión de facturas, esto conlleva a tener costos adicionales e inversión de tiempo.
- ✓ El SRI rechaza automáticamente a los documentos electrónicos mal emitidos, teniendo como consecuencia que, al querer corregirlos, implica que cambien la fecha de emisión, llegando a perjudicar a las empresas en el cobro de intereses, la contabilidad y a los inventarios de la misma.
- ✓ Adquisición de sistemas informáticos para emitir documentos electrónicos, lo que implica una inversión económica representativa, ya que es recomendable disponer de herramientas y sistemas de seguridad rigurosos que garanticen el almacenamiento completo de la información.

1.1.5. Proceso de Facturación electrónica

El proceso de facturación electrónica en el Ecuador consiste en la emisión de comprobantes de pago electrónicos (factura, nota de débito y crédito, entre otros) en formato XML de acuerdo a las características y condiciones establecidas por el SRI. Dentro del proceso se puede describir los siguientes pasos: (SRI, Ficha técnica comprobantes electrónicos esquema offline, 2020)

- ✓ El sistema de facturación genera el XML de la factura con la firma electrónica.
- ✓ Validación y recepción de Factura, el SRI es quien se encarga de la legalización. En caso que los comprobantes no sean autorizados, el contribuyente recibirá una descripción indicando el motivo.

- ✓ Cada comprobante generado tiene una clave de acceso única, la cual está compuesta por 49 dígitos numéricos y se genera automáticamente, ya que constituye un requisito obligatorio que le dará el carácter de único a cada comprobante y a la vez se genera el número de autorización, en base a lo mencionado anteriormente el SRI emite una respuesta del estado de la autorización.
- ✓ Finalmente, el receptor almacena la factura y es obligación de los ciudadanos que reciben la factura electrónica validar sus comprobantes mediante el portal WEB del Servicio de Rentas Internas.

1.1.6. XML

Extensible Markup Language, conocido por sus siglas en inglés como XML, es un lenguaje de marcado que establece un conjunto de códigos que se pueden adaptar en el análisis de datos o la lectura de textos creados por personas o computadoras. Su principal función es representar información estructurada en la web, de tal manera que la información pueda ser almacenada, procesada, visualizada e impresa por cualquier aplicación o dispositivo. La ventaja de utilizar XML es que diferencia el contenido y el formato de presentación, lo que le convierte en un archivo fácilmente procesable. (exes, s.f.)

En base al contexto de generar comprobantes electrónicos, el XML es un archivo que se genera con toda la información que contiene la factura, como fecha, datos del cliente y producto, precios, entre otros. Incluye la información del certificado electrónico que se utilizó para firmar la factura con el fin de validar el documento y confirmar que no ha sido modificado después de ser emitido.

El formato XML tiene grandes beneficios para el intercambio de facturas ya que se plantea como un estándar para el intercambio de información estructurada entre diferentes plataformas. Además, si una tercera persona decide utilizar un documento creado en este formato, le resultará

sencillo de entender su estructura y procesarla. En base a la sintaxis que tiene el archivo XML, todo elemento tiene que tener su correspondiente etiqueta de inicio y de cierre, las etiquetas pueden estar vacías. A su vez, debe contener un elemento, llamado raíz de documento, el cual debe contener a los demás y estar correctamente anidados. (exes, s.f.)

El SRI cuenta con un portal en línea, donde se encuentran disponibles los esquemas de las facturas electrónicas en formato .xsd, con los cuales los contribuyentes deben emitir el comprobante en formato XML. Cabe mencionar que dicho sistema tiene sus propias aplicaciones informáticas que brindan el servicio gratuito de facturación para los contribuyentes. (Sovos, 2021)

1.2.Firma electrónica

De acuerdo al Registro Civil “La firma electrónica es el documento virtual que certifica la vinculación de la firma electrónica con una persona determinada; es la equivalencia de la firma manuscrita ya que tiene la misma validez legal y está amparada en la Ley de Comercio Electrónico” (Registro Civil, s.f.)

El uso de la firma electrónica tiene como garantía brindar autenticidad, es decir que la información del documento y su firma electrónica corresponde sin duda a la persona que ha firmado, a su vez garantiza integridad, lo cual indica que la información no ha sido modificada o alterada luego de su firma, otro punto es el No repudio, donde la persona que ha firmado electrónicamente no puede negar su auditoría y por último, se garantiza la confidencialidad, debido a que la información es cifrada y por voluntad del emisor, solo permite que el receptor pueda descifrarla. (SecurityData, 2019)

De acuerdo con la Ley de Comercio Electrónico, Firmas y Mensajes de Datos específicamente el artículo 14 menciona: “La firma electrónica tendrá igual validez y se le reconocerán los mismos efectos jurídicos que a una firma manuscrita en relación con los datos consignados en

documentos escritos, y será admitida como prueba en juicio” (Congreso Nacional, 2002), de lo que se concluye que la firma electrónica si tiene validez legal.

1.2.1. Certificados digitales

Un certificado digital es un fichero informático, el cual es generado por una entidad que brinda el servicio de certificación, y además asocia unos datos de identidad a una persona física, empresa u organismo, con el fin de crear una identidad digital (QualityConta, 2020). En otras palabras, es el único medio que permite garantizar técnica y legalmente la identidad de una persona virtualmente.

Es de suma importancia conocer que un certificado digital consta de dos tipos de claves criptográficas: una pública y una privada, creadas con un algoritmo matemático, el cual funciona de forma que cifra con una de las claves y únicamente se puede descifrar con su clave pareja. El propietario del certificado debe mantener bajo su custodia la clave privada, ya que en el caso de que se pierda, el sustractor podría suplantar la identidad del titular. Por otro lado, se tiene la clave pública, la cual forma parte de lo que se denomina Certificado digital. (Universidad Politécnica de Valencia, 2020)

Con el uso de los certificados, los usuarios ganan confidencialidad, integridad y seguridad, ya que el receptor del documento firmado puede tener la certeza que es original y no ha sido manipulado. Además, permite verificar documentos digitales, evitando la falsificación o modificación de escritos, garantiza la autenticidad del firmante, incrementa la eficiencia de la gestión pública y privada, y finalmente ahorra tiempo y dinero.

1.2.2. Entidades Certificadoras en Ecuador

En Ecuador, la Agencia de Regulación y Control de las Telecomunicaciones (Arcotel) es la encargada de autorizar a entidades emitir certificados digitales de firma electrónica. Algunas de ellas se listan a continuación: (SRI, Guía para contribuyentes, s.f.)

Banco Central

El Banco Central del Ecuador emite certificados digitales de firma electrónica y otros servicios relacionados con la certificación electrónica para el Sector Público, Personas Jurídicas y Personas Naturales; garantizando la seguridad jurídica y tecnológica en entornos electrónicos cumpliendo el marco legal, las normas y estándares nacionales e internacionales de certificación electrónica. (Banco Centra del Ecuador, 2020)

Security Data

Seguridad en Datos y Firma Digital S.A, Security Data, es una entidad de certificación de la información privada para la emisión y comercialización de firmas electrónicas y servicios relacionados, la cual quiere crear una identidad digital a todos los ciudadanos para interactuar dentro de un entorno tecnológico. (Security Data, 2021)

ANFAC

ANFAC, Autoridad de Certificación Ecuador CA, es la entidad sucesora de Authority of Certification Ecuador S.A. (ANF), fue una de las primeras autoridades de certificación, siendo la única que dispone de su propia tecnología de firma electrónica con infraestructura PKI de nivel Notariado Electrónico, y servicios oficialmente acreditados de emisión de certificados y sellos digitales. (Jaramillo, 2019)

Consejo de la Judicatura

El Consejo de la Judicatura, CJ, es una entidad acreditada en el país para emitir y gestionar certificados de firma electrónica a personas que acreditan la representación legal de la empresa o institución. Cabe mencionar, que dicha entidad se deslinda toda responsabilidad de las solicitudes de certificados y registros de suscriptores realizados con suplantación de identidad o datos fraudulentos. (Consejo de la Judicatura, 2019)

1.2.3. Tipos de firma

La entrega de la Firma Electrónica se realiza en dos tipos de formatos: token y en archivo, según las entidades certificadoras del Ecuador, ya que se han identificado dos principales trámites, facturación electrónica e importación y exportación de productos, los cuales diferencian los dos tipos de formatos. (Security Data, 2021)

Token

Es un dispositivo inteligente que no requiere instalación ni configuración adicional, solo se necesita un puerto USB libre en la computadora y no permite copiar la información del archivo fuente. Además, sirve para personas o empresas que se dedican a la importación y exportación.

Archivos PFX-P12

El formato en archivo, anteriormente era entregado en CD o podía ser copiado en una USB pero, actualmente debido a la pandemia del covid-19 es enviado mediante correo electrónico un archivo con extensión .p12. Se puede utilizar la firma electrónica sin necesidad de memorias USB porque se puede almacenar directamente en la computadora.

Una gran diferencia entre el formato Token es que se puede realizar múltiples respaldos del archivo en varios dispositivos.

1.2.4. Funcionamiento de la firma electrónica

El funcionamiento de la firma electrónica consta de una clave pública y una clave privada, cada una de ellas tiene un par de claves, una se usa para cifrar y la otra para descifrar.

El emisor mediante una función llamada “hash” genera un resumen del mensaje al firmar. Hash es una operación que se ejecuta sobre una colección de datos, de manera que independientemente obtiene como resultado otro conjunto de datos. El emisor cifra el resumen del mensaje con la clave privada, originando la firma electrónica, la cual se añade al mensaje original.

El receptor, al recibir el mensaje, obtiene mediante la función “hash” un nuevo resumen. Además, descifra la firma utilizando la clave pública del emisor obteniendo la información que el emisor envió. Si ambos coinciden la firma es válida, ya que cumple los criterios de autenticidad e integridad además del de no repudio. (AESA, s.f.)

RSA-SHA1

RSA es en realidad dos algoritmos, uno para cifrado asimétrico y otro para firmas digitales, en el caso que se utilice para firmas digitales RSA realiza un cifrado híbrido, ya que tiene algunas limitaciones operativas con la variante más utilizada conocida como PKSC. Normalmente no se cifran los datos directamente, en su lugar se selecciona una pequeña secuencia de bytes aleatorios, llamada claves de sesión, la cual es encriptada con RSA y luego utiliza la clave de sesión con un algoritmo simétrico de cifrado para procesar todo el mensaje. (Pornin, 2011)

Por otra parte, SHA es una función hash, es decir no tiene clave, solo código, es decir, un algoritmo ejecutable. La función Hash es un componente importante para la firma digital, ya que prueba la acción consciente de un firmante designado sobre un dato. Al igual que el cifrado asimétrico, utiliza pares de claves y restricciones asociadas en los datos firmados. (Pornin, 2011)

Por lo tanto, se puede decir que SHA1 es una función hash criptográfica, el cual es utilizado para la firma de documentos y certificaciones, mientras que RSA es un algoritmo de cifrado y descifrado para brindar comunicaciones seguras, es por este motivo que se juntan estos dos conceptos para comprender el funcionamiento y la seguridad de la firma electrónica.

1.2.5. Estándar XadES_BES

XadES es un estándar del tipo de firma más simple, utiliza una extensión XMLDSig, por lo tanto, es un fichero XML que proporciona validez a largo plazo a certificados, listas de revocación, etc. Además, es utilizado para evitar problemas de codificación de caracteres y

poder firmar archivos binarios. Cabe recalcar, que en el caso de que el documento se incluya en la firma siempre debe estar codificado en base64. (Agencia de Tecnología y Certificación electrónica, 2019)

Para la emisión de los documentos electrónicos se debe obligatoriamente firmar cada archivo XML bajo el estándar de firma digital de documentos XadES_BES, el cual pertenece a la familia de firmas avanzadas (Viafirma, s.f.). Dicho estándar es una firma básica que cumple con los requisitos legales de la directiva y es utilizada cuando no se ha definido una política. Existen tres tipos de firmas XadES, considerando la manera en la que se relacionan la firma y el documento. (Agencia de Tecnología y Certificación electrónica, 2019)

- ✓ **Attached:** La firma y la información se devuelven en un único fichero XML, sin embargo, los dos se encuentran separados mediante etiquetas dentro del mismo y se firma únicamente la parte donde se encuentra el documento.
- ✓ **Detached:** se encuentran en archivos diferentes la firma y el documento.
- ✓ **Enveloped:** se encuentran en el mismo archivo, pero a diferencia de “Attached” se firma todo el fichero

1.3.Desarrollo de aplicaciones Web

Actualmente, el desarrollo de aplicaciones web tiene una gran importancia ya que presenta posibilidades ilimitadas, debido a que se impone de manera global como una herramienta tecnológica.

Las aplicaciones web son programas que se ejecutan gracias a un servidor web, donde los usuarios pueden acceder a través de un dispositivo inteligente con Internet. Todos los datos con los que interactúa el usuario con la aplicación están almacenados en la web y son procesadas en la misma. Cabe resaltar que están basadas en la arquitectura cliente – servidor, es decir, combinan scripts del lado del servidor para el almacenamiento y recuperación de la

información, con scripts del lado del cliente para mostrar la información a los usuarios. Como resultado se obtiene interacción a través de formularios en línea, sistema de gestión de contenido, carrito de compras, entre otros. (Vega, 2020)

Existen aplicaciones web desarrolladas con sistemas de gestión de contenidos (CMS), permitiendo a los administradores crear y gestionar el contenido de la aplicación de manera rápida y sencilla. Es recomendable utilizar dicho sistema cuando el contenido de la aplicación debe ser actualizado continuamente. (Vega, 2020)

1.3.1. WordPress

WordPress es un sistema de gestión de contenido (CMS), considerado ideal para principiantes o para un público que no tiene conocimientos técnicos. Permite crear un blog u otro tipo de aplicación, además, dispone de un sistema de plugins, que permite extender sus capacidades, teniendo como resultado un CMS más flexible. (WebEmpresa, 2019)

WordPress se divide en tres partes: primero el Core, el cual es WordPress en sí, totalmente gratis y descargable. En segundo lugar, se encuentran los temas, que sirven para cambiar la apariencia de la web, existe un repositorio gratuito para acceder a los diferentes temas, pero también hay recursos de pago fuera del repositorio. Por último, se encuentran los plugins, quienes son los encargados de extender las capacidades del CMS, siendo utilidades que pueden convertir la aplicación web según las necesidades del usuario. (WordPress, WordPress, s.f.)

Por otro lado, es importante conocer que existen otros gestores de contenido específicos para crear tiendas virtuales como PrestaShop, Magento, entre otros. Sin embargo, WordPress puede ser una opción válida, debido a que dispone de varios plugins que permiten incorporar una tienda en línea, uno de los más importantes a señalar es WooCommerce. (WebEmpresa, 2019)

1.3.2. WooCommerce

WooCommerce es un plugin que permite la implementación de una tienda virtual completamente funcional en la web. Tiene todo lo necesario para empezar a vender en línea, ya que contiene las páginas de la tienda, el carrito de compras, implementación desde el inicio y fin de la compra, pasarela de pago y permite la subida ilimitada de productos con su respectivo precio, descripción, variación, etc.

La mayoría de desarrolladores prefieren trabajar con WooCommerce para crear, personalizar y escalar una tienda para cumplir con los requerimientos del usuario y realizar mejoras a través de extensiones o soluciones personalizadas en corto tiempo. La principal ventaja que tiene dicho plugin es la integración con otras aplicaciones o servicios mediante una API REST sólida y webhooks, eventos que desencadenan acciones. (WordPress, WooCommerce, s.f.)

1.3.3. Integración de aplicaciones

La integración de aplicaciones es un proceso que permite el intercambio de datos, recursos y procesos entre diferentes sistemas, es decir, mejora el flujo de trabajo y los datos que se tiene en las aplicaciones de software.

La integración no solo establece conexión entre sistemas, también aporta la utilidad y los beneficios que se obtienen con las funciones que surgen al vincular los distintos sistemas. Para realizar la integración de aplicaciones se debe utilizar una interfaz de programación o API. (Naeem, 2020)

API REST

“Las API son conjuntos de definiciones y protocolos que se utilizan para diseñar e integrar el software de las aplicaciones. Suele considerarse como el contrato entre el proveedor de información y el usuario, donde se establece el contenido que se necesita por parte del consumidor (la llamada) y el que requiere el productor (la respuesta).” (Red Hat, 2020) En

otras palabras, las API permiten la interacción con un sistema para obtener datos o ejecutar una función, de tal manera que se comprenda la solicitud y se cumpla.

Para que una API se considere Restful, se debe tomar en cuenta los siguientes criterios (Red Hat, 2020):

- ✓ Arquitectura cliente – servidor para realizar peticiones HTTP.
- ✓ Datos que puedan almacenarse en caché y optimizar las interacciones entre el cliente y el servidor.
- ✓ Un sistema en capas para organizar en jerarquías invisibles para el cliente cada uno de los servidores que participan en la recuperación de la información solicitada.
- ✓ Una interfaz uniforme, para que la información se transfiera de forma estandarizada, cumpliendo las siguientes condiciones: Los recursos solicitados deben ser identificados e independientes de las representaciones enviadas al cliente y el cliente debe poder manipular los recursos a través de la representación que recibe, ya que esta contiene suficiente información para permitirlo.

En conclusión, las API de REST son más rápidas y ligeras, cuentan con mayor capacidad de ajuste y, por ende, resultan ideales para el internet de las cosas (IoT) y la integración de aplicaciones.

2. MARCO METODOLÓGICO

La construcción del sistema de facturación electrónica se basa en la metodología ágil de Scrum. Scrum es un proceso de gestión que ayuda a reducir la complejidad en el desarrollo de productos para satisfacer las necesidades y requerimientos de los clientes. A la vez, se basa en el empirismo, es decir, asegura que el conocimiento es resultado de la experiencia y de la toma de decisiones basándose en lo que se conoce. Finalmente emplea un enfoque iterativo e incremental que optimiza la predictibilidad y el control del riesgo. (Scrum Org., s.f.)

Cabe mencionar que existen tres pilares que soportan toda la implementación del control de los procesos empíricos: transparencia, inspección y adaptación. (Scrum Org., s.f.)

Para la aplicación de la metodología Scrum en el sistema se va a utilizar la herramienta de Jira para poder gestionar de una manera adecuada el desarrollo del proyecto. Jira Software proporciona tableros de Scrum listos para utilizar, los cuales son centros de gestión de tareas, donde se asignan trabajos personalizables. (Jira Software, s.f.)

2.1.Eventos

En esta sección se describe los cuatro eventos formales que ayuda a la metodología Scrum para la inspección y adaptación en el desarrollo de proyectos.

- Planificación del Sprint

En este evento es donde se planifica las actividades de trabajo durante el Sprint por el equipo de Scrum. Quien se asegura que lo planificado se lleve a cabo y que los integrantes del desarrollo entiendan su propósito es el Scrum Master.

En el desarrollo del sistema dicho evento se realiza los días viernes de cada semana, durante la construcción del proyecto.

- Scrum Diario

El Scrum Diario es una reunión de 15 minutos para que el equipo de Desarrollo sincronice sus actividades y se cree un plan para el siguiente día. El desarrollo se lleva a cabo revisando los avances considerando el último Scrum diario y a la vez, se realiza una proyección acerca del trabajo a presentar antes del siguiente.

- Revisión del Sprint

Durante este evento, el equipo Scrum y las personas interesadas mencionan acerca de lo realizado durante el Sprint, basándose en los avances y cambios, todos los participantes dentro de la revisión colaboran para determinar las siguientes tareas a realizar para optimizar el valor.

En el presente proyecto se realiza la revisión del Sprint los días martes, donde se presenta avances e inquietudes que se tiene al momento del desarrollo y los días viernes de todas las semanas se presenta el resultado planteado en el primer evento.

- Retrospectiva del Sprint

Este punto es importante ya que el equipo puede auto inspeccionarse a sí mismo y sus actividades para crear un plan de mejoras que sean llevadas a cabo en el siguiente Sprint.

2.2.Roles

Considerando los roles que tiene dicha metodología, se procede a describir la función de cada uno y además señalar quien está a cargo en el desarrollo del presente proyecto.

- Product Owner

Se denomina como propietario del producto, es la persona conocedora del negocio y la visión del producto final. En el desarrollo de este proyecto, quien cumple con este rol es el dueño de la tienda virtual que requiera el servicio de facturación electrónica.

- Scrum Master

El Scrum Master es quien se encarga de la correcta ejecución de la metodología durante la construcción del sistema y cuya función primordial es la de actuar como enlace entre el dueño del producto y el equipo de desarrollo. El tutor del presente trabajo de titulación es quien cumple este rol dentro del desarrollo de KV – Facturación.

- Equipo de desarrollo

El equipo de desarrollo es conformado por las personas que realizan el trabajo de entregar el producto terminado. La autora del desarrollo del presente sistema es quien cumple este rol.

2.3.Elementos

En esta sección se mencionan los elementos que tiene la metodología Scrum, los cuales son técnicas que se emplea para comprender el desarrollo del producto y además organizar las actividades pendientes y realizadas.

- Product Backlog

Enumera todas las características, funcionalidades, requisitos, mejoras y correcciones que se puede realizar sobre el proyecto. El dueño del producto es el responsable de su contenido y disponibilidad.

En el presente proyecto se realiza un análisis de requerimientos donde se detalla las características y funcionalidad del sistema, con el cual fue desarrollado. (Ver Análisis de Requerimientos)

A continuación, se presenta el Backlog con su respectiva épica, utilizada para el desarrollo del sistema.

Figura 1

Backlog según requerimientos del dueño del producto.

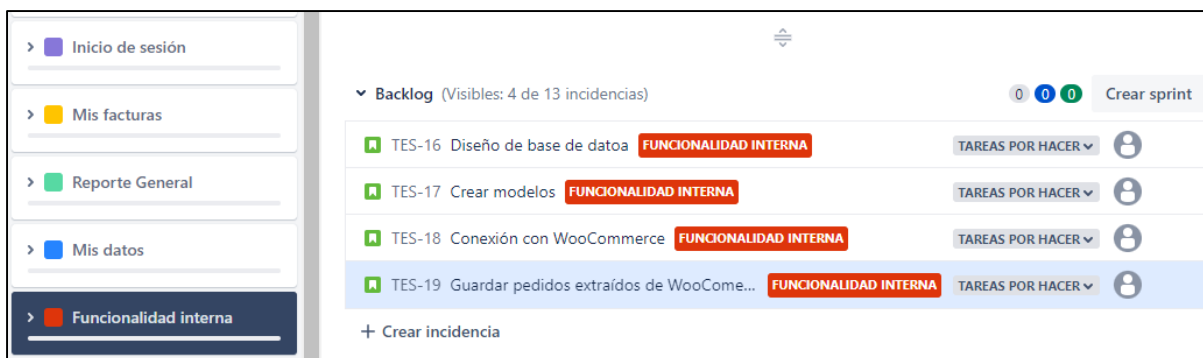


Elaborado por: Karla Moyón, autora del presente proyecto.

Por otro lado, se tiene que considerar las funcionalidades internas del sistema para lograr las bases iniciales del proyecto. A continuación, se presenta el backlog correspondiente:

Figura 2

Backlog para el desarrollo del sistema



Elaborado por: Karla Moyón, autora del presente proyecto.

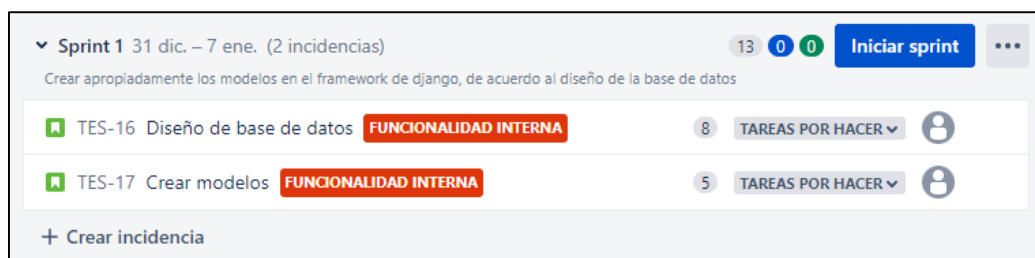
- **Sprint Backlog**

También conocido como lista de pendientes del Sprint, es el subconjunto de elementos del Product Backlog. En pocas palabras es una predicción hecha por el equipo de desarrollo acerca del trabajo necesario para entregar los avances del proyecto.

A continuación, se detalla los Sprints realizados para la entrega del sistema completo, donde se puede observar las fechas asignadas para cada sprint, el objetivo y sus respectivas incidencias, es decir las actividades que se deben realizar para cumplir con el objetivo planteado.

Figura 3

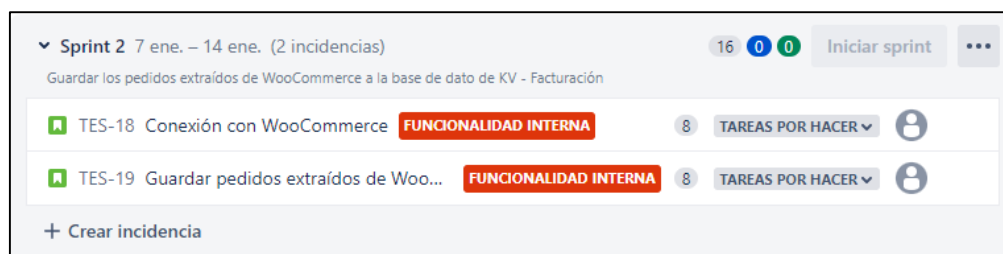
*Sprint 1 – (Ver construcción **Sprint 1**)*



Elaborado por: Karla Moyón, autora del presente proyecto.

Figura 4

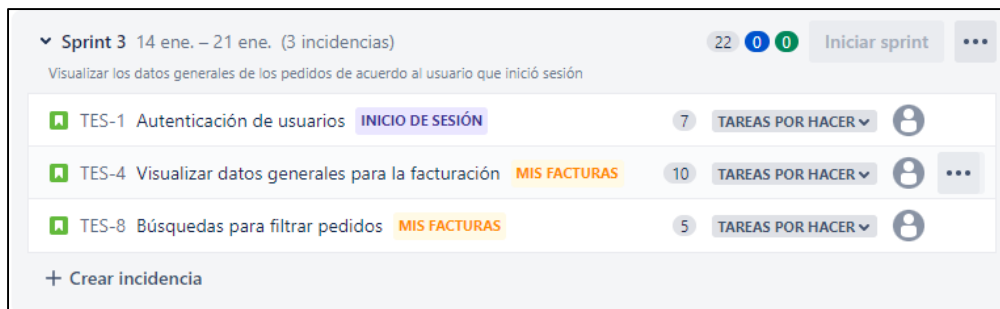
*Sprint 2 – (Ver construcción **Sprint 2**)*



Elaborado por: Karla Moyón, autora del presente proyecto.

Figura 5

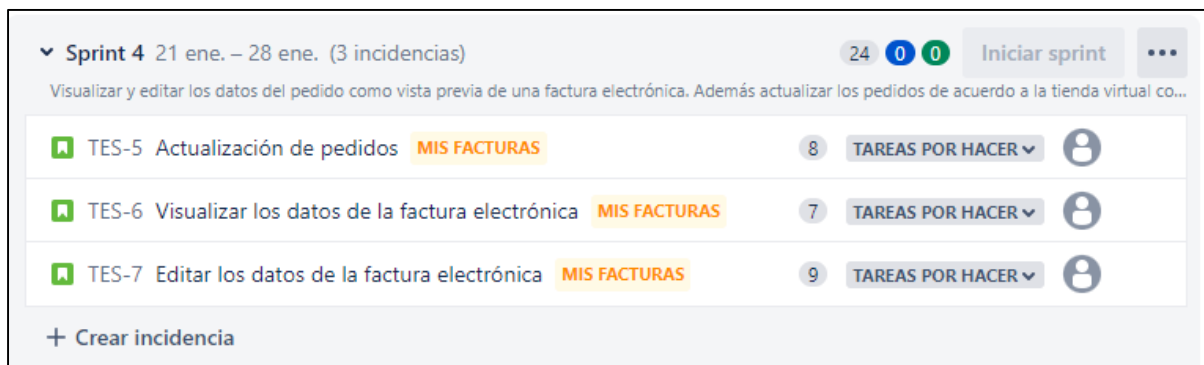
*Sprint 3 – (Ver construcción **Sprint 3**)*



Elaborado por: Karla Moyón, autora del presente proyecto.

Figura 6

*Sprint 4 – (Ver construcción **Sprint 4**)*



Elaborado por: Karla Moyón, autora del presente proyecto.

Figura 7

*Sprint 5 – (Ver construcción **Sprint 5**)*



Elaborado por: Karla Moyón, autora del presente proyecto.

3. RESULTADOS

3.1. Análisis

En el presente capítulo se analizará el problema planteado, el cual abarca el análisis del proceso de facturación electrónica, análisis de viabilidad y finalmente el análisis de requerimientos. Al terminar el capítulo se podrá identificar estrategias a utilizar para resolver el proyecto, elementos necesarios que se necesita para cumplir con el fin del sistema. Por último, conocer cada tarea a realizar para la “Construcción de una aplicación Web de facturación electrónica para tiendas virtuales implementadas con WooCommerce en Ecuador”.

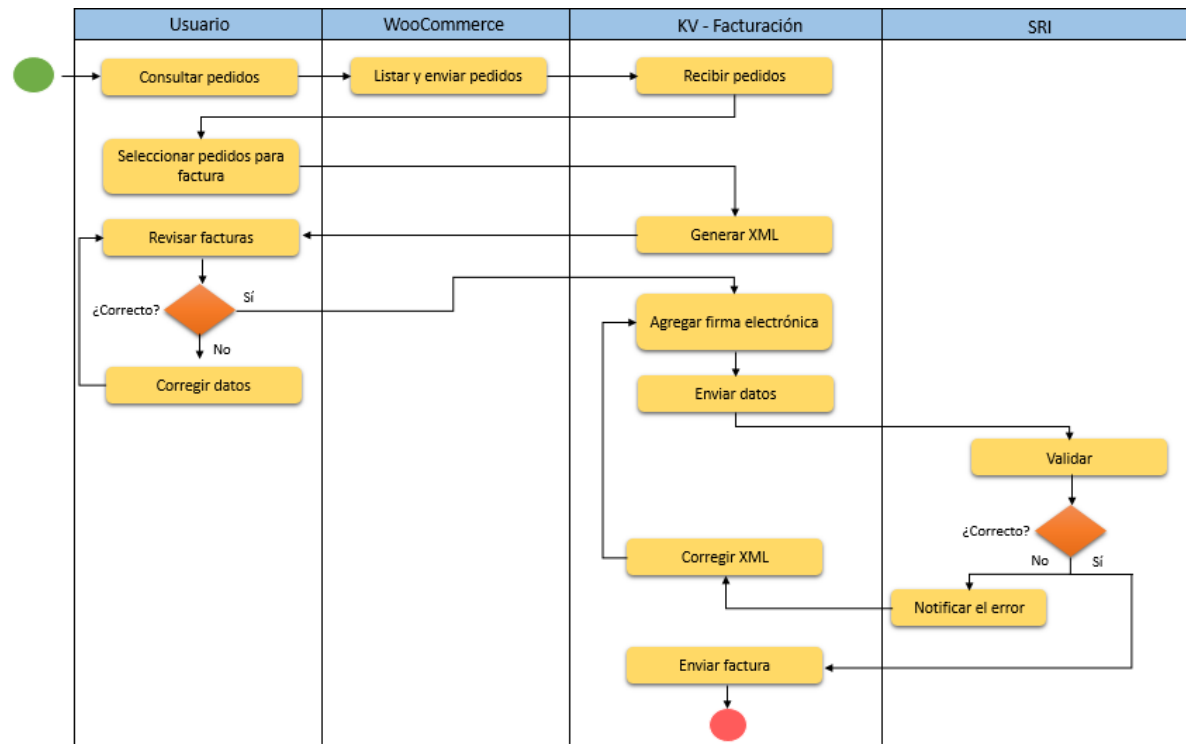
3.1.1. Análisis del proceso de facturación electrónica

La factura electrónica, al ser el nuevo esquema de emisión de comprobantes del SRI y además un servicio que tiene los mismos efectos legales y tributarios que una factura en papel, debe cumplir con los mismos requisitos. Es decir, debe constar con los datos de la compra donde se mencione los detalles del producto, cliente y del emisor de la factura.

En la figura 8 se muestra el diagrama del procedimiento de negocio para la construcción del sistema de facturación electrónica, llamado KV-Facturación.

Figura 8

Diagrama de Flujo del Proceso de Facturación electrónica



Nota: Proceso de Facturación Electrónica en el Ecuador con cada uno de los actores
Elaborado por: Karla Moyón, autora del presente proyecto.

El proceso de facturación está compuesto por cuatro actores: usuario, WooCommerce, KV-Facturación y el SRI, los cuales son responsables de ejecutar el proceso del presente proyecto.

A partir del Diagrama de Flujo de Proceso de Facturación electrónica (Figura 8 - **Análisis del proceso de facturación electrónica**), se muestra a continuación un detalle de cada tarea que realiza el sistema.

1. Consultar pedidos

En primera instancia el usuario realiza una petición para consultar pedidos través de la tienda virtual, esta información se obtiene a través de WooCommerce. Cabe mencionar que la construcción de la tienda virtual no es el objetivo del presente proyecto. Esta información es proporcionada al Sistema KV-Facturación a través de los servicios Web que proporciona la API de WooCommerce. El formato de intercambio es en formato Json.

2. Listar y enviar pedidos

En esta tarea el actor, WooCommerce, al momento de listar los pedidos presenta y envía la siguiente información: Fecha, producto, valor, estado (Completado, En espera, Espera de pago, Procesando, Cancelado, etc.), cliente, entre otros.

3. Recibir pedidos

El Sistema KV-Facturación a través de la REST API de WooCommerce realiza una petición para obtener los pedidos de la tienda virtual. Al realizarse el proceso de sincronización, se debe realizar una revisión automática de los pedidos que anteriormente estaban con estado de Espera, Espera de pago y Procesando. Se actualiza desde la última vez que se haya ejecutado la sincronización hasta la fecha actual.

4. Seleccionar pedidos para facturar

En la presente tarea el usuario prosigue a seleccionar los pedidos que desea facturar, ya que debido al estado que tiene la compra, no todos pueden ser facturados, únicamente los pedidos que tengan un estado de “Completado”.

5. Generar XML

Dentro de este paso se genera el archivo XML que constituye la factura electrónica, con los datos proporcionados por la tienda virtual. A continuación, se detallan los principales campos a considerar.

Tabla 1.

Datos del cliente para la factura electrónica

No	Descripción	Requisito
1	Identificación (Número de RUC, Cédula o Pasaporte)	Obligatorio
2	Nombres y Apellidos o Razón Social	Obligatorio
3	Dirección de Correo electrónico	Obligatorio

4	Teléfono	Opcional
5	Dirección	Opcional

Fuente: (SRI, Ficha técnica comprobantes electrónicos esquema offline, 2020)

Tabla 2.

Datos del producto o servicio que ofrece el vendedor

No	Descripción	Requisito
1	Impuestos o tarifas parametrizables	Obligatorio
2	Código identificador del producto o servicio asignado por el contribuyente	Obligatorio
3	Código identificador auxiliar del producto o servicio	Opcional
4	Nombre del producto o servicio	Obligatorio
5	Cantidad	Obligatorio
6	Valor unitario	Obligatorio
7	Descuento	Obligatorio
8	Impuesto (Ver tabla 3)	Obligatorio
9	Tarifa de impuesto (Ver tabla 4)	Obligatorio
10	Campos adicionales (máximo tres campos de hasta 300 caracteres)	Opcional

Fuente: (SRI, Ficha técnica comprobantes electrónicos esquema offline, 2020)

Tabla 3

Código de los impuestos

Impuesto	Código
IVA	2
ICE	3
IRBPNR	5

Fuente: (SRI, Ficha técnica comprobantes electrónicos esquema offline, 2020)

Tabla 4

Tarifa de impuestos

Porcentaje del IVA	Código
0%	2
12 %	3
14 %	5
No objeto de Impuesto	6
Exento de IVA	7

Fuente: (SRI, Ficha técnica comprobantes electrónicos esquema offline, 2020)

Según las normas del SRI, la tarifa vigente del impuesto al valor agregado (IVA) es del 12%. (SRI, IMPUESTO AL VALOR AGREGADO (IVA), s.f.) La manera de calcular dicho valor es de la siguiente manera.

Nota: Los precios indicados en la tienda virtual deberán incluir el IVA

$$valor_unitario = \frac{valor_total}{cantidad}$$

$$subtotal = \frac{valor_total}{1.12}$$

$$IVA = \frac{valor_unitario * 12}{100}$$

A continuación, se presenta un ejemplo del cálculo del valor unitario, subtotal e IVA. Además, como se puede observar en la Figura 9 las fórmulas presentadas anteriormente no alteran el valor total que se recibe del pedido, únicamente nos ayuda a confirmar que el Subtotal y el IVA están correctos.

Figura 9

Ejemplo del cálculo del valor unitario, subtotal e IVA a partir del total del pedido recibido

ID	CANT	DESCRIPCIÓN	TARIFA	TOTAL	V. UNITARIO	SUBTOTAL	IVA	TOTAL 2
1	2	Producto 1	12%	352	176	314.29	37.71	352.00
2	1	Producto 2	12%	120	120	107.14	12.86	120.00
3	3	Producto 3	12%	548	182.67	489.29	58.71	548.00
4	2	Producto 4	12%	412	206	367.86	44.14	412.00
5	1	Producto 5	12%	80	80	71.43	8.57	80.00

Realizado por: Karla Moyón, autora del presente proyecto

Tabla 5

Información del Emisor.

No	Descripción	Tipo de campo	Longitud	Requisito
1	Fecha de emisión	Numérico	8	Obligatorio
2	Tipo de Comprobante	Numérico	2	Obligatorio
3	Número de RUC	Numérico	13	Obligatorio
4	Razón Social / Nombres o Apellidos	Alfanumérico	Máx. 300	Obligatorio
5	Nombre comercial	Alfanumérico	Máx. 300	Opcional
6	Dirección del establecimiento Matriz	Alfanumérico	Máx. 300	Obligatorio
7	Dirección del establecimiento Emisor	Alfanumérico	Máx. 300	Opcional
8	Código del establecimiento Emisor	Numérico	3	Obligatorio
9	Código del punto de Emisión	Numérico	3	Obligatorio
10	Contribuyente especial (Número de resolución)	Numérico	Min 3 y máx. 5	Opcional
11	Obligatorio a llevar Contabilidad (opciones Sí y No)	De selección	2	Opcional
12	Logo del Emisor	Imagen	-	Opcional
13	Tipo de Ambiente	Numérico	1	Obligatorio
14	Tipo de Emisión	Numérico	1	Obligatorio

Fuente: (SRI, Ficha técnica comprobantes electrónicos esquema offline, 2020)

Tablas a considerar para completar los campos de los datos del emisor

Tabla 6

Tipo de comprobante

No	Nombre comprobante	Código	Requisito
1	Factura	01	Obligatorio
2	Liquidación de compra de bienes y prestación de servicios	03	Obligatorio
3	Nota de crédito	04	Obligatorio
4	Nota de débito	05	Obligatorio
5	Guía de remisión	06	Obligatorio
6	Comprobante de retención	07	Obligatorio

Fuente: (SRI, Ficha técnica comprobantes electrónicos esquema offline, 2020)

Tabla 7

Tipo de ambiente

No	Tipo de ambiente	Código	Requisito
1	Pruebas	1	Obligatorio
2	Producción	2	Obligatorio

Fuente: (SRI, Ficha técnica comprobantes electrónicos esquema offline, 2020)

Tabla 8

Tipo de emisión

No	Tipo de emisión	Código	Requisito
1	Emisión normal	1	Obligatorio

Fuente: (SRI, Ficha técnica comprobantes electrónicos esquema offline, 2020)

6. Revisar factura

Además, el usuario puede visualizar los datos de la factura después de generar el XML ya que se podrá modificar únicamente los datos del cliente (cédula, ruc o pasaporte, dirección, teléfono, correo electrónico, etc.). En caso de que un dato haya sido mal ingresado se procede al paso de “Corregir datos” en caso contrario se debe continuar agregando la firma electrónica.

7. Corregir datos

El usuario a través del sistema KV- Facturación edita los datos del cliente que puedan tener algún error, los cuales serán cambiados en la base de datos y se generará un nuevo XML.

8. Agregar firma electrónica

El formato de comprobantes electrónicos es XML, los cuales deben ser firmados electrónicamente para garantizar la integridad de datos.

La firma electrónica se considera un nodo más a añadir en documento XML, la cual obligatoriamente tiene que estar en el estándar XadES_BES, misma que se puede realizar con librerías, tales como MITyCLibXADES, MITyCLibTSA, MITyCLibAPI, MITyCLibOCSP y MITyCLibTrust, las mismas que el SRI utiliza para validar e incorporar la firma de cada comprobante. Esto quiere decir que constituirá un documento válido una vez que el SRI proceda con la autorización. (SRI, Ficha técnica comprobantes electrónicos esquema offline, 2020)

En la Tabla 9, se detallan las especificaciones técnicas relacionadas al estándar.

Tabla 9

Especificaciones técnicas del estándar XadES_BES

Descripción	Especificación
Estándar de firma	XadES_BES
Versión del esquema	1.3.2
Codificación	UTF-8
Tipo de firma	ENVELOPED

Fuente: (SRI, Ficha técnica comprobantes electrónicos esquema offline, 2020)

El nivel de seguridad en la firma electrónica está ejecutado sobre tres partes: la primera, todos los elementos o nodos que conforman el comprobante electrónico, la segunda parte son los

elementos de firma ubicados en el contenedor “SignedProperties” y por último el certificado digital con el que se ha firmado incluido en el elemento “KeyInfo”.

El algoritmo de encriptación utilizado es RSA-SHA1, el cual tiene 2048 bits de longitud de clave, según la recomendación técnica basada en el documento “Recommendation for Key Management, Part 1: General (Revised)” (Barker, Barker, Burr, Polk, & Smid, 2007) y el archivo de Intercambio de Información PKCS12 (extensión .p12.)

9. Enviar datos

El sistema enviará en formato XML a los servidores del SRI, los datos de la factura dividido por etiquetas donde constará los datos del usuario final (*Tabla 1*), del producto (*Tabla 2*), el emisor (*Tabla 5*) y por última la firma electrónica según especificaciones de la *Tabla 9*.

10. Validar:

El SRI al momento de recibir el archivo en formato XML realiza seis validaciones, según el orden especificado, es importante resaltar que las validaciones son lineales; es decir, si el sistema detecta que el tamaño del archivo no es el establecido, ese será el error remitido y no continuará con el resto de las validaciones.

Tabla 10

Validaciones realizadas por el SRI

Orden	Validación	Descripción
1	Validación XML	<ul style="list-style-type: none"> ✓ Tamaño archivo ✓ Esquema archivo ✓ XML bien formado y válido
2	Validación contribuyente emisor	<ul style="list-style-type: none"> ✓ RUC activo ✓ Establecimiento activo ✓ Autorización para emitir comprobantes electrónicos activa. ✓ Autorización para emisión del tipo de comprobante

3	Validación unicidad	✓ Clave acceso única ✓ Secuencial único ✓ Clave acceso bien formada
4	Validación firma	✓ Validez firma y cadena de confianza OSCP
5	Validaciones adicionales	✓ Fecha de emisión ✓ Identificación del receptor del comprobante ✓ Documentos de sustento
6	Otras	

Fuente: (SRI, Ficha técnica comprobantes electrónicos esquema offline, 2020)

A fin de garantizar que la información transmitida a la base de datos del SRI cumpla con los requisitos establecidos en la normativa tributaria y comercio electrónico, se plantea las siguientes validaciones, las cuales deben ser implementadas en el sistema de facturación.

Tabla 11

Errores emitidos del SRI en el sistema de facturación electrónica

Código de error	Descripción	Posible solución	Validación: Recepción /Autorización /Emisor
34	Comprobante no autorizado	Cuando el comprobante no ha sido autorizado como parte de la solicitud de emisión del contribuyente	Emisor
42	Certificado revocado	Certificado que ha superado su fecha de caducidad, y no ha sido renovado.	Emisor
52	Error en diferencias	Cuando existe error en los cálculos del comprobante.	Emisor
64	Código documento sustento	Cuando el código del documento sustento no existe en el catálogo de documentos que se tiene en la Administración	Emisor
65	Fecha de emisión extemporánea	Cuando el comprobante emitido no fue enviado de acuerdo al tiempo del tipo de emisión en el cual fue realizado.	Emisor/ Recepción
69	Identificación del receptor	Cuando la identificación asociada al adquirente no existe. En general cuando el RUC del adquirente no existe en el Registro Único de Contribuyentes.	Emisor

Fuente: (SRI, Ficha técnica comprobantes electrónicos esquema offline, 2020)

Si las validaciones realizadas son correctas el SRI retorna el XML al sistema de facturación con su respectiva autorización y carga la factura en el sistema de SRI en línea del cliente, caso contrario, el SRI notifica el error, para posteriormente continuar con la siguiente tarea “Corregir XML”.

11. Notificar el error

En esta etapa, el SRI notifica el error al sistema KV – Facturación de acuerdo a las validaciones, presentada en la Tabla 10 y códigos de errores presentadas en la Tabla 11. Un punto que merece resaltar es que, si el sistema de facturación se encuentra correctamente automatizado para el proceso, los errores más importantes a tener en cuenta son las validaciones del contribuyente emisor; como, por ejemplo, que se encuentre el RUC y establecimiento activo con sus respectivas autorizaciones para la emisión de comprobantes electrónicos, y las validaciones de la firma electrónica, es decir, se valida su vigencia y su autenticidad.

12. Corregir XML

En esta etapa, de acuerdo al error presentado al sistema se corrige el XML, tomando en cuenta las consideraciones presentadas en la tarea “Generar XML”.

13. Enviar factura

Finalmente, en la última etapa el sistema KV-Facturación se encarga de enviar la factura electrónica mediante correo electrónico al cliente.

3.1.2. Análisis de Viabilidad

Analizar la viabilidad de un proyecto antes de emprenderlo suele ser una buena práctica para determinar qué tan probable es llevarlo a cabo, ya que se analiza la capacidad que existe para el desarrollo del mismo (Quintana, 2021). En el presente tema se pretende determinar los aspectos más importantes que se deben considerar en el desarrollo de un proyecto de software, como son: viabilidad técnica, económica y operativa.

El tiempo estimado de desarrollo del proyecto es de seis meses.

3.1.2.1. Viabilidad técnica

La viabilidad técnica analiza e identifica en detalle los recursos técnicos necesarios, como los requisitos en hardware, software y tecnología, los cuales se requieren para el desarrollo de un proyecto técnico. Para el desarrollo del sistema KV – Facturación, se determinaron los siguientes recursos:

- **Recursos de Hardware**

Tabla 12

Viabilidad técnica hardware

Cantidad	Descripción
1	Ordenador Portátil HP Core i5, 8Gb RAM, 1T Disco duro

Nota: Viabilidad técnica hardware. Elaborado por: Karla Moyón, autora del presente proyecto

- **Recursos de Software**

Tabla 13

Viabilidad técnica software

Recurso	Descripción
Sistema operativo (ordenador)	Windows 10
Sistema operativo (servidor)	Ubuntu 20.04
PostgreSQL	Base de datos
Python	Lenguaje de programación
Django	Framework de Python
Visual Studio Code	Editor de código
Postman	Cliente de servicio REST

Nota: Viabilidad técnica software. Elaborado por: Karla Moyón, autora del presente proyecto

De acuerdo al análisis de viabilidad técnica de los recursos tanto de software como hardware se puede concluir que es viable para la construcción del sistema de facturación electrónica, ya que se cuenta con los recursos necesarios para satisfacer los requerimientos del proyecto técnico, sin la necesidad de comprar o repotenciar algún recurso. Además, que la autora del proyecto cuenta con todas las destrezas técnicas para el manejo de los mismos.

3.1.2.2.Viabilidad económica

El análisis de viabilidad económica se enfoca en analizar los costes de software, hardware y otros, además de los beneficios del plan propuesto. En otras palabras, se refiere a los recursos financieros que permitirán la construcción del sistema y adquirir los elementos que se necesitan.

Para evaluar la inversión del proyecto, hay que considerar criterios que sean confiables, además de un manejo ágil y preciso para toma de decisiones o mejorar procesos productivos. La tabla presentada a continuación detalla los gastos en: hardware, software, servicios básicos, alimentación y costos de operación.

Tabla 14

Presupuesto de inversión

Hardware					
Cant	Detalle	C.Unitario	T. Meses	Total	Observación
1	Ordenador portátil HP Core i5, 8 GB RAM, 1 T Disco Duro	\$700.00		\$ 700.00	Permanente
Total Hardware:				\$ 700.00	
Software					
Cant	Detalle	C.Unitario	T. Meses	Total	Observación
1	WooCommerce	\$0.00	6	\$0.00	Capa Gratuita
1	Visual Studio Code	\$0.00	6	\$0.00	Open Source
1	GitHub	\$0.00	6	\$0.00	Open Source
1	MySQLWorkbench	\$0.00	6	\$0.00	Open Source
1	Django	\$0.00	6	\$0.00	Open Source
1	Amazon Web Service	\$0.00	6	\$0.00	Capa Gratuita
Total Software:				\$0.00	
Servicios y alimentación					
Cant	Detalle	C.Unitario	T. Meses	Total	Observación
1	Servicio de agua potable	\$10.00	6	\$60.00	
1	Servicio de Luz eléctrica	\$25.00	6	\$150.00	
1	Transporte	\$20.00	6	\$120.00	
1	Alimentación	\$100.00	6	\$600.00	
1	Servicio de Internet y Telefonía	\$48.00	6	\$288.00	
1	Arriendo	\$150.00	6	\$900.00	
Total servicios y alimentación:				\$2,118.00	
Costos de Operación					
Cant	Detalle	C.Unitario	T. Meses	Total	Observación
1	Firma electrónica (Security Data)	\$22.40		\$22.40	1 año de vigencia
Total costos de operación:				\$22.40	
PRESUPUESTO TOTAL :		\$2,840.40			

Nota: Presupuesto de la inversión en la elaboración del proyecto. Elaborado por: Karla Moyón, autora del presente proyecto.

El presupuesto total estimado para la construcción del sistema KV – Facturación tiene una estimación de \$2840.40.

3.1.2.3. Viabilidad operacional

Las tiendas virtuales se han convertido en un gran éxito mundial para los negocios gracias a la apertura y facilidad de acceso a Internet. Por otra parte, la facturación electrónica ha ganado mucha importancia debido a que permite una automatización y optimización de los procesos administrativos y contables que reducen los errores humanos y costos operacionales.

Mediante el sistema de facturación electrónica que se propone en el presente proyecto, se pretende disminuir el proceso manual de facturación, además de ofrecer otras ventajas como mejorar la velocidad, facilidad y seguridad al momento de realizar procesos contables con el SRI. Finalmente brindar la disponibilidad del servicio 24/7 mejorando la operatividad de los negocios.

Durante el proceso del análisis de la viabilidad del proyecto se puede concluir que:

- ✓ Existen los recursos tecnológicos necesarios para la construcción del sistema de facturación.
- ✓ De acuerdo al análisis económico, cabe mencionar que los gastos serán asumidos por la autora del sistema, por lo tanto, se considera que el proyecto es sostenible y rentable económicamente.
- ✓ Desde el punto de vista operacional, este sistema viene a cubrir una necesidad emergente de negocios que están empezando con el comercio electrónico, para brindar un servicio de facturación electrónica, mejorando el proceso y seguridad de los procesos contables.

3.1.3. Análisis de Requerimientos

El propósito de realizar dicho análisis es describir de forma detallada los puntos necesarios que se requieren para el diseño y construcción del sistema de facturación electrónica, KV – Facturación.

Para definir los requerimientos se ha utilizado la plantilla de “Especificación de requisitos de software”, conforme con el estándar IEEE Std 830-1998. (Especificación de requisitos de software, s.f.) A continuación, se mostrará el alcance del proyecto con la visión y descripción general del software, principalmente se especificará en detalle las operaciones que debe realizar el sistema.

3.1.3.1. Alcance

KV – Facturación propone la creación de un prototipo de una aplicación web que permita brindar un servicio de facturación electrónica, que se rija a las especificaciones y normativas técnicas determinadas por el SRI, para usuarios u organizaciones que cuenten con una tienda online creada con woocommerce.

Específicamente, el sistema permitirá a los usuarios obtener reportes de los pedidos realizados en la tienda virtual, los mismos que detallan su respectiva fecha, número y estado del pedido, para la generación de la factura electrónica.

Además, el sistema permite la visualización previa de la factura electrónica, para verificar los datos de la compra. En caso de que un dato del consumidor como cédula, nombre, teléfono, entre otros, estén incorrectos, el sistema permite la modificación de los datos. Finalmente se agrega la firma electrónica para su facturación.

Por último, el sistema detalla el total de ventas, total de clientes, consumo promedio y el total IVA de ventas que se ha realizado en la tienda virtual.

3.1.3.2. Definiciones, acrónimos y abreviaturas

- **SRI**

Servicio de Rentas Internas, institución que gestiona la política tributaria.

- **RUC (Registro Único de Contribuyentes)**

Sirve para realizar alguna actividad económica de forma permanente u ocasional en el Ecuador.

(SRI, REGISTRO ÚNICO DE CONTRIBUYENTES, s.f.)

- **Tienda virtual**

Sitio Web diseñado para vender productos u ofrecer servicios mediante Internet.

- **Usuario**

Persona que utiliza el sistema de facturación electrónica, quien posee una tienda virtual creada con WooCommerce, es decir, el administrador.

- **Consumidor**

Persona que adquiere productos o servicios de la tienda virtual del usuario.

- **WooCommerce**

Plugin de WordPress que permite implementar una tienda virtual totalmente funcional en la web.

3.1.3.3.Descripción general

El presente subcapítulo, ofrece un resumen de la funcionalidad del sistema, donde se describen los requisitos informales y establece un contexto para determinar los requerimientos del software.

Perspectivas del producto

El sistema de facturación electrónica que se propone depende de la conexión a una tienda virtual realizada con WordPress. La petición que realiza el sistema para obtener los pedidos de las compras, se realiza mediante la API que nos ofrece el plugin de WooCommerce. Los datos obtenidos mediante el servicio REST serán almacenados en una base de datos y mostrados mediante un servidor de aplicaciones para su funcionamiento. Finalmente, mediante una API se enviará los datos de la compra en formato XML al SRI para sus respectivas validaciones.

El fin del proyecto es construir un sistema web de facturación electrónica, en el cual los usuarios accederán a través de un navegador Web

Funciones del producto

El sistema se lo desarrollará como una aplicación Web, los usuarios accederán a través de un proceso de autenticación y dispondrá de los siguientes módulos:

- Mis facturas: Gestiona el proceso de facturación electrónica, es decir, mediante este módulo se puede realizar la actualización de los pedidos existentes en la tienda virtual. Además, el sistema permite la búsqueda de cualquier campo disponible en la tabla donde se despliega los pedidos como, por ejemplo: fecha, estado de la compra (Completado, Cancelado, En Espera, entre otros), estado de la factura (Autorizado, En procesamiento y No Autorizado), entre otros

Por otra parte, también permite la consulta de los datos del pedido, en donde el sistema mostrará los datos del consumidor y producto comprado. Así mismo, permite la visualización de los datos de la compra en forma de factura. Cabe mencionar que el sistema permite editar los datos del consumidor, en caso que se requiera cambiar, por motivos de error. Finalmente, el sistema permite facturar electrónicamente.

- Reporte General: En este módulo permitirá generar un reporte general de las facturas emitidas, mostrando el total en ventas, donde se desglosará el subtotal y el IVA 12%, con su respectivo número de transacciones. Productos más vendidos, donde se indicará los cinco primeros productos que tengan popularidad.

Además, el sistema indicará el Total de clientes, consumo promedio y el total IVA de ventas, considerando los últimos 30 días.

- Mis datos: Se podrá visualizar la información general del usuario, como nombre, correo electrónico, teléfono y los datos del RUC con el cual está ofreciendo los productos en la tienda virtual

Características del usuario

Se ha definido un solo tipo de usuario en función de las actividades que pueden realizar dentro del sistema.

Tipo de usuario	Administrador
Formación	Conocimiento de los productos o servicios de la tienda virtual.
Habilidades	Conocimiento básico de manejo de sistemas.
Actividades	Administrar y utiliza el sistema KV – Facturación.

Restricciones

El sistema debe ser instalado en un servidor de aplicaciones, compatible con los gestores de base de datos más comunes del mercado y su acceso será a través de un navegador web. KV – Facturación se desarrollará mediante software libre, por lo tanto, no tiene costo por el uso de las siguientes herramientas: sistema de base de datos, PostgreSQL, Python junto al framework de Django, y finalmente, el servidor AWS (Amazon Web Services) en la capa gratuita.

- **Integración de aplicaciones**

El sistema KV-Facturación se integra a la tienda virtual a través de la API que proporciona WooCommerce.

Suposiciones y dependencias

Para el funcionamiento completo del sistema, se requiere que el usuario utilice navegadores modernos como Google Chrome, Firefox o Brave. Además, tener buena conexión a Internet y finalmente un sistema operativo Windows 8 o superior.

Requisitos para futuras versiones del sistema

Se podría pensar en generar alertas tempranas que notifique cuando la firma electrónica esté a punto de vencer su funcionamiento. Así mismo, implementar interfaces para que el usuario configure su información tributaria, ya que esta versión solo podrá consultar los datos guardados en la base de datos. Finalmente, se podría implementar una interfaz para que el usuario que quiera utilizar el sistema KV – Facturación se registre, ingresando sus propios

datos, ya que la versión actual el usuario administrador es quien guarda los datos del dueño de la tienda virtual.

3.1.3.4.Requerimientos específicos

La presente sección es la más importante y extensa del documento, debido a que el nivel de detalle de los requisitos debe ser autosuficiente para el desarrollo, diseño y construcción de un sistema. Al momento de realizar las pruebas se deberá determinar si se ha cumplido cada requerimiento planteado.

A continuación, se detalla los requisitos comunes de las interfaces:

- **Interfaces de usuario**

Las interfaces de usuario serán Web, deber ser intuitivas y dinámicas para que la interacción de los usuarios sea lo más comprensible posible, sin la necesidad de utilizar un manual. Además, es necesario que se pueda identificar rápidamente las secciones y funcionamientos del sistema.

- **Interfaces de Hardware**

Pantalla del monitor: Lugar por donde el software deberá mostrar información al usuario. La resolución mínima que se necesita es 1024 x 768.

Mouse (ratón): El usuario debe interactuar con el sistema mediante el movimiento del ratón para la navegación dentro de la aplicación web.

Teclado: El usuario debe interactuar con el sistema mediante las pulsaciones de cada tecla, ya que servirá para la entrada de datos al sistema. Ejemplo: ingreso de credenciales, modificación de datos, etc.

- **Interfaces de Software**

El sistema debe permitir la comunicación con la tienda virtual creada con WooCommerce, para la obtención de los pedidos. KV – Facturación recibirá los datos mediante un archivo en formato

Json. Por otro lado, también es necesario la comunicación del sistema con el SRI para el proceso de facturación, para lo cual el envío y recepción de los datos será mediante un archivo en formato XML. Para ello, se requiere de una interfaz de programación de aplicaciones(API) para poder realizar la integración respectiva.

- **Interfaces de Comunicación**

El sistema debe trabajar utilizando el protocolo HTTP.

3.1.3.5.Requerimientos Funcionales

Para la creación de los requerimientos funcionales, se considerará cada interfaz del sistema.

- **Inicio de sesión**

Requerimiento funcional 01: Autenticación de usuarios

Descripción	El sistema permitirá al usuario iniciar sesión por medio de sus credenciales
Precondición	<ul style="list-style-type: none"> ✓ Los datos del usuario deben estar guardados en la base del sistema KV – Facturación. ✓ Tener una tienda virtual creada con WooCommerce.
Entrada	Credenciales del usuario (correo electrónico y contraseña)
Proceso	Validación de credenciales
Salida	Interfaz “Mis facturas” / Error
Prioridad	Alta

- **Mis facturas**

Requerimiento funcional 02: Visualizar datos generales para la facturación

Descripción	El sistema permitirá al usuario visualizar los datos generales para la facturación.
Precondición	Inicio de sesión del usuario
Entrada	Ninguna

Proceso	Consulta de pedidos a la tienda virtual mediante la API que ofrece WooCommerce.
Salida	Tabla datos generales, considerando: <ul style="list-style-type: none"> ✓ Fecha ✓ Nombre consumidor (Cliente) ✓ Número del pedido ✓ Valor del Pedido ✓ Estado de la compra (Completado, Cancelado, En Espera, entre otros) ✓ Vista previa, donde se pueda visualizar o editar los datos de la factura ✓ Estado de la factura (Autorizado, No Autorizado, En procesamiento)
Prioridad	Alta

Requerimiento funcional 03: Actualización de pedidos

Descripción	El sistema permitirá al usuario actualizar los pedidos realizados en la tienda virtual.
Precondición	✓ Inicio de sesión del usuario
Entrada	Ninguna
Proceso	Actualización de pedidos, considerando la última actualización registrada en la base de datos
Salida	Tabla actualizada de los pedidos, considerando: <ul style="list-style-type: none"> ✓ Nuevos pedidos ✓ Estado de la compra ✓ Estado factura
Prioridad	Alta

Requerimiento funcional 04: Visualizar los datos de la factura electrónica

Descripción	El sistema permitirá al usuario visualizar los datos de la factura electrónica.
Precondición	✓ Inicio de sesión del usuario

Entrada	Ninguna
Proceso	Considerando la tabla donde se indica los datos generales, específicamente la columna de acciones, el usuario puede seleccionar la opción de visualizar, mediante un clic.
Salida	<p>Interfaz, donde se indique los datos para la facturación electrónica, considerando:</p> <ul style="list-style-type: none"> ✓ Fecha ✓ Serie de la factura ✓ Autorización ✓ Datos del usuario (RUC, Nombre comercial, dirección, teléfono, entre otros) ✓ Datos del consumidor (Nombre, cédula, dirección, teléfono y correo electrónico) ✓ Datos producto (nombre producto, descripción, cantidad y precio) ✓ Opción de facturar
Prioridad	Alta

Requerimiento funcional 05: Editar los datos de la factura electrónica

Descripción	El sistema permitirá al usuario editar los datos del cliente para la factura electrónica.
Precondición	✓ Inicio de sesión del usuario
Entrada	<ul style="list-style-type: none"> ✓ Número de Cédula o RUC ✓ Nombres y apellidos ✓ Correo electrónico ✓ Dirección ✓ Teléfono
Proceso	Considerando la pantalla donde se visualiza los datos de la factura, específicamente el campo de datos del consumidor, el usuario puede seleccionar la opción de edición, mediante un clic.
Salida	Interfaz, donde se indique los datos para la facturación electrónica, pero con campos de edición en los datos del consumidor, considerando los campos de entrada.
Prioridad	Alta

Requerimiento funcional 06: Búsquedas para filtrar pedidos

Descripción	El sistema permitirá al usuario realizar búsquedas para filtrar pedidos de la tabla de los datos generales para la facturación.
Precondición	✓ Inicio de sesión del usuario
Entrada	Según sea la elección de búsqueda por parte del usuario.
Proceso	Consulta de los pedidos de acuerdo a la búsqueda realizada por el usuario.
Salida	<ul style="list-style-type: none"> ✓ Tabla de los datos generales para la facturación, según la búsqueda realizada por el usuario ✓ Mensaje, en caso de no encontrar pedidos de acuerdo a la búsqueda.
Prioridad	Alta

Requerimiento funcional 07: Facturar pedido

Descripción	El sistema permitirá al usuario facturar electrónicamente la compra realizada por el consumidor.
Precondición	<ul style="list-style-type: none"> ✓ Inicio de sesión del usuario ✓ Disponer de firma electrónica. ✓ Enviar solicitud de autorización en línea de comprobantes electrónicos en el SRI.
Entrada	Contraseña de la firma electrónica
Proceso	Al seleccionar la opción de que se desea facturar, el sistema indicará un mensaje para la confirmación. Al aceptar el usuario debe ingresar la contraseña para agregar al XML la firma electrónica. (Ver <i>Figura 8 Análisis del proceso de facturación electrónica</i>)
Salida	Mensaje de aceptación / Error.
Prioridad	Alta

• Reporte General

Requerimiento funcional 08: Generar reporte general

Descripción	El sistema generará un reporte general de los pedidos realizados en la tienda virtual.
Precondición	✓ Inicio de sesión del usuario
Entrada	Ninguna
Proceso	Generación de consultas del reporte general.
Salida	<p>Interfaz de Reporte general, considerando los últimos 30 días:</p> <ul style="list-style-type: none"> ✓ Total en ventas, se desglosa el subtotal y el IVA, indica el número de transacciones. ✓ Productos más vendidos. ✓ Total de clientes ✓ Consumo promedio ✓ Total de impuestos
Prioridad	Alta

- **Mis datos**

Requerimiento funcional 09: Consultar datos usuario

Descripción	El sistema mostrará los datos del usuario que han sido almacenados en la base de datos.
Precondición	<ul style="list-style-type: none"> ✓ Inicio de sesión del usuario ✓ Los datos del usuario deben estar guardados en la base del sistema KV – Facturación.
Entrada	Ninguna
Proceso	Consulta a la base de datos de la información del usuario.
Salida	<p>Interfaz “Mis datos”:</p> <ul style="list-style-type: none"> ✓ RUC ✓ Nombre Comercial ✓ Autorización SRI ✓ Nombres y Apellidos ✓ Correo electrónico ✓ Dirección

	✓ Teléfono ✓ Fecha de emisión y vencimiento de la firma electrónica ✓ Archivo de la firma electrónica
Prioridad	Alta

3.1.3.6.Requerimientos No Funcionales

- **Requisitos de rendimiento**

1. El sistema deberá estar disponible 24 horas al día y 7 días a la semana (24/7)

- **Seguridad**

2. El sistema deberá guardar la contraseña encriptada del usuario en la base de datos. Por otro lado, la contraseña de la firma electrónica no se almacena.
3. El sistema debe proveer mecanismos de seguridad para que solo usuarios autorizados ingresen.

- **Usabilidad**

4. El sistema debe tener un diseño de interfaces dinámicas e intuitivas para el usuario.
5. El sistema garantizará una adecuada visualización de las interfaces de usuario, es decir, debe contar con un diseño responsivo.

3.1.3.7.Restricciones de diseño

Las herramientas y plataformas que se utilice para la construcción de la aplicación web de facturación electrónica deberán ser software libre.

3.2. Diseño

En el presente subcapítulo se mostrará las funcionalidades más relevantes que los usuarios podrán realizar en el sistema KV – Facturación. Por una parte, se muestra de manera visual los prototipos de las interfaces del usuario. De manera análoga, las acciones serán representadas mediante los diagramas de casos de uso y secuencia. Finalmente, el diseño para la construcción del sistema será representado por los diagramas de clases y base de datos.

3.2.1. Interfaces de usuario

La aplicación web de facturación electrónica será intuitiva, funcional, dinámica y especialmente diseñada para un solo actor que será el Administrador del sistema.

A continuación, se mostrará el diseño que será considerado para la construcción de la aplicación, con su respectivo detalle del funcionamiento.

- **Inicio de Sesión**

La primera pantalla será para el inicio de sesión del usuario, cuyo objetivo es permitir al administrador ingresar sus credenciales para acceder al sistema de facturación electrónica.

Figura 10

Diseño Prototipo Inicio de Sesión



Elaborado por: Karla Moyón, autora del presente proyecto

- **Mis facturas**

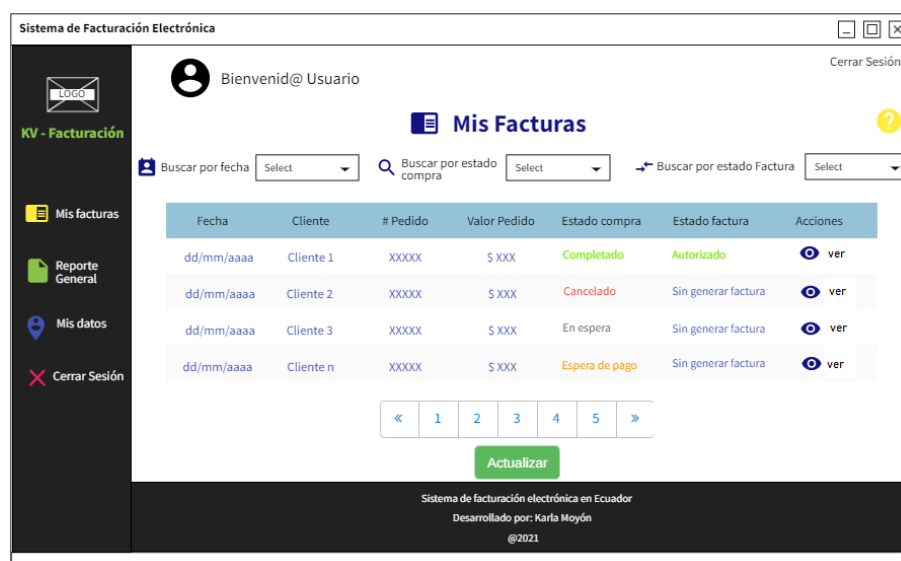
Dentro de esta interfaz se muestra el detalle de los pedidos realizados a la tienda virtual. Uno de los aspectos más destacables es el estado de la compra, ya que únicamente se podrá facturar los pedidos que se encuentren como “Completado”. Por otro lado, se tiene los estados de la factura, los cuales se mencionan en el apartado de funciones del producto, dentro del análisis de requerimientos. Los estados mencionados anteriormente se deberán diferenciar por el color de la fuente.

Además, en esta pantalla se puede buscar información mediante filtros, ya sea que se desee encontrar pedidos por fecha o según el estado de la compra o factura.

Finalmente, el usuario tiene dos diferentes acciones: visualizar y facturar.

Figura 11

Diseño Prototipo “Mis facturas”



Elaborado por: Karla Moyón, autora del presente proyecto

Acción “Visualizar”

Al seleccionar la presente acción el sistema indicará la siguiente pantalla, donde se muestra la información de la factura, información del emisor, datos del consumidor, detalle de la compra, desglose de valores y la información adicional.

En el apartado de datos de consumidor, mediante un botón, el usuario tiene la opción de editar los datos, como, por ejemplo: el número de cédula o RUC, nombre, dirección, etc. Sin embargo, si el pedido ya fue facturado, el sistema mostrará un mensaje, indicando que no se puede editar.

Figura 12

Diseño Prototipo Vista previa de la factura electrónica

The screenshot shows a web application titled "Sistema de Facturación Electrónica". It features a dark sidebar on the left with icons and labels for navigation: "KV - Facturación", "Mis facturas", "Reporte General", "Mis datos", and "Cerrar Sesión". The main content area has a light blue header with a message bar. Below this, there's a form for "Información Emisor" and "Información Factura". The form includes fields for "Razón Social / Nombres y apellidos", "RUC / C.I.", "Fecha de emisión", "Dirección", and "Teléfono". A green "Actualizar" button is positioned below the form. Below the form is a table with columns: "Cod Producto", "Cantidad", "Descripción", "Precio unitario", "Descuento", and "Precio Total". The table contains one row of placeholder data. Below the table is a section for "Información Adicional" and a summary table with rows for "SubTotal 12%", "IVA 12%", and "Valor Total". At the bottom of the main area are two buttons: "Regresar" (orange) and "Facturar" (blue). The footer contains the text "Sistema de facturación electrónica en Ecuador", "Desarrollado por: Karla Moyón", and "@2021".

Elaborado por: Karla Moyón, autora del presente proyecto

Opción de Actualizar

Al seleccionar dicha opción, el sistema abrirá una nueva ventana donde los datos del consumidor se convierten en campos de entrada, donde el usuario podrá modificar. Al dar clic en el botón "Actualizar". El sistema almacena automáticamente en la base de datos.

Acción "Facturar"

Al seleccionar la opción facturar, el sistema internamente validará los siguientes puntos:

- ✓ El estado de la compra debe ser "completado", de ser el caso mostrará una ventana para confirmar la acción ingresando la contraseña de la firma electrónica. Teniendo como

opciones: “Aceptar”, donde se procederá a agregar la firma al XML y enviar al SRI para su validación, y “Cancelar” (Ver Figura 13). Caso contrario, se indicará un mensaje que no se puede realizar la acción.

- ✓ Cuando un pedido ya es facturado, no se podrá visualizar el botón “Facturar”

Figura 13

Diseño Prototipo Confirmar facturación

Confirmar facturación

Agregar contraseña firma electrónica

Cancelar Aceptar

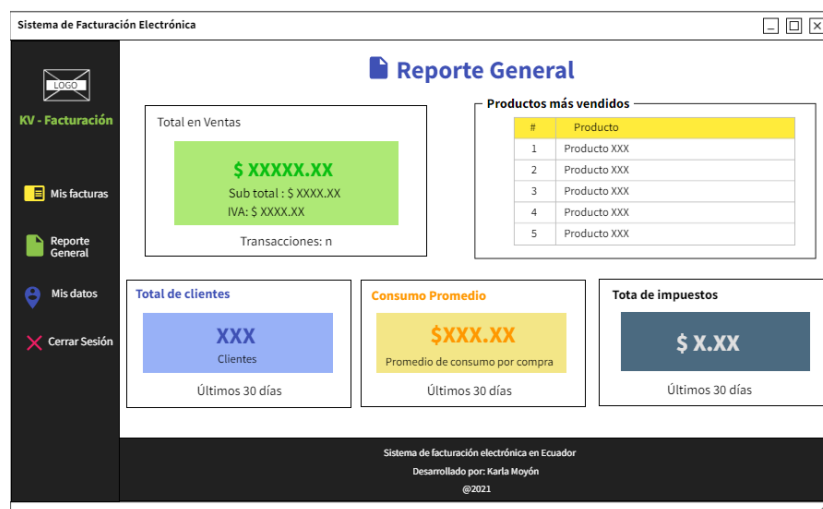
Elaborado por: Karla Moyón, autora del presente proyecto

- **Reporte General**

Se requiere que el usuario pueda observar un reporte general con los datos de los últimos 30 días, indicando mediante tarjetas cada información de: total de ventas, productos más vendidos, total de clientes, consumo promedio y total IVA de ventas.

Figura 14

Diseño Prototipo Reporte General



Elaborado por: Karla Moyón, autora del presente proyecto

- **Mis datos**

Finalmente, la pantalla “Mis datos” mostrará la información general del usuario, la misma que ha sido almacenada en la base de datos.

Figura 15

Diseño Prototipo Mis datos



Elaborado por: Karla Moyón, autora del presente proyecto

3.2.2. Casos de uso

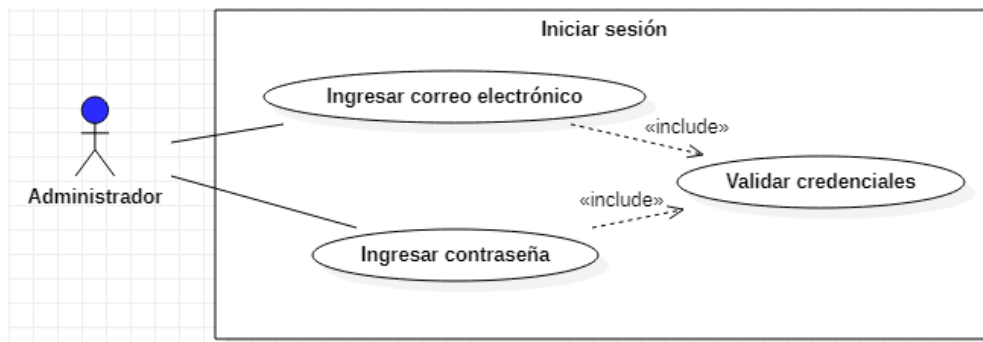
Los siguientes diagramas de caso de uso ayudan a visualizar de una mejor manera las funcionalidades de los principales módulos y actores que conforman, intervienen e interactúan con el software. Cabe mencionar que solo un usuario, el administrador, tiene acceso al sistema KV – Facturación.

3.2.2.1.Caso de uso 01: Inicio de sesión

Este módulo permitirá al administrador ingresar sus credenciales: correo electrónico y contraseña, para acceder al sistema informático, las mismas que serán validadas comprobando su autenticidad con los datos guardados en la base de datos del programa.

Figura 16

Caso de uso 01: Iniciar sesión



Elaborado por: Karla Moyón, autora del presente proyecto

3.2.2.2.Caso de uso 02: Menú

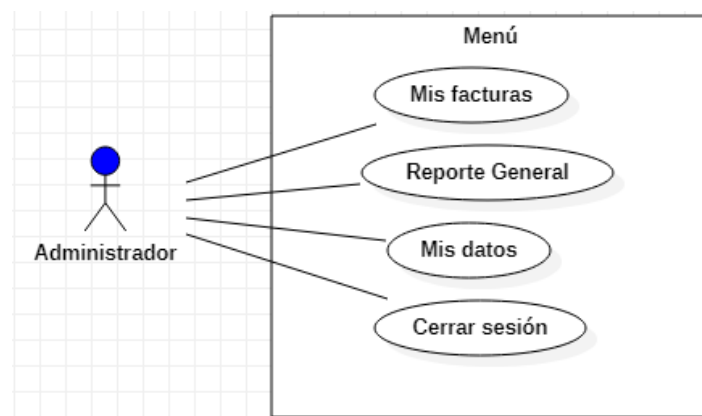
Este caso de uso representa todas las tareas que dispone el usuario, dentro del programa. Las funcionalidades a las que tiene acceso son relacionadas a la gestión de información que requiere el sistema de facturación.

El menú presentado al usuario, está dividido en tres módulos: Mis facturas, Reporte General, donde el usuario podrá visualizar de manera general estadísticas de las compras realizadas y Mis datos, en el cual se presenta la información del administrador.

Finalmente se encuentra la acción de Cerrar sesión, esto quiere decir que el usuario cierra el sistema KV – Facturación y termina con sus tareas.

Figura 17

Caso de uso 02: Menú



Elaborado por: Karla Moyón, autora del presente proyecto

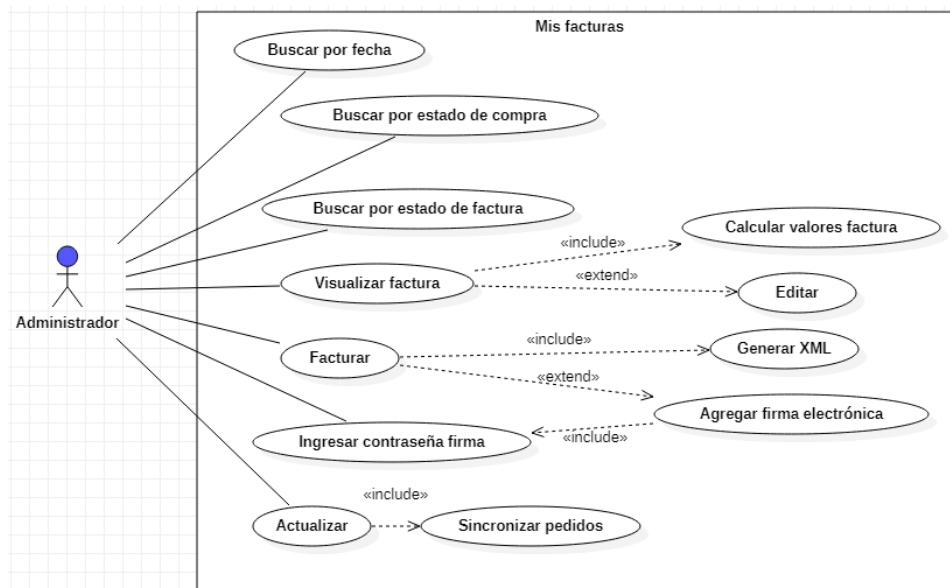
3.2.2.3.Caso de uso 03: Mis facturas

Este módulo permitirá al administrador realizar varias acciones con los pedidos obtenidos de la tienda virtual, entre ellos tenemos:

1. Realizar búsquedas por fecha, estado de compra o estado de la factura.
2. Visualizar factura, donde el sistema internamente calcula los valores de la factura, es decir, sub total, IVA, descuentos, etc. Por otro lado, el usuario podrá editar los datos del consumidor.
3. Facturar, el usuario al selecciona esta opción, el sistema genera un archivo en formato XML con los datos para la factura. Además, para que concluya exitosamente dicha acción, el administrador debe ingresar la contraseña de la firma electrónica, para que el programa agregue al archivo XML.
4. Actualizar, al seleccionar esta opción el sistema realiza una sincronización de los pedidos realizados anteriormente con los actuales. (Ver Figura 21).

Figura 18

Caso de uso 03: Mis facturas



Elaborado por: Karla Moyón, autora del presente proyecto

3.2.3. Diagrama de secuencia

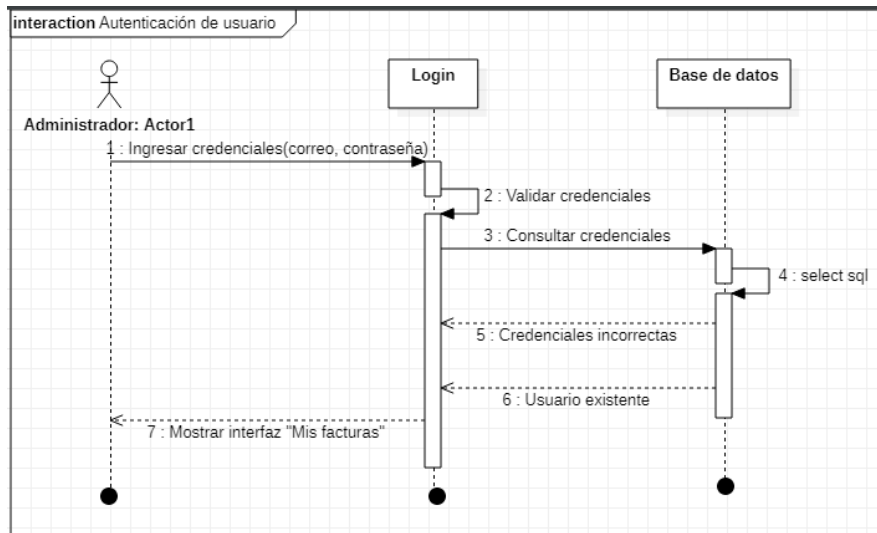
Los diagramas de secuencia se basan en los requerimientos funcionales planteados para la construcción del sistema de facturación electrónica. Además, permite visualizar la interacción y comunicación entre los diferentes objetos del software.

3.2.3.1. Autenticación de usuario

A continuación, se muestra la secuencia y los diferentes objetos que permiten al administrador acceder al sistema, para lo cual desde la interfaz de Login deben ingresar sus credenciales (correo electrónico y contraseña). Internamente se realiza una conexión a la base de datos para la validación de los campos ingresados. En caso de ser correctos, se mostrará la interfaz principal, “Mis facturas” y además tendrá acceso a los diferentes módulos del sistema. Caso contrario, se mostrará un mensaje indicando que las credenciales son incorrectas.

Figura 19

Diagrama de secuencia RF01: Autenticación de usuario



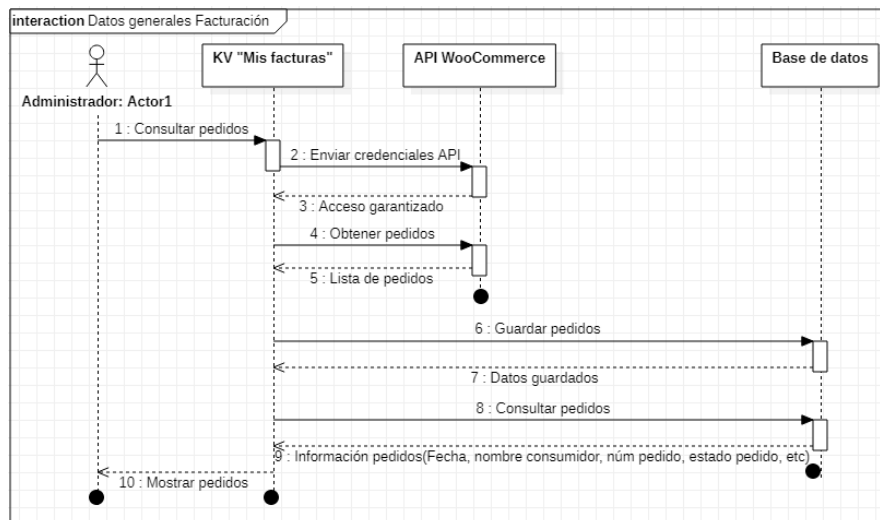
Elaborado por: Karla Moyón, autora del presente proyecto

3.2.3.2. Visualizar datos generales para la facturación

En la figura 20, se muestra la secuencia y los diferentes objetos que permiten al administrador visualizar los datos generales para la facturación. En primer lugar, el usuario puede consultar pedidos a través de la interfaz del sistema KV – Facturación “Mis facturas”, la misma que envía las credenciales para conectar con la API de WooCommerce y obtener los pedidos de la tienda virtual, los cuales serán recibidos en formato JSON. Luego el sistema guarda la información en la base de datos, con el fin de consultar el detalle de los pedidos (Fecha, nombre consumidos, número del pedido, estado de compra, etc.) y mostrar al usuario.

Figura 20

Diagrama de secuencia RF02: Visualizar datos generales para la facturación



Elaborado por: Karla Moyón, autora del presente proyecto

3.2.3.3.Actualización de pedidos

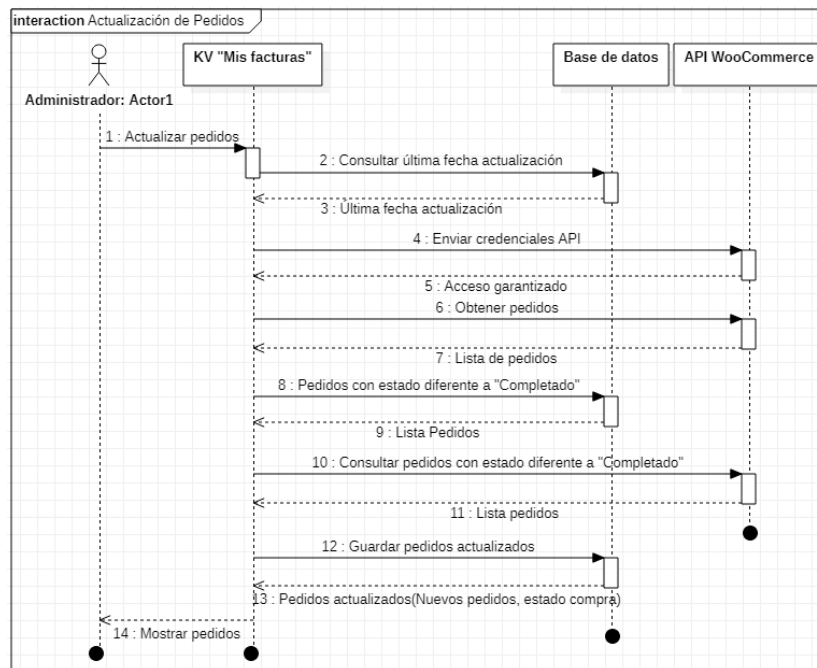
En la figura 21, se muestra la secuencia y los diferentes objetos que permiten al administrador actualizar los pedidos. Para empezar, el usuario solicita actualizar los pedidos a través de la interfaz de KV – Facturación “Mis facturas”, donde automáticamente el sistema consulta la última fecha de actualización y envía las credenciales para acceder a la API de WooCommerce con el fin de obtener los pedidos de la tienda virtual, sin considerar los pedidos anteriores a la fecha consultada.

Luego, el sistema realiza una consulta a la base de datos de los pedidos que tengan un estado diferente a “Completado”, con el fin de consultar a la API de WooCommerce por dichos pedidos y comprobar el estado de la compra, recordando que solo se podrán facturar los “Completado”.

KV – Facturación guarda los pedidos nuevos y actualizados en la base de datos para finalmente mostrar el listado al Administrador.

Figura 21

Diagrama de secuencia RF03: Actualización de pedidos



Elaborado por: Karla Moyón, autora del presente proyecto

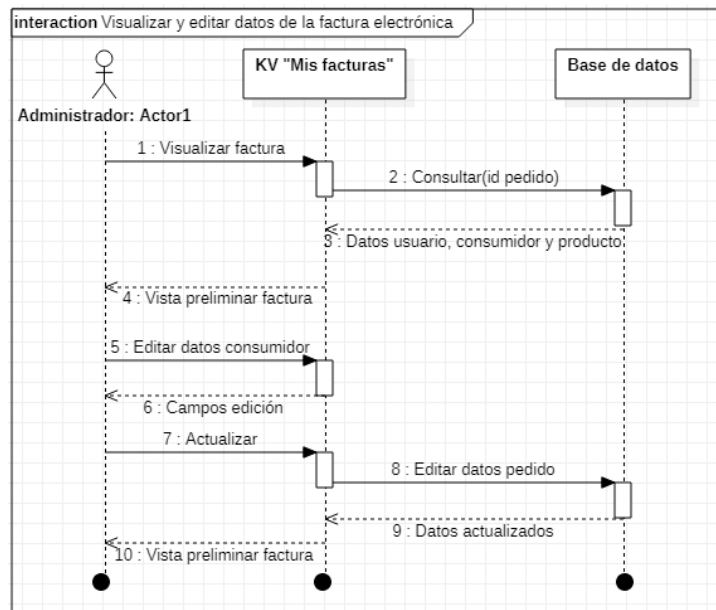
3.2.3.4. Visualizar los datos de la factura electrónica

En la figura 22, se muestra la secuencia y los diferentes objetos que permiten al administrador visualizar y editar los datos de la factura electrónica. En primer lugar, el usuario solicita visualizar la factura del pedido que desee a través de la interfaz “Mis facturas” de KV – Facturación.

El sistema realiza una consulta a la base de datos del pedido seleccionado y muestra al Administrador la vista preliminar de la factura, donde existe una opción de editar los datos del consumidor, de tal forma que, pueda modificar los campos que se requieran. Luego de la actualización, el sistema edita el pedido en la base de datos y nuevamente muestra al usuario la vista preliminar de la factura.

Figura 22

Diagrama de secuencia RF04 y RF05: Visualizar y editarlos datos de la factura electrónica



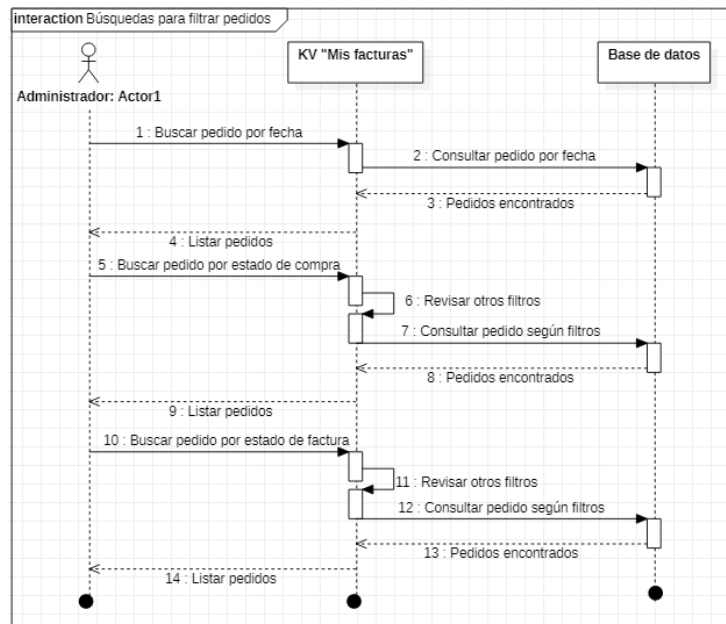
Elaborado por: Karla Moyón, autora del presente proyecto

3.2.3.5. Búsquedas para filtrar pedidos

En la figura 23, se muestra la secuencia y los diferentes objetos que permiten al administrador realizar búsquedas para filtrar pedidos. El usuario a través de la interfaz “Mis facturas” del sistema KV – Facturación tiene la opción de buscar pedidos por fecha, estado de compra y estado de factura, para lo cual, en cada búsqueda, el sistema consulta a la base de datos los pedidos de acuerdo al o los filtros seleccionados.

Figura 23

Diagrama de secuencia RF06: Búsquedas para filtrar pedidos



Elaborado por: Karla Moyón, autora del presente proyecto

3.2.3.6. Facturar pedido

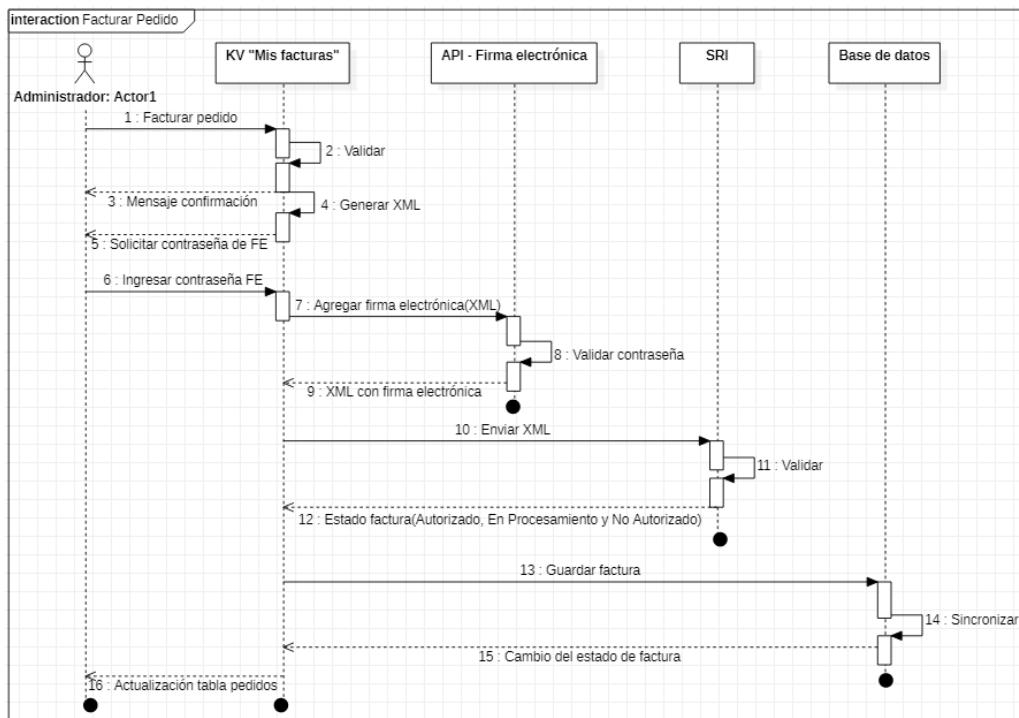
En la figura 24, se muestra la secuencia y los diferentes objetos que permiten al administrador facturar electrónicamente el pedido. En primer lugar, el usuario selecciona la acción de facturar pedido a través de la interfaz “Mis facturas” de KV – Facturación, donde el sistema valida automáticamente el estado de la compra, el cual debe ser “Completado”, y además que no se haya facturado para generar el archivo en formato XML.

El usuario deberá ingresar la contraseña de la firma electrónica, la misma que será validada con la API correspondiente y el sistema le agregue al final del XML. Como siguiente paso el KV – Facturación enviará dicho archivo al SRI, quien es el encargado de validar el formato y estructura, e indicará el estado de la factura (Autorizado, En procesamiento y No autorizado).

Finalmente, los datos recibidos y validados serán guardados en la base de datos para mostrar al Administrador la actualización, indicando que el pedido ha sido facturado.

Figura 24

Diagrama de secuencia RF07: Facturar pedido



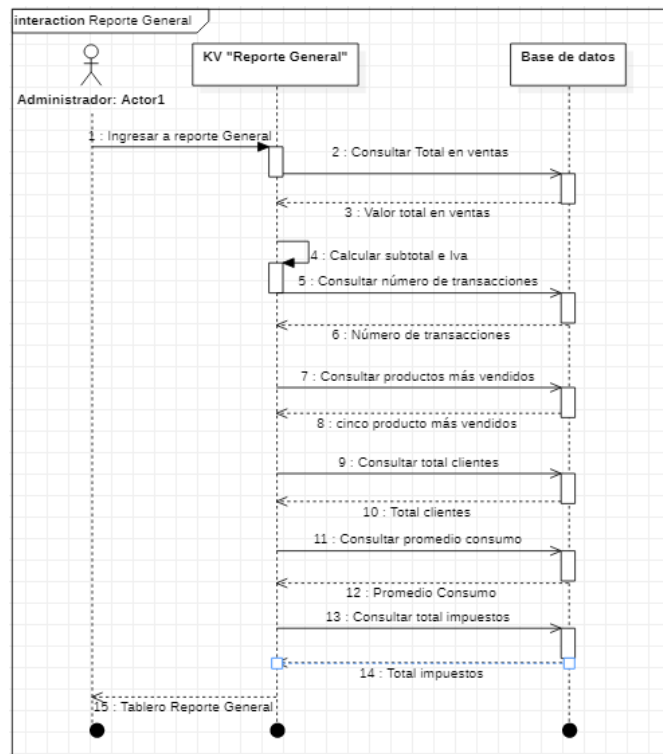
Elaborado por: Karla Moyón, autora del presente proyecto

3.2.3.7. Generar reporte general

En la figura 25, se muestra la secuencia y los diferentes objetos que permiten al administrador visualizar el reporte general de los pedidos. En primer lugar, el usuario ingresa al módulo de “Reporte General”, donde de manera asíncrona, el sistema consulta a la base de datos el total en ventas, donde se calcula el subtotal e IVA del resultado, número de transacciones, cinco de los productos más vendidos, total de clientes, promedio consumo y el promedio de compras registradas. Cabe recordar, que los datos mostrados son considerando los 30 últimos días. Finalmente, el reporte general es mostrado en forma de tablero.

Figura 25

Diagrama de secuencia RF08: Generar reporte general



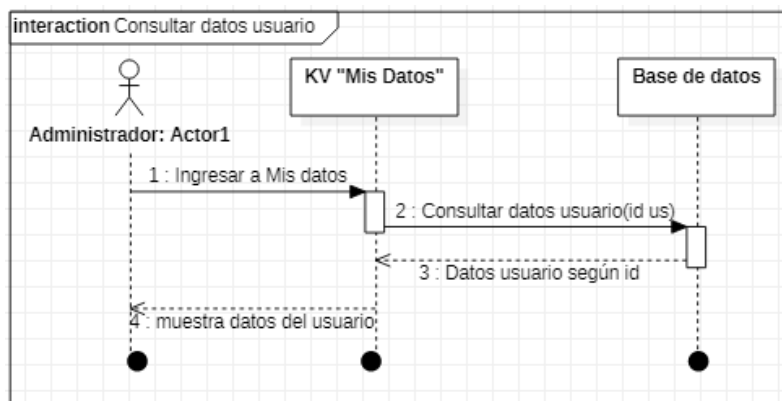
Elaborado por: Karla Moyón, autora del presente proyecto

3.2.3.8.Consultar datos usuario

En la figura 26, se muestra la secuencia y los diferentes objetos que permiten al administrador consultar sus datos. El usuario al ingresar al módulo “Mis datos” el sistema KV – Facturación consulta a la base de datos la información del Administrador y muestra a través de la interfaz.

Figura 26

Diagrama de secuencia RF09: Consultar datos usuario



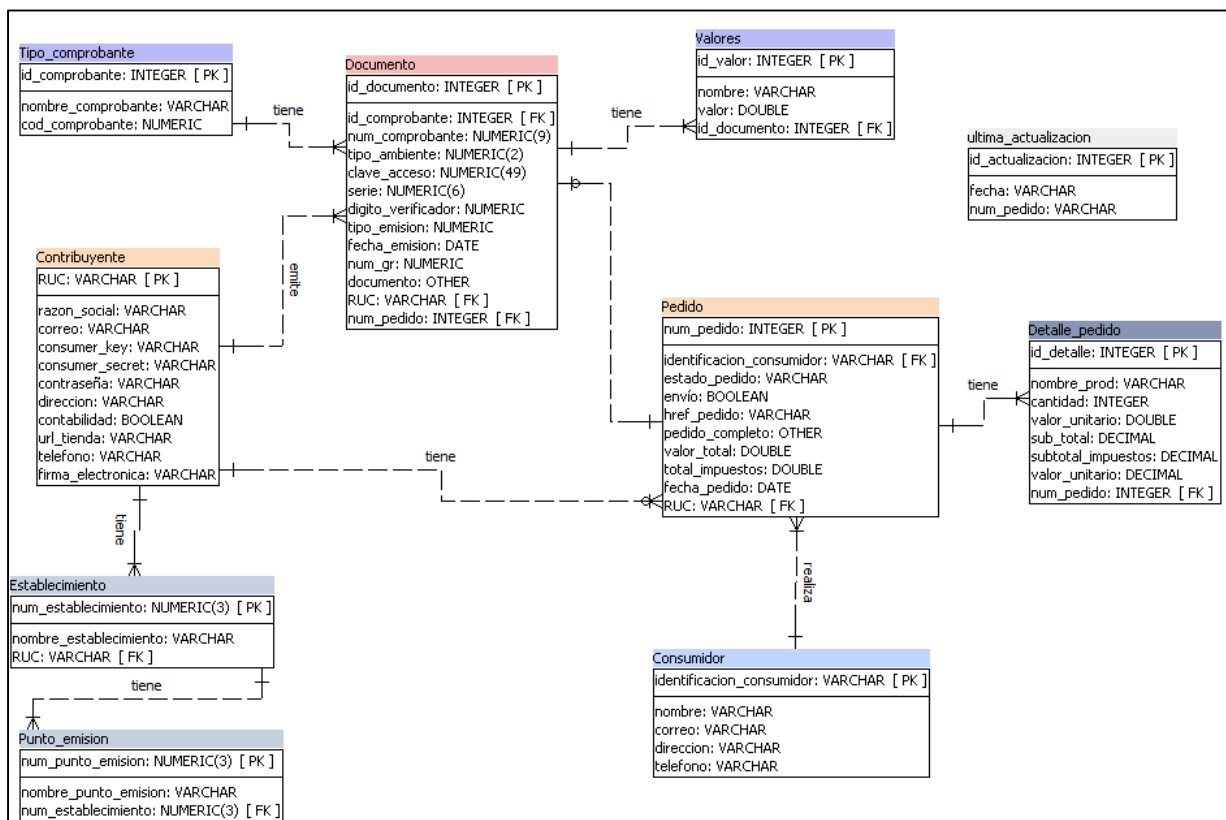
Elaborado por: Karla Moyón, autora del presente proyecto

3.2.4. Diagrama de base de datos

En el presente apartado se presenta el diagrama de base de datos, la cual ha sido diseñada en base a los requerimientos planteados. Se ha desarrollado el siguiente diagrama conceptual en donde consta las tablas necesarias para almacenar los datos relacionados a la emisión de una factura electrónica y sus respectivas relaciones.

Figura 27

Diagrama de base de datos



Elaborado por: Karla Moyón, autora del presente proyecto

A continuación, se detalla las tablas, claves y tipos de datos presentados en el diagrama de base de datos.

Tabla Contribuyente:

Esta tabla almacena información del Contribuyente, es decir, el usuario del sistema KV-Facturación, quien va a ser propietario de la tienda virtual.

Tabla 15

Tabla Contribuyente

Campo	PK	FK	Tipo	Descripción
RUC			Varchar(15)	Identifica al contribuyente con su respectiva actividad económica.
razon_social			Varchar(40)	Nombre del negocio del usuario, está ligado al RUC.
Correo			Varchar(100)	Correo electrónico del contribuyente.
consumer_key			Varchar(100)	Clave para acceder a la API de WooCommerce.
consumer_secret			Varchar(100)	Clave para acceder a la API de WooCommerce.
Contraseña			Varchar(16)	Contraseña para ingresar al sistema.
Dirección			Varchar(200)	Dirección del negocio del contribuyente.
Contabilidad			Boolean	Obligado a llevar contabilidad.
url_tienda			Varchar(300)	URL de la tienda virtual del contribuyente.
Teléfono			Varchar(10)	Teléfono del contribuyente
firma_electronica			File	Archivo .p12, el cual contiene la firma electrónica del contribuyente.

Elaborado por: Karla Moyón, autora del presente proyecto

Tabla Establecimiento

Almacena la información del establecimiento, donde se va a emitir las facturas electrónicas.

Se relaciona con la tabla de Contribuyente, ya que un contribuyente puede tener uno o varios establecimientos.

Tabla 16

Tabla Establecimiento

Campo	PK	FK	Tipo	Descripción
num_establecimiento			Numeric(3)	Número del establecimiento, donde se va a realizar la facturación

RUC			Varchar(15)	Identifica al contribuyente con su respectiva actividad económica.
nombre_establecimiento			Varchar(50)	Nombre del establecimiento

Elaborado por: Karla Moyón, autora del presente proyecto

Tabla Punto_emision

Almacena la información del punto de emisión, donde se emite la factura electrónica, relacionado con el establecimiento, ya que un establecimiento, puede tener varios puntos de emisión.

Tabla 17

Tabla Punto de emisión

Campo	PK	FK	Tipo	Descripción
num_punto_emision			Numeric(3)	Número del establecimiento, donde se va a realizar la facturación
num_establecimiento			Numeric(3)	Identifica al contribuyente con su respectiva actividad económica.
nombre_punto_emision			Varchar(50)	Nombre del establecimiento

Elaborado por: Karla Moyón, autora del presente proyecto

Tabla Consumidor

Almacena la información del consumidor, es decir, de la persona u organización que compra mediante la tienda virtual del contribuyente.

Tabla 18

Tabla Consumidor

Campo	PK	FK	Tipo	Descripción
identificacion_consumidor			Varchar(15)	RUC, cédula o pasaporte del consumidor.
Nombre			Varchar(500)	Nombre del consumidor.
Correo			Varchar(100)	Correo del consumidor.
Dirección			Varchar(300)	Dirección del consumidor.

Teléfono			Varchar(10)	Teléfono del consumidor.
----------	--	--	-------------	--------------------------

Elaborado por: Karla Moyón, autora del presente proyecto

Tabla Pedido

Almacena la información del pedido realizado por el consumidor mediante la tienda virtual. Se relaciona con la tabla del Contribuyente y Consumidor, ya que un contribuyente puede tener uno o varios pedidos realizados a su tienda virtual y a su vez un consumidor puede realizar uno o varios pedidos.

Tabla 18

Tabla Pedido

Campo	PK	FK	Tipo	Descripción
num_pedido			Integer	Número del pedido.
identificacion_consumidor			Varchar(15)	RUC, cédula o pasaporte del consumidor.
RUC			Varchar(15)	RUC del contribuyente.
estado_pedido			Varchar(20)	Estado del pedido (Completado, en espera, procesando, cancelado, etc.)
Envío			Boolean	Sirve para conocer si el consumidor realizó su pago con envío o no.
href_pedido			URLField	URL del pedido desde la API de WooCommerce.
valor_total			Double	Total del Subtotal de la compra.
total_impuestos			Double	Total de impuesto de la compra.
total_compra			Double	Suma de los campos: valor_total y total_impuestos.
fecha_pedido			Date	Fecha del pedido.

Elaborado por: Karla Moyón, autora del presente proyecto

Tabla Detalle_pedido

Almacena el detalle del pedido, es decir, en una compra pueden solicitar uno o más productos, por esta razón, esta tabla se encuentra relacionada con Pedido.

Tabla 19*Tabla Detalle Pedido*

Campo	PK	FK	Tipo	Descripción
id_detalle			Integer	Número de identificación del detalle del pedido.
num_pedido			Integer	Número de pedido.
nombre_prod			Varchar(100)	Nombre del producto.
Cantidad			Integer	Cantidad de productos que desea el consumidor.
valor_unitario			Double	Valor unitario del producto
sub_total			Double	Subtotal del valor del producto
subtotal_impuestos			Double	Subtotal de impuestos del producto
valor_unitario			Double	Valor unitario del producto

Elaborado por: Karla Moyón, autora del presente proyecto

Tabla Tipo_comprobante

Almacena la información del tipo de comprobante que se va a emitir. (Ver Tabla 6 – Tipo de comprobante)

Tabla 20*Tabla Tipo de comprobante*

Campo	PK	FK	Tipo	Descripción
cod_comprobante			Numeric	Código de comprobante.
nombre_comprobante			Varchar(50)	Nombre comprobante.

Elaborado por: Karla Moyón, autora del presente proyecto

Tabla Documento

Almacena la información del documento que va a ser emitido electrónicamente. Se encuentra relacionado con las tablas: Contribuyente, Pedido y Comprobante, debido a que un contribuyente va a emitir un documento, de acuerdo al tipo de comprobante (factura, nota de

débito, etc.). En este proyecto se va a realizar facturas electrónicas, por este motivo el documento va a tener una relación uno a uno con pedido.

Tabla 21

Tabla Documento

Campo	PK	FK	Tipo	Descripción
num_comprobante			Numeric(9)	Número de comprobante.
RUC			Varchar(15)	RUC del Contribuyente.
num_pedido			Integer	Número del pedido.
cod_comprobante			Numeric	Código de comprobante.
tipo_ambiente			Integer	Tipo de ambiente de emisión de la factura. (Ver tabla 7).
clave_acceso			Numeric(49)	Clave de acceso para la emisión de la factura.
Serie			Numeric(6)	Serie de la factura. (unión de número de establecimiento y punto de emisión).
digito_verificador			Numeric	Dígito verificador para la emisión de la factura.
tipo_emision			Numeric	Tipo de emisión: por defecto es igual a 1, que corresponde a emisión normal.
fecha_emision			Date	Fecha de emisión de la factura electrónica.
Documento			File	Documento en formato XML.

Elaborado por: Karla Moyón, autora del presente proyecto

Tabla valores

Almacena la información de los valores que va a contener el documento que se va a emitir.

Tabla 22

Tabla Valores

Campo	PK	FK	Tipo	Descripción
id_valor			Numeric	Código de comprobante.
num_comprobante			Varchar(50)	Nombre comprobante.
Valor			Double	Valor
Nombre			Varchar(100)	Nombre del valor (subtotal, valor IVA, total descuento, etc.)

3.2.5. Diagrama de clases

El propósito de este diagrama es mostrar las clases necesarias para poder construir una aplicación web de facturación electrónica. Las clases a utilizar son:

- **Clase Login**

Su función será realizar la autenticación de los usuarios que ingresen al sistema.

- **Clase Woocommerce**

Se encargará de establecer la conexión con WooCommerce y el sistema KV – Facturación.

- **Clase Sincronización**

Es quien se encargará de realizar la sincronización de la tienda virtual para obtener los pedidos y mostrar mediante la interfaz de “Mis facturas”.

- **Actualización_pedido**

Esta clase se encargará de actualizar en la base de datos del presente proyecto el estado del pedido quien tenga un estado diferente a completado.

- **Documento XML:**

Dentro de la clase, es donde se va a formar la estructura XML según las especificaciones del SRI, de acuerdo al pedido que el usuario desee facturar.

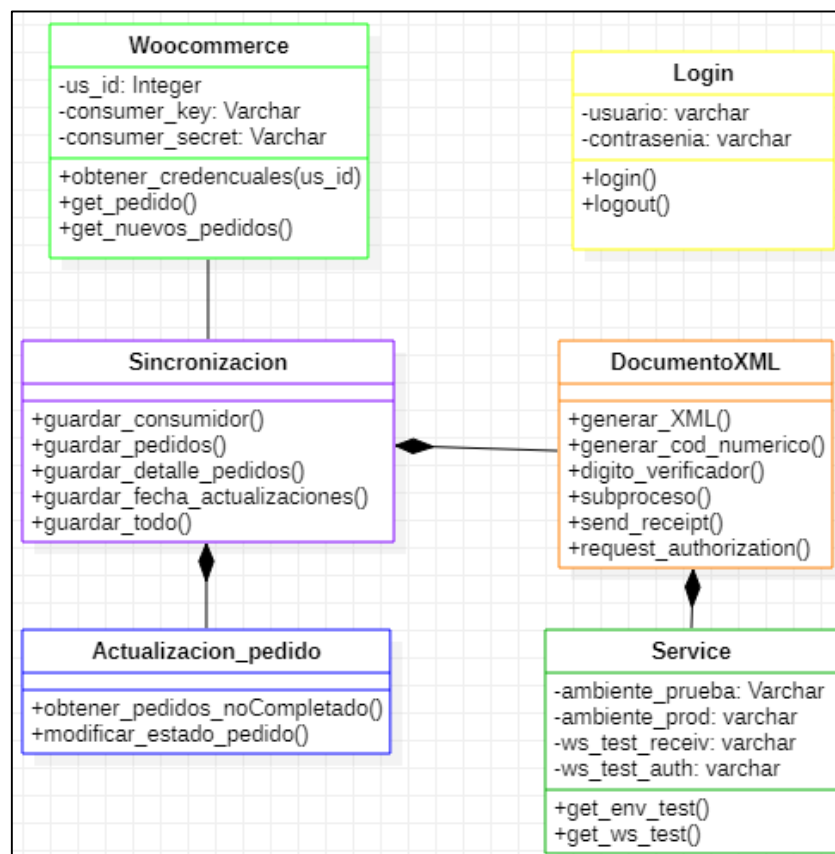
- **Service:**

Clase donde se va a establecer una conexión en el ambiente de pruebas con el SRI para la validación y autorización del XML enviado con los datos del pedido seleccionado.

A continuación, se presenta el diagrama de clases, que será utilizada como base para la construcción del sistema.

Figura 28

Diagrama de clases



Elaborado por: Karla Moyón, autora del presente proyecto

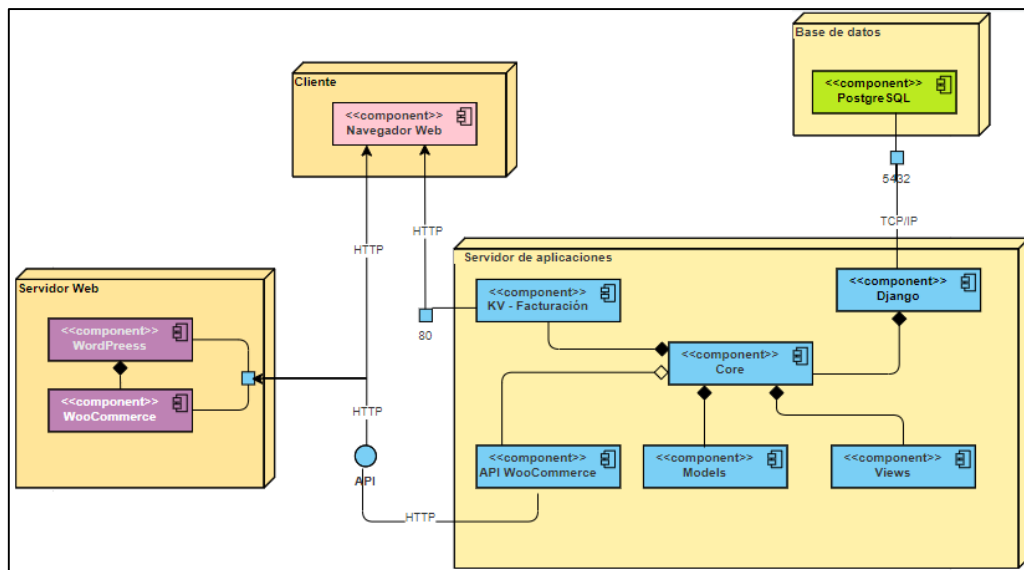
3.2.6. Diagrama de despliegue

El diagrama de despliegue permite visualizar y comprender de mejor manera los componentes más relevantes que se utilizaron dentro de la construcción de KV – Facturación.

A continuación, se muestra el diseño del diagrama de despliegue.

Figura 29

Diagrama de despliegue



Elaborado por: Karla Moyón, autora del presente proyecto

En la figura 29, se puede observar que se presenta tres nodos importantes: cliente, tienda y KVFacturación.

- **Nodo cliente:** Se encarga de mostrar la aplicación de manera intuitiva y amigable para el usuario, es decir es el navegador web quien ejecuta el código HTML, JavaScript y CSS.
- **Nodo tienda:** Este nodo contiene dos subelementos más relevantes

WordPress: Sistema de gestión de contenidos que se utilizó para la creación de la tienda virtual.

WooCommerce: Plugin de WordPress que se encarga de implementar una tienda virtual totalmente funcional y empezar a vender en línea.

- **Nodo Facturación:** Dentro de este nodo se describen tres componentes relevantes para el desarrollo del sistema

Django: Framework de desarrollo

Python: Lenguaje de programación

Facturación: Aplicación que contiene los componentes más importantes de KV – Facturación, como los modelos que son base principal para la creación de la base de datos, a la vez contiene un archivo Core.py, el cual contiene clases que se encarga de la

gestión de pedidos. Finalmente, el archivo `sri.py`, donde existen clases que se encargan de la conexión del sistema con el Web Service del SRI, quien se encarga de la validación de la factura electrónica.

3.3.Construcción

En este subcapítulo se mostrará el desarrollo para la construcción de la aplicación web de facturación electrónica para tiendas virtuales, basado en la metodología Scrum. Se describirá el código más relevante del sistema y las actividades realizadas en cada iteración para cumplir con el desarrollo del proyecto.

3.3.1. Configuraciones Principales

Antes de empezar con el desarrollo del sistema, se procede a la instalación y configuración del entorno.

Para la construcción del sistema se utilizó Python como lenguaje de programación, Django como framework, postgresSQL como gestor de base de datos y como editor de código se determinó Visual Studio Code.

Creación del proyecto en Django

Una vez instalado Django, mediante el comando:

```
$ pip install django
```

Se utilizó la herramienta de gestión de `django-admin`, el cual permite crear la carpeta donde se almacenará el proyecto principal, ficheros básicos y los scripts para su gestión. Para la creación del proyecto en django se utilizó el siguiente comando:

```
$ django-admin startproject KVFacturacion
```

A continuación, se procede a crear la aplicación de nuestro sistema utilizando:

```
$ python manage.py startapp Facturacion
```

Para iniciar el servidor web se utiliza el siguiente comando:

```
$ python manage.py runserver
```

Cabe resaltar, que al transcurso del desarrollo de KV – Facturación se almacenaron sus cambios en el repositorio de GitHub. [Ver repositorio](#)

A continuación, se describe cada iteración con sus respectivas tareas planteadas de acuerdo a la metodología seleccionada para el desarrollo del sistema.

3.3.2. Sprint 1

El primer Sprint empezó el 31 de diciembre del 2021, teniendo como tiempo de duración de una semana. El objetivo de esta iteración es crear apropiadamente los modelos en el framework de Django, de acuerdo al diseño de la base de datos. Para cumplir dicho sprint se planteó dos tareas, las cuales se describen a continuación con su respectivo detalle de la construcción para el cumplimiento de las mismas.

- Diseño de base de datos

Ver subcapítulo **Diagrama de base de datos** donde se muestra el diagrama relacional con la explicación de cada tabla y los campos de sus columnas.

- Crear modelos

Para la creación de los modelos de las bases de datos, es primordial establecer la conexión con el gestor de la base de datos. Al utilizar el framework de Django la configuración se realiza en el proyecto principal en el archivo settings.py.

En la figura 30, se presenta la configuración realizada:

Figura 30

Configuración base de datos en KVFacturacion/setting.py

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'KV-facturacion',
        'USER': 'postgres',
        'PASSWORD': '123',
        'HOST': '127.0.0.1',
        'DATABASE_PORT': '5432',
    }
}
```

Elaborado por: Karla Moyón, autora del presente proyecto

De acuerdo al diagrama de base de datos presentado se utilizó el siguiente código para la creación de modelos, es decir las tablas con sus respectivas columnas, donde se van a almacenar los datos a utilizar para la creación de la factura electrónica.

A continuación, se detalla los modelos más relevantes del sistema.

Figura 31

Clase modelo Contribuyente creada en Facturacion/models.py

```
class Contribuyente(models.Model):
    usuario = models.OneToOneField(Usuario, on_delete=models.CASCADE)
    RUC = models.CharField(primary_key=True, max_length=15)
    razon_social = models.CharField(max_length=40, verbose_name="Razón social")
    nombre_comercial = models.CharField(max_length=300, verbose_name="Nombre Comercial", blank=True)
    url_tienda = models.CharField(max_length=300, verbose_name="URL Tienda")
    consumer_key = models.CharField(max_length=100)
    consumer_secret = models.CharField(max_length=100)
    direccion = models.CharField(max_length=900, verbose_name="Dirección")
    contabilidad = models.BooleanField()
    telefono = models.CharField(max_length=10, verbose_name="Teléfono")
    firma_electronica = models.FileField()
    fecha_caducidad = models.DateField()

    def __str__(self):
        return self.razon_social
```

Elaborado por: Karla Moyón, autora del presente proyecto

Figura 32

Clase modelo Pedido y Detalle Pedido creada en Facturacion/models.py

```
class Pedido (models.Model):
    num_pedido = models.CharField(primary_key=True,max_length=50)
    id_consumidor = models.ForeignKey(Consumidor,on_delete=models.CASCADE,null=True)
    fecha_pedido = models.CharField(max_length=50)
    estado_pedido = models.CharField(max_length=20)
    envio = models.BooleanField(default=False)
    href_pedido = models.URLField()
    valor_total = models.DecimalField(max_digits=9, decimal_places=4,default=0)
    total_impuestos = models.DecimalField(max_digits=9, decimal_places=4,default=0)
    valor_transporte = models.DecimalField(max_digits=9, decimal_places=4,default=0)
    total_compra = models.DecimalField(max_digits=9, decimal_places=4,default=0)
    id_contribuyente = models.ForeignKey(Contribuyente,on_delete=models.CASCADE,null=True)
    esta_facturado= models.BooleanField(default=False)

class Detalle_pedido(models.Model):
    id_pedido = models.ForeignKey(Pedido,on_delete=models.CASCADE,null=False)
    sku = models.CharField(max_length=10)
    nombre_prod = models.CharField(max_length=100)
    cantidad = models.IntegerField()
    valor_unitario = models.DecimalField(max_digits=9,decimal_places=4, default=0)
    subtotal = models.DecimalField(max_digits=9,decimal_places=4,default=0 )
    subtotal_impuestos = models.DecimalField(max_digits=9,decimal_places=4,default=0 )
```

Elaborado por: Karla Moyón, autora del presente proyecto

Figura 33

Clase modelo Documento creada en Facturacion/models.py

```
class Documento(models.Model):
    num_comprobante = models.CharField(primary_key=True,max_length=9)
    id_pedido = models.OneToOneField(Pedido,on_delete=models.CASCADE)
    tipo_ambiente = models.CharField(max_length=2)
    tipo_emision = models.IntegerField(default=1) #Emisión normal
    id_usuario = models.ForeignKey(Contribuyente,on_delete=models.CASCADE, null=True)
    clave_acceso = models.CharField(max_length=49)
    tipo_comprobante = models.ForeignKey(Tipo_comprobante,on_delete=models.CASCADE, null=True)
    serie = models.CharField(max_length=6) #establecimiento + punto emision
    fecha_emision = models.DateTimeField(auto_now_add=True)
    documento = models.FileField()
```

Elaborado por: Karla Moyón, autora del presente proyecto

El código fuente, para ver todos los modelos creados, se puede encontrar en el siguiente enlace:

[Facturacion/models.py](#)

Para crear los modelos en PostgreSQL se utilizó el siguiente comando: `$ Python manage.py makemigrations` con el fin de transformar el modelo en sentencia SQL y a su vez crear una carpeta con las migraciones basadas de los cambios realizados. Por otro lado, para aplicar los cambios de las migraciones en el gestor de base de datos se debe utilizar `$ Python manage.py migrate`.

Al finalizar el sprint se confirma que el objetivo ha sido cumplido.

3.3.3. Sprint 2

El segundo Sprint comenzó el 07 de enero del 2022, de igual manera con la duración de una semana, teniendo como objetivo guardar los pedidos extraídos de WooCommerce a la base de datos de KV – Facturación.

Para cumplir con dicho objetivo tenemos que agregar un súper usuario, es decir quien tiene el control sobre todo el sistema.

El proyecto está configurado para que el súper usuario, al que llamaremos administrador, tenga acceso y los permisos necesarios para agregar, editar o eliminar: consumidores, contribuyentes, establecimientos, pedidos, punto de emisión, tipo de comprobantes y usuarios.

A través del siguiente comando se puede agregar un súper usuario:

```
$ python manage.py createsuperuser
```

Cabe aclarar que, para el desarrollo del sistema, hemos considerado que la autenticación para acceder a la administración sea mediante correo electrónico, por tal motivo se realiza las siguientes configuraciones:

Figura 34

Clase usuario dentro del fichero Facturacion/models.py

```
class Usuario(AbstractUser):
    email = models.EmailField(unique=True)
    USERNAME_FIELD = 'email'
    REQUIRED_FIELDS = ['username']
    def nombre(self):
        return self.last_name + " " + self.first_name
```

Elaborado por: Karla Moyón, autora del presente proyecto

- Conexión con WooCommerce

Dentro de la clase de Woocommerce se crearon las siguientes funciones

Figura 35

Clase Woocommerce dentro del fichero Facturacion/Core.py

```
class Woocommerce:
    def __init__(self, us_id):
        self.us_id = us_id
        cred1, cred2, url_t = self.obtener_credenciales(self.us_id)
        self.wcapi = API(
            url=url_t,
            consumer_key = cred1,
            consumer_secret = cred2,
            wp_api=True,
            version="wc/v3"
        )

    def obtener_credenciales(self, us_id):
        for datos in Contribuyente.objects.filter(usuario_id= us_id):
            cred1=datos.consumer_key
            cred2=datos.consumer_secret
            url = datos.url_tienda
        return cred1, cred2, url

    def get_pedido(self, id_pedido):
        url = 'orders/' + str(id_pedido)
        print(url)
        pedido = self.wcapi.get(url).json()
```

```

return pedido

def get_nuevos_pedidos(self):
    cant_pedidos=100
    var = Actualizaciones.objects.order_by('ultima_fecha_pedido').last() #Obtengo la última fecha de actualización

    if (var == None):
        ultima_fecha = "2022-01-01T00:00:00"
    else:
        ultima_fecha= var.ultima_fecha_pedido
    url='orders/?per_page={} &after={} '.format(cant_pedidos,ultima_fecha)
    print(url)
    pedido = self.wcapi.get(url).json()
    return pedido,cant_pedidos

```

Elaborado por: Karla Moyón, autora del presente proyecto

A continuación, se describe las funciones utilizadas en la clase Woocommerce.

✓ **obtener credenciales**

En esta función el sistema realiza una consulta a la base de datos de la información almacenada en la tabla contribuyente para obtener las credenciales para acceder a la API de WooCommerce, consumer key y el consumer secret, las cuales son proporcionadas por la tienda virtual creada por el usuario y el enlace de la tienda. Cabe mencionar que la búsqueda se realiza dependiendo del usuario quien inició sesión.

✓ **get_pedido**

La presente función realiza una consulta a la API de WooCommerce para consultar el pedido, de acuerdo a la identificación del mismo, llamado (id_pedido)

Al ejecutar esta función se genera la url: orders/id_pedido, donde como resultado toda la información del pedido a través de un json.

✓ **get_nuevos_pedidos**

De igual manera esta función realiza una consulta a la API de WooCommerce para consultar varios pedidos.

Al ejecutar la función da como resultado la siguiente url: `orders/?per_page=100&after=2022-01-29T20:55:01`, donde podemos observar que el parámetro `per_page` es igual a `cant_pedidos(100)`. Cabe mencionar que, si no se indica este parámetro, por defecto muestra 10 pedidos, siendo 100 lo máximo que permite. Por otro lado, se tiene `after` lo cual, en la función realiza una consulta a la base de datos para conocer la última fecha registrada de actualización, en el caso de no existir, la fecha será tomada en cuenta desde 2022-01-01T00:00:00. Finalmente, la url de ejemplo nos dará como resultado 100 pedidos, considerando los pedidos realizados después del 29 de enero del 2022 a las 20 horas 55 minutos, 01 segundos. El resultado será en formato json.

El código fuente de la clase Woocommerce, se puede obtener en el siguiente enlace:

[Facturacion/Core.py](#)

- Guardar pedidos extraídos de WooCommerce

Para guardar los pedidos extraídos de WooCommerce se procede a crear una nueva clase llamada sincronización, en la cual se tiene funciones, tales como: guardar consumidor, guardar pedidos, guardar detalle de los pedidos, guardar fecha actualizaciones los cuales serán ejecutados en el método principal de la clase llamado guardar todo.

Cabe mencionar que en esta nueva clase creada se tiene que usar la clase de Woocommerce para establecer una conexión con la API y obtener los pedidos.

A continuación, se presenta el código de la función guardar todo.

Figura 36

Método guardar todo de la clase Sincronización dentro del fichero Facturacion/Core.py

```

def guardar_todo(self,us_id):
    datos_json = Woocommerce(us_id)
    pedidos, cant = datos_json.get_nuevos_pedidos()
    for pedido in pedidos:
        cedula_consumidor = ""

        if (pedido['billing']['address_2']==""):
            #cedula_consumidor = ""
            meta_data = pedido['meta_data']
            for data in meta_data:
                if (data['key']=='cedula'):
                    cedula_consumidor = data['value']
            else:
                cedula_consumidor = pedido['billing']['address_2']
        if(pedido['billing']['first_name'] == ""):
            cedula_consumidor = '99999999999999'

        if (Consumidor.objects.filter(identificacion = cedula_consumidor).exists()):
            self.guardar_pedidos(pedido['id'],cedula_consumidor,pedido['date_created'],pedido['status'],
            pedido['_links']['self'][0]['href'],pedido['total'],pedido['total_tax'],us_id)
        else:
            if (pedido['billing']['first_name']==""): ## Cuando es consumidor final
                self.guardar_consumidor(cedula_consumidor,'CONSUMIDOR FINAL', " , ",",")
            else:
                self.guardar_consumidor(cedula_consumidor,pedido['billing']['first_name']+' '
                +pedido['billing']['last_name'], pedido['billing']['email'] ,pedido['billing']['address_1'],
                pedido['billing']['phone'])

            self.guardar_pedidos(pedido['id'],cedula_consumidor,pedido['date_created'],pedido['status'],
            pedido['_links']['self'][0]['href'],pedido['total'],pedido['total_tax'],us_id)

    detalle = pedido['line_items']
    for det in detalle:
        nombre_prod = det['name']
        cantidad = det['quantity']
        valor_unitario = det['price']
        subtotal = det['total']
        sku = det['sku']
        if (det['total_tax']=="0.00"):
            subtotal_impuestos = 0
        else:
            subtotal_impuestos = det['taxes'][0]['total']
        self.guardar_detalle_pedidos(pedido['id'],sku,nombre_prod,cantidad,valor_unitario,subtotal,
        subtotal_impuestos)

    # Tabla actualizaciones
    self.guardar_fecha_actualizaciones()

```

Elaborado por: Karla Moyón, autora del presente proyecto

En el método guardar todo, representado en la Figura 36, se muestra el siguiente análisis realizado al json recibido con la información de los pedidos.

Análisis de datos

1. Cada pedido es un objeto, es los cuales puede contener objetos y arreglos con información.
2. Al recorrer cada pedido se considera el elemento “address_2” del objeto “billing”, en caso de estar vacío la cedula del consumidor será extraída del arreglo “meta data” quien tenga como llave “cédula”. Caso contrario cédula consumidor será igual al valor que se encuentre en el elemento “address_2” del objeto “billing”.
3. Si el objeto billing está vacío, quiere decir que el cliente no ingresó sus datos para la factura, lo que se considerará como consumidor final con cédula de identidad igual a “99999999999999”.
4. Se comprueba la existencia del consumidor en la base de datos, en caso de que el número de cédula ya esté registrada. Caso contrario, se realiza la validación mencionados 2 y 3, y se procede a guardar los pedidos de acuerdo a las columnas definidas en el modelo con su respectivo número de cedula del consumidor.
5. Se recorre el objeto de “line_items” donde se encuentra el detalle del pedido como, por ejemplo, el nombre del producto, cantidad, valor unitario, etc. Con el fin de guardar en la tabla detalle pedido, mediante el método guardar detalle del pedido.
6. Finalmente, el sistema guarda la fecha de la actualización, última fecha pedido y último número del pedido en la tabla actualizaciones.

El código fuente de la clase Sincronización, se puede obtener en el siguiente enlace:

[Facturacion/Core.py](#)

Al finalizar el sprint se cumple con el objetivo planteado con sus respectivas validaciones.

3.3.4. Sprint 3

El tercer Sprint comenzó el 14 de enero del 2022, de igual manera con la duración de una semana, teniendo como objetivo visualizar y filtrar los datos generales de los pedidos, de acuerdo con el usuario que inició sesión. Las tareas planteadas en el presente Sprint son:

- Autenticación de usuarios

Para la autenticación de usuarios se siguieron los siguientes pasos:

1. Crear interfaz

En la creación de la interfaz de inicio de sesión del usuario se utilizó como plantilla SB Admin 2, el cual está basado en Bootstrap 4, de código abierto, ideal para la creación de paneles y aplicaciones web de forma rápida y sencilla. (startbootstrap, 2021) [Ver plantilla](#)

Dentro del archivo Facturacion/templates/facturacion/login.html se creó una alerta en caso de que las credenciales utilizadas sean incorrectas, utilizando el siguiente código.

Figura 37

Alerta en template Facturacion/login.html

```
{% if form.errors %}
{% for field in form %} {% if field.errors %} {% for error in field.errors %}
  <div class="alert alert-success">
    <a href="{%url 'login' %}" class="close" data-dismiss="alert" aria-label="close">&times;</a>{{ error|escape }}
  </div>

{% endfor %} {% endif %} {% endfor %}
{% if form.non_field_errors %} {% for error in form.non_field_errors %}
  <div class="alert alert-danger">
    <a href="{%url 'login' %}" class="close" data-dismiss="alert" aria-label="close">&times;</a> {{ error|escape }}
  </div>
{% endfor %} {% endif %} {% endif %}
```

Elaborado por: Karla Moyón, autora del presente proyecto

En el siguiente enlace se encuentra el código fuente completo del HTML de inicio de sesión

[Facturacion/templates/facturacion/login.html](#)

2. Generar URL

El framework de Django ya tiene vistas predeterminadas, para la autenticación de usuarios. En el archivo `KVFacturacion/urls.py` utilizaremos la vista: `from django.contrib.auth.views import LoginView, LogoutView`. A continuación, se procede a generar los path dentro del arreglo `urlpatterns`, como se muestra en el siguiente código:

Figura 38

Generar URL en el archivo KVFacturacion/urls.py

```
urlpatterns = [
    path('accounts/login/', LoginView.as_view(template_name='Facturacion/login.html', redirect_authenticated_user=True), name='login'),
    path('logout/', LogoutView.as_view(template_name='Facturacion/login.html'), name='logout'),
]
```

Elaborado por: Karla Moyón, autora del presente proyecto

En la figura anterior, se puede observar que se definen los templates correspondientes al iniciar sesión (`LoginView`) y cerrar sesión (`LogoutView`). Un elemento importante a mencionar es `redirect_authenticated_user = True`, el cual, al momento de la autenticación, guarda el id del usuario.

En el siguiente enlace se encuentra el código fuente completo de [KVFacturacion/urls.py](#)

3. Crear vista

Dentro de la aplicación facturación en el archivo `views.py` se define la vista como se muestra en el siguiente código.

Figura 39

Generar método para renderizar la interfaz de inicio de sesión.

```
def login(request):  
    return render(request,'Facturacion/login.html')
```

Elaborado por: Karla Moyón, autora del presente proyecto

Un punto a resaltar es que al momento que se realiza la autenticación, la idea es que el usuario no pueda acceder a otras vistas por medio de la url, sin haber iniciado sesión, por este motivo se debe importar la librería:

```
from django.contrib.auth.decorators import login_required
```

La cual se encarga de comprobar la autenticación del usuario. Además, se debe agregar `@login_required` antes de cada método que renderice una vista html, como se indica en el siguiente código para que únicamente puedan acceder a las vistas una vez que el usuario haya iniciado sesión.

Figura 40

Archivo Facturacion/view.py.

```
@login_required  
def Mis_facturas(request):  
  
@login_required  
def vista_prev_fact(request,num_pedido):  
  
@login_required  
def reporte_general(request):  
  
@login_required  
def mis_datos(request):  
  
@login_required  
def editar_consumidor(request, identificacion,num_pedido):
```

Elaborado por: Karla Moyón, autora del presente proyecto

En el siguiente enlace se encuentra el código fuente completo del archivo [Facturacion/view.py](#).

- Visualizar datos generales para la facturación

Para poder cumplir con esta tarea se requiere generar un template, el cual muestre los datos que han sido almacenados anteriormente en la base de datos de los pedidos extraídos de WooCommerce.

En el siguiente enlace se encuentra el código fuente completo del archivo templates/Facturacion/mis_facturas.html

En la función “Mis_facturas” del archivo facturacion/views.py se tiene que realizar la consulta a la base de datos de la información almacenada de los pedidos y enviar los datos a la interfaz. A continuación, se muestra el código utilizado.

Figura 41

Método Mis facturas en el archivo Facturacion/views.py

```
@login_required
def Mis_facturas(request):
    us= request.user.id #obtengo el id de quien inició sesión
    a = Contribuyente.objects.get(usuario_id = us)
```

Elaborado por: Karla Moyón, autora del presente proyecto

Como se indicó en los prototipos de interfaces, “Mis facturas” tiene que tener un botón, en el cual al ser la primera vez que ingrese el usuario tiene que actualizar para poder establecer la conexión con WooCommerce y obtener los pedidos. En caso de que la tienda virtual no tenga pedidos, indicará un mensaje de “No hay pedidos por agregar”, caso contrario, se mostrará un mensaje indicando que los nuevos pedidos han sido agregados. El código de implementación sería como se muestra en la siguiente figura.

Figura 42

Método Mis facturas al agregar nuevos pedidos

```
if request.method == 'POST':
    fun = Actualizacion_pedido()
    wc = Woocommerce(us)
    sincro = Sincronizacion()
    nuevos_pedidos,cant_pedidos = wc.get_nuevos_pedidos()

    if (len(nuevos_pedidos) == 0):
        messages.info(request,"No hay pedidor por agregar")
    elif (len(nuevos_pedidos) !=0):
        sincro.guardar_todo(us)
        messages.success(request,"Nuevos Pedidos Agregados")
```

Elaborado por: Karla Moyón, autora del presente proyecto

En el siguiente enlace se encuentra el código fuente completo del archivo Facturacion/views.py método [Mis Facturas](#).

- Búsqueda para filtrar pedidos.

Para cumplir con la tarea de filtrar pedidos de acuerdo a los campos ingresados por el usuario, se usó una plantilla SB Admin ([Ver plantilla](#)), donde se utiliza DataTables, es decir un plug-in que pertenece a la librería de jQuery Javascript y se encarga de realizar el filtrado de manera asíncrona de acuerdo con el campo ingresado por el usuario y los datos disponibles de la tabla.

Al finalizar el sprint se cumple con el objetivo planteado con sus respectivas validaciones.

3.3.5. Sprint 4

El cuarto Sprint comenzó el 21 de enero del 2022, de igual manera con la duración de una semana, teniendo como objetivo visualizar y editar los datos del pedido como vista previa de una factura electrónica, además de actualizar los pedidos de acuerdo a la tienda virtual. Las tareas planteadas en el presente Sprint son:

- Actualización de pedidos

Para dicha tarea se procede a crear una clase llamada “Actualizacion_pedido. En el siguiente enlace se encuentra el código fuente completo de la clase [Actualización pedido](#)

A continuación, se describe el método “obtener pedidos no completado”, en el cual realiza una consulta a la base de datos de los pedidos que tengan un estado diferente a “completed” y “cancelled”, completado y cancelado, los cuales serán almacenados en un arreglo.

Figura 43

Método obtener pedidos no completados de la clase Actualización pedido

```
def obtener_pedidos_noCompletado(self):
    pedidos_noC=[]
    for pedido in Pedido.objects.exclude(estado_pedido = 'completed').exclude(estado_pedido = 'cancelled'):
        pedidos_noC.append(pedido.num_pedido)
    return pedidos_noC
```

Elaborado por: Karla Moyón, autora del presente proyecto

En la figura 44, se presenta el siguiente método de la clase, llamado “modificar estado pedido”, en el cual recibe como entrada el arreglo con los números de los pedidos, mencionados anteriormente.

Figura 44

Método modificar estado pedido de la clase Actualización pedido

```
def modificar_estado_pedido(self,us_id,pedidos_noC):
    wc= Woocommerce(us_id)
    sincro = Sincronizacion()
    for pedido in pedidos_noC:
        datos_pedidos = wc.get_pedido(pedido)
        consulta=Pedido.objects.get(num_pedido = pedido)
        if (consulta.estado_pedido != datos_pedidos['status']):
            consulta.estado_pedido =datos_pedidos['status']
            consulta.save()
    sincro.guardar_fecha_actualizaciones()
    return
```

Elaborado por: Karla Moyón, autora del presente proyecto

Como se puede observar en el método “modificar estado pedido”, se conecta a la API de WooCommerce, con el fin de comparar cada número del pedido que se encuentra en el vector

de entrada con el estado del pedido actual que consta en la tienda virtual, mediante el método `get pedido` de la clase `Woocommerce` y se realiza una comparación del estado del pedido, en caso que este haya cambiado se procede a guardar en la base de datos este cambio.

- Visualizar los datos de la factura electrónica

En la interfaz “Mis facturas” se debe seleccionar la opción de “Ver” del pedido que se desea facturar, el cual se encuentra en la columna de “acciones”, el sistema automáticamente muestra una nueva ventana de la vista preliminar de los datos de la factura electrónica, es decir, información de la factura, emisor, pedido, consumidor y detalle pedido.

Figura 45

Método vista previa factura en el fichero `Facturación/views.py`

```
@login_required
def vista_prev_fact(request,num_pedido):
    sum=0
    fecha_emision = datetime.today().strftime('%d/%m/%Y')
    us= request.user.id
    emisor = Contribuyente.objects.get(usuario_id = us)
    pedido = Pedido.objects.get(num_pedido = num_pedido)
    consumidor = Consumidor.objects.get(identificacion = pedido.id_consumidor_id)
    detalle_pedido = Detalle_pedido.objects.filter(id_pedido_id = num_pedido)
```

Elaborado por: Karla Moyón, autora del presente proyecto

En la figura 45, se muestra que se realiza una consulta a la base de datos para consultar los datos del emisor, pedido seleccionado por el usuario, consumidor y finalmente del detalle del pedido.

Un aspecto importante a detallar en esta tarea es conocer cómo se obtiene los datos de la factura, es decir, RUC, número de la factura, ambiente y clave de acceso.

RUC: por medio de la consulta a la base de datos de la información del emisor, también llamado contribuyente, se obtiene este campo.

Número de la factura: se compone por los 3 dígitos del número de establecimiento del contribuyente, añadiendo los 3 dígitos del punto de emisión y finalmente por los 9 dígitos del número de comprobante.

Ambiente: De acuerdo a la ficha técnica de comprobantes electrónicos esquema offline el SRI tiene dos tipos de ambiente: pruebas y desarrollo. En el presente proyecto se va a utilizar el ambiente de pruebas puesto que el contribuyente podrá realizar todas las acciones en desarrollo, ejecutando y verificando que los comprobantes electrónicos cumplan con los esquemas planteados. (SRI, Ficha técnica comprobantes electrónicos esquema offline, 2020)

Clave de acceso: Como se había mencionada en el marco teórico del presente proyecto, la clave de acceso tiene que estar formada por 49 dígitos, los cuales están conformados por:

Tabla 23

Tabla campos para formar la clave de acceso

Campo	Formato	Longitud
Fecha de emisión	ddmmaaaa	8
Tipo de comprobante	Tabla 6	2
Número de RUC	17249979500001	13
Tipo de Ambiente	Tabla 7	1
Serie	001002	6
Número del Comprobante (secuencial)	000000001	9
Código numérico	Numérico	8
Tipo de emisión	Tabla 8	1
Dígito verificador (módulo 11)	Numérico	1
Total		49

Elaborado por: Karla Moyón, autora del presente proyecto

El código numérico es un mecanismo el cual brinda seguridad al contribuyente en cada emisión de un comprobante electrónico, el algoritmo para conformar este código es decisión absoluta

del emisor (SRI, Ficha técnica comprobantes electrónicos esquema offline, 2020). En el presente proyecto, para generar dicho código se utiliza una función llamada “generar_cod_numerico” que genera 8 dígitos aleatorios. En la figura 46 se indica cómo se desarrolló.

Figura 46

Método generar_cod_numerico de la clase DocumentoXML

```
def generar_cod_numerico(self):  
    num = 8  
    cod=""  
    for e in range(num):  
        cod+=str(random.randint(1, 9))  
    return cod
```

Elaborado por: Karla Moyón, autora del presente proyecto

El dígito verificador será aplicado sobre toda la clave de acceso, es decir los 48 dígitos y deberá ser implementado por el emisor a través del método denominado “módulo 11”, considerando como factor de chequeo ponderado al número dos. En el caso que el resultado del dígito verificador sea igual a once, el resultado será el cero y cuando el resultado obtenido sea diez, el dígito verificador será el uno.

Figura 47

Método digito_verificador de la clase DocumentoXML

```
def digito_verificador(self,codigo):  
    ban=2  
    sum= 0  
    codigo = codigo[len(codigo):-1] # invierto  
    for cod in codigo:  
        cod= int(cod)  
        if (ban == 8):  
            ban=2  
        sum += cod*ban  
        ban += 1
```

```

    digito = 11 - (sum%11)
    if digito == 11:
        digito = 0
    elif digito == 10:
        digito = 1
    return str(digito)

```

Elaborado por: Karla Moyón, autora del presente proyecto

En el siguiente enlace se encuentra el código fuente completo de la clase [DocumentoXML](#)

Finalmente, en la figura 48, se muestra el resultado de cada campo para completar la información de la factura.

Figura 48

Método vista_prev_fact del fichero Facturacion/views.py

```

def vista_prev_fact(request,num_pedido):
    ....
    ....
    establecimiento = Establecimiento.objects.get(id_usuario_id = emisor.RUC)
    num_emision = Punto_emision.objects.get(establecimiento_id = establecimiento.num_establecimiento)
    num_comprobante = Documento.objects.order_by('num_comprobante').last()

    if (num_comprobante == None):
        num_comprobante = "1".zfill(9)
    else:
        num_comprobante = str(int(num_comprobante.num_comprobante)+1).zfill(9)

    num_factura = str(establecimiento.num_establecimiento)+"-"+str(num_emision.num_punto_emision)
    + "-" + num_comprobante

    ambiente = 1
    serie = str(establecimiento.num_establecimiento)+str(num_emision.num_punto_emision)
    tipo_comprobante = '01'
    cod_numerico = doc.generar_cod_numerico()
    tipo_emision = "1"
    clave_acceso = str(fecha2)+ tipo_comprobante+emisor.RUC + str(ambiente)+serie+
    num_comprobante+cod_numerico+tipo_emision

    dig_verificador = doc.digito_verificador(clave_acceso)
    clave_acceso = clave_acceso+dig_verificador

```

Elaborado por: Karla Moyón, autora del presente proyecto

En el siguiente enlace se encuentra el código fuente completo de la función para generar la vista previa de la factura electrónica [Facturacion/views.py](#)

- Editar los datos de la factura electrónica

En la sección de datos del consumidor de la factura electrónica, hay una opción para editar estos campos. Al seleccionar este botón, el sistema abre una nueva ventana donde se indica los datos del consumidor, los cuales pueden ser editados y guardados. Por otro lado, existe la opción del regresar que nos indica nuevamente la vista preliminar del pedido.

En primer lugar, se crea un nuevo fichero llamado “forms.py”, el cual utiliza una librería de formularios propia de Django y además se importa el modelo de consumidor, para obtener todos los datos y poder acceder a ellos. A continuación, se muestra el código del nuevo fichero.

Figura 49

Fichero Facturacion/forms.py

```
from django import forms
from Facturacion.models import Consumidor

class FormularioConsumidor(forms.ModelForm):
    class Meta:
        model = Consumidor
        fields = '__all__'
```

Elaborado por: Karla Moyón, autora del presente proyecto

Se procede a crear el HTML, donde se llama al formulario creado con la librería de Django ([Ver código](#)).

Finalmente, se crea un nuevo método en el fichero Facturación/views.py, el cual permita editar los datos del consumidor a través del HTML templates/Facturacion/editar_consumidor.html.

Al momento de guardar los cambios el sistema valida el formulario y a su vez guarda los datos modificados. A continuación, se muestra el código mencionado anteriormente.

Figura 50

Método editar_consumidor del fichero Facturacion/views.py

```
@login_required
def editar_consumidor(request, identificacion,num_pedido):
    consumidor = Consumidor.objects.filter(identificacion= identificacion).first()
    form = FormularioConsumidor(instance=consumidor)
    if request.method == 'POST':
        consumidor = Consumidor.objects.get(identificacion= identificacion)
        form = FormularioConsumidor(request.POST, instance=consumidor)
        if form.is_valid():
            form.save()
            messages.success(request,"Cambios guardados exitosamente")
    return render(request, 'Facturacion/editar_consumidor.html',{'form': form, "consumidor":consumidor,
        "num":num_pedido})
```

Elaborado por: Karla Moyón, autora del presente proyecto

Al finalizar el sprint se cumple con el objetivo planteado con sus respectivas validaciones.

3.3.6. Sprint 5

El quinto y último Sprint comenzó el 28 de enero del 2022, fue el único sprint con la duración de dos semanas, teniendo como objetivo formar el XML de la factura y enviar al SRI para su validación. Las tareas planteadas en el presente Sprint son:

- Consultar datos usuario

En la presente tarea se creó un método, en el cual se realiza una consulta a la base de datos de la información del usuario que inició sesión en KV – Facturación.

En la siguiente figura se muestra el código utilizado conforme a lo mencionado anteriormente.

Figura 51

Método mis_datos del fichero Facturacion/views.py

```
@login_required
def mis_datos(request):
```

```

us= request.user.id #obtengo el id de quien inició sesión
a = Contribuyente.objects.get(usuario_id = us)
datos2 = Usuario.objects.get(id= us)
return render(request,'Facturacion/mis_datos.html',{'datos': a, 'datos2': datos2})

```

Elaborado por: Karla Moyón, autora del presente proyecto

- Facturar pedido

Esta tarea es fundamental para la construcción del sistema ya que se considera el núcleo principal de todo el proyecto.

En la parte inferior de la interfaz “VistaPrevia” de la factura electrónica tiene la opción de facturar, en la cual se desprende una nueva ventana denominada modal para que el usuario ingrese la contraseña de la firma electrónica. En caso de que la contraseña sea correcta el sistema procede a generar el XML según el Anexo 1 de la ficha técnica de comprobantes electrónicos esquema offline. Caso contrario indicará un mensaje indicando que la contraseña es incorrecta.

Figura 52

Fragmento de código de la función vista_prev_fact (método POST)

```

if request.method == 'POST':

    doc.generar_XML(str(ambiente),tipo_emision, clave_acceso,tipo_comprobante,
    str(establecimiento.num_establecimiento),
    str(num_emision.num_punto_emision), num_comprobante, fecha2, emisor, consumidor,
    pedido,detalle_pedido)

    passfirma=request.POST['firma']
    respuesta = doc.subproceso(passfirma,num_pedido)
    if int(respuesta) == 0:
        messages.error(request,"Contraseña de la firma incorrecta. Vuelve a intentarlo")
    elif int(respuesta) == 1:
        messages.success(request,"Documento Generado")

```

Elaborado por: Karla Moyón, autora del presente proyecto

En el siguiente enlace se encuentra el código fuente completo del método [vista_prev_fact](#)

A continuación, se presenta un fragmento de código donde se genera el XML con los datos obtenidos de la vista previa de la factura (ambiente, tipo de emisión, clave de acceso, código del documento, establecimiento, punto de emisión, secuencial, fecha de emisión, emisor, consumidor, pedido, detalle del pedido).

Figura 53

Método generar_XML de la clase DocumentoXML

```
def generar_XML(self, ambiente, emision, clave, codDoc, estab, ptoEmi, secuencial,
    fecha, emisor, consumidor, pedido, detalle_pedido):
    factura = ET.Element("factura", id="comprobante", version="1.0.0")
    ## Información tributaria
    infoTributaria = ET.SubElement(factura, "infoTributaria")
    ET.SubElement(infoTributaria, "ambiente").text = ambiente
    ET.SubElement(infoTributaria, "tipoEmision").text = emision
    ET.SubElement(infoTributaria, "razonSocial").text = emisor.razon_social
    if emisor.nombre_comercial != "":
        ET.SubElement(infoTributaria, "nombreComercial").text = emisor.nombre_comercial
    .....
    .....
    arbol = ET.ElementTree(factura)
    arbol.write("Facturacion/XMLs/" + pedido.num_pedido + ".xml", encoding="UTF-8", xml_declaration=True)
```

Elaborado por: Karla Moyón, autora del presente proyecto

En el siguiente enlace se encuentra el código fuente completo del método [generar_XML](#)

Una vez generado el XML, como siguiente paso se agrega la firma electrónica, la cual se basa en las especificaciones definidas por el estándar XADES_BES. Para agregar la firma electrónica se necesita compilar el código presentado en la figura 54, obteniendo como resultado un archivo ejecutable con extensión .jar.

Figura 54

Clase principal del firmador con estándar XADES_BES

```

public class DevelopedSignature {
    //private static Object XAdESBESSignature;

    public static void main(String[] args) throws Exception {
        String pathSignature = args[0];
        String passSignature = args[1];
        String xmlPath = args[2];
        String pathOut = args[3];
        String nameFileOut = args[4];

        try{
            XAdESBESSignature.firmar(xmlPath, pathSignature, passSignature, pathOut, nameFileOut);
            System.out.println(1);
        }catch(Exception e){
            //System.out.println("Error: " + e);
            System.out.println(0);
        }
    }
}

```

Elaborado por: Karla Moyón, autora del presente proyecto

Como se puede observar en la figura anterior, el código está realizado con Java, debido a que este lenguaje de programación cuenta con todas las librerías necesarias para cumplir con el estándar necesario para la validación del XML con el SRI. En el siguiente enlace se encuentra el código fuente completo del firmador [bin/sri](#)

El archivo .jar permite firmar el documento XML antes mencionado de la siguiente forma:

```
java -jar arg1 arg2 arg3 arg4 arg5 arg6
```

Donde:

- **arg1:** Corresponde a la ubicación donde se encuentra el archivo.jar
- **arg2:** Corresponde a la ubicación donde se encuentra el archivo de la firma con extensión .p12
- **arg3:** Corresponde a la contraseña de la firma electrónica
- **arg4:** Corresponde a la ubicación dd el archivo XML sin firma
- **arg5:** Corresponde a la ubicación donde se quiere que se guarde el nuevo XML.
- **arg6:** Corresponde al nombre del XML firmado

Al obtener el archivo con extensión .jar, se agrega al sistema KV-Facturación y es ejecutado mediante un subprocesso de Python, almacenando en una variable su resultado. En caso de ser la ejecución exitosa se obtiene como resultado “1”, caso contrario “0”.

En la siguiente figura se muestra el código donde se utiliza un subprocesso para poder firmar un documento XML.

Figura 55

Método subprocesso de la clase DocumentoXML

```
def subprocesso(self,contra,num_pedido):  
    a = subprocess.check_output('java -jar ' + r'.\Facturacion\privada\firmador.jar ' +  
    r'.\Facturacion\privada\FIRMA.p12 ' +  
    contra+' ' +  
    '.\Facturacion\XMLs\' +str(num_pedido)+".xml " +  
    r'.\Facturacion\XMLs ' +  
    str(num_pedido)+'.xml',shell=True)  
  
    return a
```

Elaborado por: Karla Moyón, autora del presente proyecto

Finalmente, con el archivo XML firmado se procede a enviar a la validación y autorización del SRI, donde el contribuyente certificará que su aplicación de facturación funcione correctamente con cada tipo de comprobante electrónico, las direcciones para la conexión con el Web Service, son los siguientes para el ambiente de pruebas: (SRI, Ficha técnica comprobantes electrónicos esquema offline, 2020)

- ✓ <https://celcer.sri.gob.ec/comprobantes-electronicos-ws/RecepcionComprobantesOffline?wsdl>
- ✓ <https://celcer.sri.gob.ec/comprobantes-electronicos-ws/AutorizacionComprobantesOffline?wsdl>

A continuación, se muestra un fragmento de código de la clase Service, donde se encarga de la conexión con el ambiente de pruebas del Web Service del SRI. Como se puede observar en la figura 56 se utiliza la librería suds. Suds es un cliente de servicio web ligero basado en SOAP para Python. (suds-community, 2022)

Figura 56

Fragmento de código de la clase Service

```
try:
    from suds.client import Client
    from suds.transport import TransportError
except ImportError:
    raise ImportError('Instalar Libreria suds')

class Service(object):

    __AMBIENTE_PRUEBA = '1'
    __AMBIENTE_PROD = '2'
    __WS_TEST_RECEIV = 'https://celcer.sri.gob.ec/comprobantes-electronicos-ws/RecepcionComprobantesOf-
fline?wsdl'
    __WS_TEST_AUTH = 'https://celcer.sri.gob.ec/comprobantes-electronicos-ws/AutorizacionComprobantesOf-
fline?wsdl'
    .....
    .....
```

Elaborado por: Karla Moyón, autora del presente proyecto

Para la validación del XML se creó un nuevo método llamado “send_receipt”, donde recibe el xml codificado en base64 y a través del Web Service es validado, obteniendo como respuesta que no existe errores, caso contrario los motivos por el cual fue Devuelto, como se indica en la figura 57.

Figura 57

Método send_receipt de la clase DocumentoXML

```
def send_receipt(self):
    name = "Facturacion/XMLs/probando.xml"
```

```

cadena = open(name, mode='rb').read()
document = parseString(cadena.strip())
xml = base64.b64encode(document.toxml('UTF-8'))
client = Client(SRIService.get_ws_test()[0])
result = client.service.validarComprobante(xml)
mensaje_error = ""

if (result[0] == 'DEVUELTA'):
    comprobante = result[1].comprobante
    mensaje_error += 'Clave de Acceso: ' + comprobante[0].claveAcceso
    mensajes = comprobante[0].mensajes
    i = 0
    mensaje_error += "\nErrores:\n"
    while i < len(mensajes):
        mensaje = mensajes[i]
        mensaje_error += 'Identificador: ' + mensaje[i].identificador + '\nMensaje: ' + mensaje[i].mensaje +
            '\nInformacion Adicional: ' + mensaje[i].informacionAdicional + '\nTipo: ' + mensaje[i].tipo + "\n"
        i += 1
    print('Error SRI', mensaje_error)

return True

```

Elaborado por: Karla Moyón, autora del presente proyecto

Una vez que el XML es validado se procede a la autorización del mismo, como se indica en el siguiente código.

Figura 58

Método send_receipt de la clase DocumentoXML

```

def request_authorization(self,access_key):
    try:
        client_auto = Client(SRIService.get_ws_test()[1])
        result_auto = client_auto.service.autorizacionComprobante(access_key)

        if result_auto[2] == "":
            print('Error SRI', 'No existe comprobante')
        else:
            autorizaciones = result_auto[2].autorizacion
            i = 0
            autorizado = False
            while i < len(authorizaciones):
                .....
                .....

```

Elaborado por: Karla Moyón, autora del presente proyecto

En el siguiente enlace se encuentra el código fuente completo de la clase [DocumentoXML](#)

- Generar reporte general

Para generar el reporte final, se creó un nuevo método en la clase sincronización, con el cual se obtiene el número de clientes, total de ventas, transacciones, consumo promedio y el total IVA ventas, considerando los datos almacenados de los último 30 días. Se utilizó la guía de Agregación de Django donde describe las formas en que se pueden generar y devolver valores agregados mediante consultas. A continuación, se presenta las librerías utilizadas para realizar las operaciones mencionadas anteriormente.

```
from django.db.models import Avg, Sum
import datetime as dt
from django.db.models import Count
```

Figura 59

Método generar_reporte_general de la Clase Sincronizacion.

```
def generar_reporte_general(self):
    actual = time.datetime.utcnow() #Fecha actual
    fecha_limite = actual - time.timedelta(days=30) #Fecha actual - 30 días
    actual = str(actual.strftime("%Y-%m-%dT%H:%M:%S"))
    fecha_limite = str(fecha_limite.strftime("%Y-%m-%dT%H:%M:%S"))

    num_clientes = Pedido.objects.filter(
        fecha_pedido__gte=str(fecha_limite),
        fecha_pedido__lte=str(actual)).values("id_consumidor_id").distinct().count()

    total_ventas = Pedido.objects.filter(
        fecha_pedido__gte=str(fecha_limite),
        fecha_pedido__lte=str(actual)).aggregate(Sum('valor_total'))

    transacciones = Pedido.objects.filter(
        fecha_pedido__gte=str(fecha_limite),
        fecha_pedido__lte=str(actual)).count()
```

```
consumo_promedio = Pedido.objects.filter(
    fecha_pedido__gte=str(fecha_limite),
    fecha_pedido__lte=str(actual)).aggregate(Avg("valor_total"))

total_impuestos= Pedido.objects.filter(
    fecha_pedido__gte=str(fecha_limite),
    fecha_pedido__lte=str(actual)).aggregate(Sum('total_impuestos'))

top_pedidos = Detalle_pedido.objects.annotate(pedidos=Count('nombre_prod')).order_by('-pedidos')[:3]
return num_clientes, total_ventas , transacciones,consumo_promedio,total_impuestos,top_pedidos
```

Elaborado por: Karla Moyón, autora del presente proyecto

En el siguiente enlace se encuentra el código fuente completo del método [generar_reporte_general](#)

Al finalizar con el Sprint 5, la construcción del sistema de facturación concluye satisfactoriamente, cumpliendo con el objetivo general del presente proyecto.

3.4.Pruebas de funcionamiento

En esta sección se muestra la ejecución de KV – Facturación, según los Requerimientos Funcionales y Requerimientos No Funcionales, planteados en el análisis de requerimientos.

En primer lugar, para la ejecución del sistema se debe ejecutar el siguiente comando:

```
$ python manage.py runserver
```

Antes de empezar con las pruebas de funcionalidad, cabe recordar que el usuario administrador es quien se encargará del registro del usuario del sistema, es decir del contribuyente.

Para acceder al panel de administración de Django se debe se debe ingresar la siguiente dirección en el navegador de su preferencia.

```
http://localhost:8000/admin/
```

El sistema mostrará, la siguiente pantalla, donde se ingresa con las credenciales del súper usuario.

Figura 60

Interfaz Administración de Django



The image shows the Django Admin login page. At the top, there is a dark blue header with the text "Administración de Django" in white. Below the header, the page has a light gray background. There are two input fields: the first is labeled "Email:" and contains the text "karla.vanessa@outlook.es"; the second is labeled "Contraseña:" and contains a series of dots. Below these fields is a blue button with the text "Iniciar sesión" in white.

Elaborado por: Karla Moyón, autora del presente proyecto

Al ingresar en el panel de administración el súper usuario puede observar, modificar, insertar y eliminar campos de los modelos creados en el sistema. De esta manera se procede a crear contribuyentes como se muestra en la siguiente figura.

Figura 61

Formulario para agregar contribuyente en el panel de Administración de Django

Usuario:	karla.vanessa@outlook.es
RUC:	1724997950001
Razón social:	MOYON RIVERA KARLA VANESSA
Nombre Comercial:	
URL Tienda:	https://avelectronics.cc/
Consumer key:	ck_31ee6694105f315f1a299b79ec4220e6f4c
Consumer secret:	cs_7a1c3f26c1da44186944f1ec8fe34e9d2c7
Dirección:	Nuevo Sur - Pedro Capiro Oe5-15 y Av. Tenien
<input type="checkbox"/> Contabilidad	
Teléfono:	0987190827
Firma electronica:	Actualmente: FIRMA_KARLA_VANESSA_MOYON_RIVERA.p12 Modificar: <input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado
Fecha caducidad:	13/10/2022 <input type="button" value="Hoy"/> Nota: Usted va 5 horas por detrás de la hora del servidor.

Elaborado por: Karla Moyón, autora del presente proyecto

Requerimientos Funcionales

1. Autenticación de usuarios

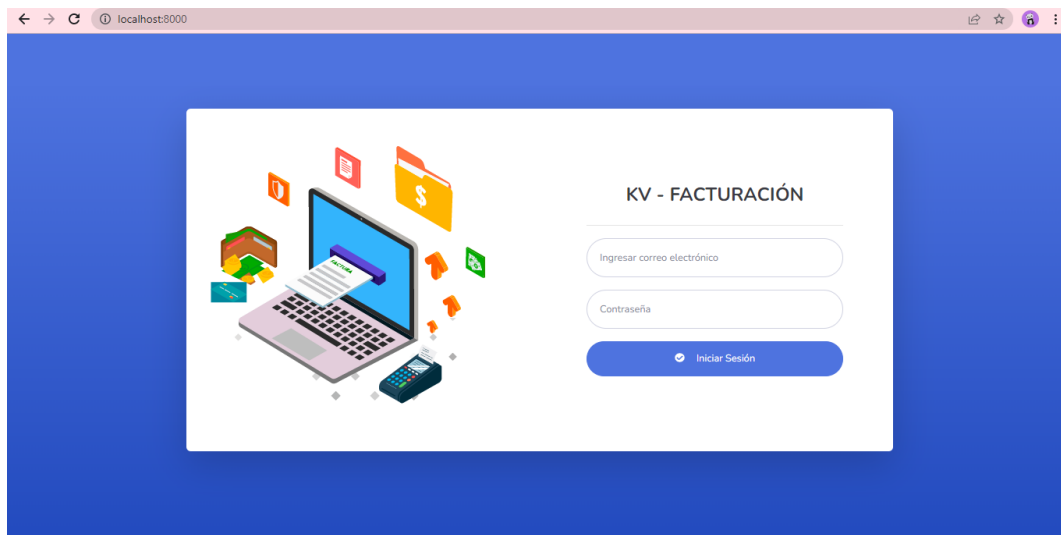
El usuario debe ingresar la siguiente dirección para acceder y utilizar la aplicación web de facturación electrónica.

`http://localhost:8000/`

En la siguiente figura se muestra la interfaz de inicio de sesión, donde el usuario debe ingresar su correo electrónico y contraseña.

Figura 62

Interfaz inicio de sesión



Elaborado por: Karla Moyón, autora del presente proyecto

Validaciones

- ✓ Todos los campos de entrada deben ser completados, caso contrario el sistema muestra un mensaje indicando que se deben completar los campos, como se indica en la siguiente figura.

Figura 63

Interfaz inicio de sesión – Validación campos completos

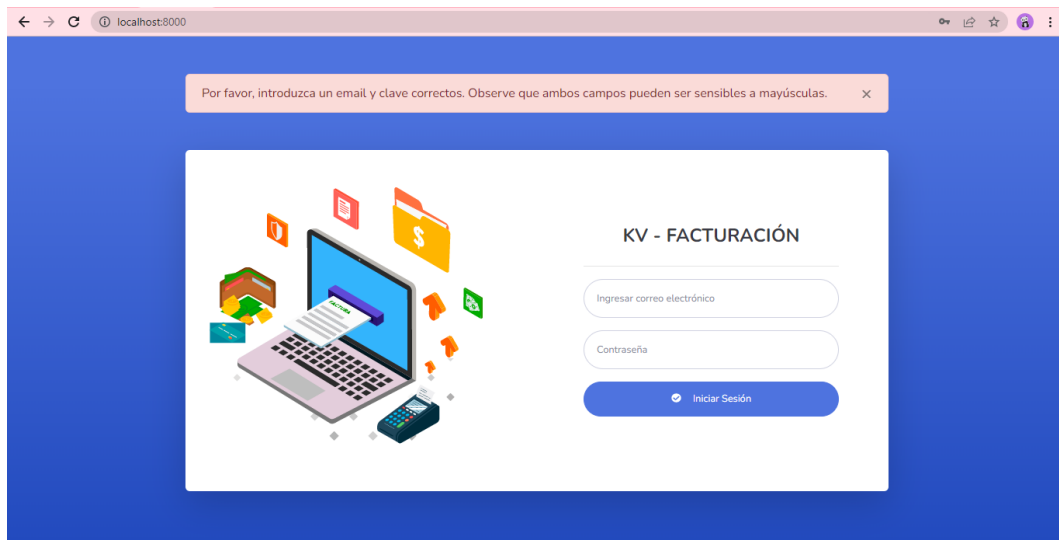


Elaborado por: Karla Moyón, autora del presente proyecto

- ✓ Si el correo electrónico o contraseña que se ingresó son incorrectos, el sistema mostrará un mensaje indicando el error.

Figura 64

Interfaz inicio de sesión – Validación campos correctos

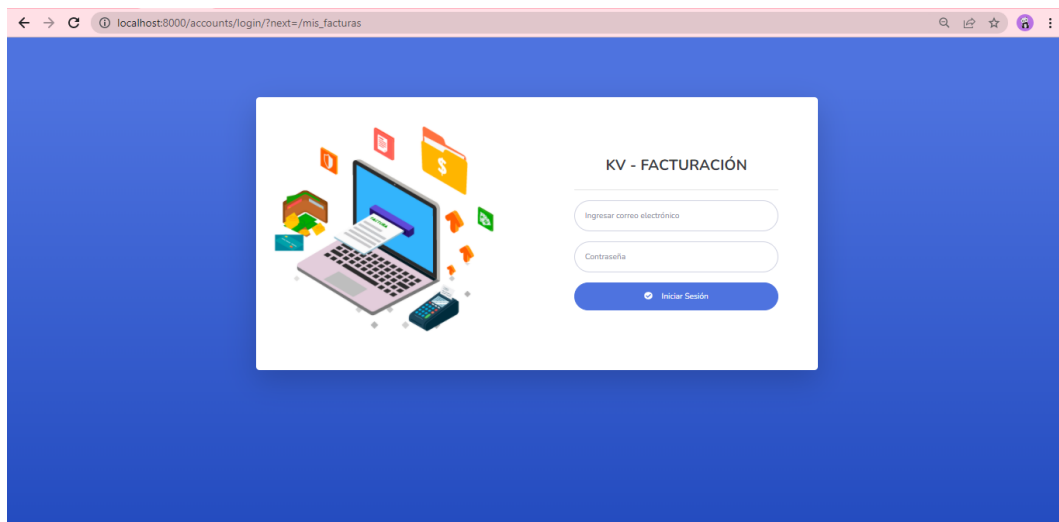


Elaborado por: Karla Moyón, autora del presente proyecto

- ✓ En caso que el usuario quiera acceder a otra interfaz del sistema, por medio de la dirección de la misma, Ejemplo: `http://localhost:8000/mis_facturas` sin haber iniciado sesión, KV – Facturación se encargará de re direccionar a la interfaz del inicio con la siguiente dirección
`http://localhost:8000/accounts/login/?next=/mis_facturas` como se indica en la siguiente figura.

Figura 66

Interfaz inicio de sesión – Validación ingreso a otra interfaz sin iniciar sesión



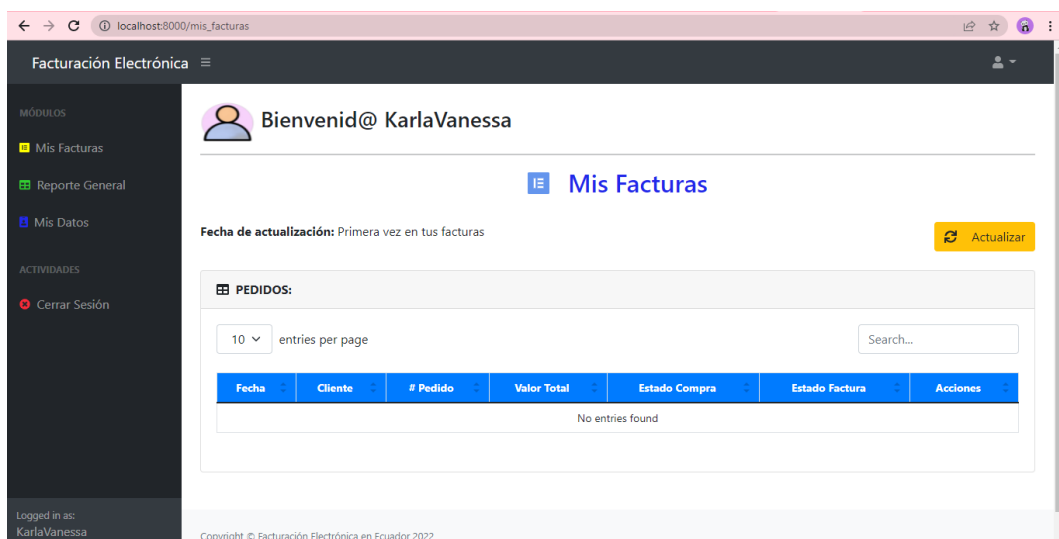
Elaborado por: Karla Moyón, autora del presente proyecto

2. Visualizar datos generales para la facturación

La primera vez que el usuario ingresa al sistema tiene que seleccionar el botón “Actualizar” para desplegar los pedidos de la tienda virtual y almacenarlos en la base de datos. En la siguiente figura se muestra la interfaz cuando no existen pedidos o es la primera vez que el usuario ingresa al sistema.

Figura 67

Interfaz “Mis facturas” sin pedidos por mostrar.

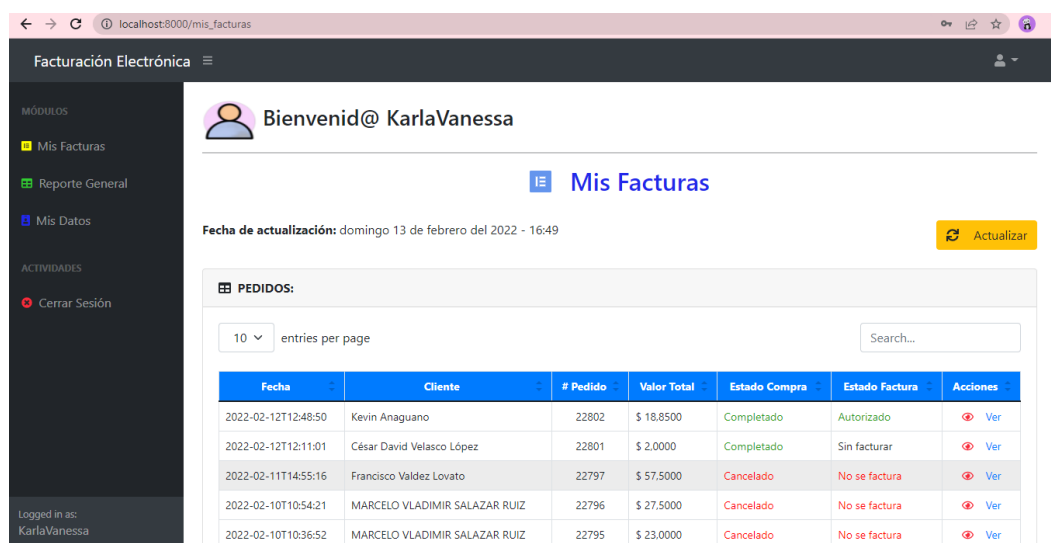


Elaborado por: Karla Moyón, autora del presente proyecto

En la siguiente figura se indica la interfaz “Mis facturas” donde se puede visualizar de forma general los datos del pedido, el estado de facturación en el que se encuentra y, además, como acción se puede visualizar la vista preliminar de la factura electrónica, de acuerdo al pedido seleccionado.

Figura 68

Interfaz “Mis facturas” con barra de menú



Facturación Electrónica

Bienvenid@ KarlaVanessa

Mis Facturas

Fecha de actualización: domingo 13 de febrero del 2022 - 16:49

Actualizar

PEDIDOS:

10 entries per page

Search...

Fecha	Cliente	# Pedido	Valor Total	Estado Compra	Estado Factura	Acciones
2022-02-12T12:48:50	Kevin Anaguano	22802	\$ 18,8500	Completado	Autorizado	Ver
2022-02-12T12:11:01	César David Velasco López	22801	\$ 2,0000	Completado	Sin facturar	Ver
2022-02-11T14:55:16	Francisco Valdez Lovato	22797	\$ 57,5000	Cancelado	No se factura	Ver
2022-02-10T10:54:21	MARCELO VLADIMIR SALAZAR RUIZ	22796	\$ 27,5000	Cancelado	No se factura	Ver
2022-02-10T10:36:52	MARCELO VLADIMIR SALAZAR RUIZ	22795	\$ 23,0000	Cancelado	No se factura	Ver

Logged in as: KarlaVanessa

Elaborado por: Karla Moyón, autora del presente proyecto

Cabe mencionar, que las interfaces que muestra el sistema a partir de iniciar sesión tiene una barra de menú que aparece al costado izquierdo, en el cual se indica los módulos que tiene acceso el usuario, el mismo que se minimiza para poder ver en pantalla completa la interfaz correspondiente como se indica en la siguiente figura.

Figura 69

Interfaz “Mis facturas” - barra de menú minimizada.

Bienvenid@ KarlaVanessa

Mis Facturas

Fecha de actualización: domingo 13 de febrero del 2022 - 16:49

Actualizar

PEDIDOS:

10 entries per page

Search...

Fecha	Cliente	# Pedido	Valor Total	Estado Compra	Estado Factura	Acciones
2022-02-12T12:48:50	Kevin Anaguano	22802	\$ 18,8500	Completado	Autorizado	Ver
2022-02-12T12:11:01	César David Velasco López	22801	\$ 2,0000	Completado	Sin facturar	Ver
2022-02-11T14:55:16	Francisco Valdez Lovato	22797	\$ 57,5000	Cancelado	No se factura	Ver
2022-02-10T10:54:21	MARCELO VLADIMIR SALAZAR RUIZ	22796	\$ 27,5000	Cancelado	No se factura	Ver
2022-02-10T10:36:52	MARCELO VLADIMIR SALAZAR RUIZ	22795	\$ 23,0000	Cancelado	No se factura	Ver

Elaborado por: Karla Moyón, autora del presente proyecto

3. Actualización de pedidos

En la siguiente figura se muestran pedidos antes de actualizar.

Figura 70

Interfaz "Mis facturas" pedidos antes de actualizar

Bienvenid@ KarlaVanessa

Mis Facturas

Fecha de actualización: domingo 13 de febrero del 2022 - 16:49

Actualizar

PEDIDOS:

10 entries per page

Search...

Fecha	Cliente	# Pedido	Valor Total	Estado Compra	Estado Factura	Acciones
2022-02-11T14:55:16	Francisco Valdez Lovato	22797	\$ 57,5000	En espera	Sin facturar	Ver
2022-02-10T10:54:21	MARCELO VLADIMIR SALAZAR RUIZ	22796	\$ 27,5000	Cancelado	Sin facturar	Ver
2022-02-10T10:36:52	MARCELO VLADIMIR SALAZAR RUIZ	22795	\$ 23,0000	Cancelado	Sin facturar	Ver
2022-02-09T17:50:06	Lino Toral	22794	\$ 37,0000	Completado	Sin facturar	Ver
2022-02-09T16:18:16	Steve Mantilla	22793	\$ 4,0000	Completado	Sin facturar	Ver
2022-02-09T12:51:33	José Molina	22792	\$ 29,2000	Completado	Sin facturar	Ver
2022-02-09T09:30:02	Paul Carpio	22790	\$ 9,0000	Completado	Sin facturar	Ver
2022-02-08T18:10:30	Omar Martínez	22789	\$ 22,6300	Completado	Sin facturar	Ver
2022-02-08T16:31:47	Christian Ninacuri	22788	\$ 143,0000	Completado	Sin facturar	Ver
2022-02-08T15:33:48	Pablo Morales	22787	\$ 10,2700	Completado	Sin facturar	Ver

Showing 1 to 10 of 97 entries

1 2 3 4 5 6 7 ... 10

Elaborado por: Karla Moyón, autora del presente proyecto

Al momento de actualizar, el sistema valida si han cambiado los estados de los pedidos, de ser el caso se modifica en la base de datos. Además, verifica si existen nuevos pedidos realizados

en la tienda virtual. Finalmente, KV - Facturación muestra mensajes, indicando que los pedidos han sido actualizados y agregados con la fecha en la que se realizó la actualización.

Figura 71

Interfaz "Mis facturas" pedidos actualizados



Elaborado por: Karla Moyón, autora del presente proyecto

Como se puede observar en la figura 70, el pedido 22797 tiene como estado de compra "En espera" y al momento de actualizar cambió a "Cancelado" y además se agregaron dos nuevos pedidos, como se indica en la siguiente figura.

Figura 72

Interfaz "Mis facturas" pedidos actualizados

The screenshot shows the 'Mis Facturas' interface with a table of orders. The table has columns: Fecha, Cliente, # Pedido, Valor Total, Estado Compra, Estado Factura, and Acciones. The row for order 22797 is highlighted with a red box, showing it is 'Cancelado'.

Fecha	Cliente	# Pedido	Valor Total	Estado Compra	Estado Factura	Acciones
2022-02-12T12:48:50	Kevin Anaguano	22802	\$ 18,8500	Completado	Sin facturar	Ver
2022-02-12T12:11:01	César David Velasco López	22801	\$ 2,0000	Completado	Sin facturar	Ver
2022-02-11T14:55:16	Francisco Valdez Lovato	22797	\$ 57,5000	Cancelado	Sin facturar	Ver
2022-02-10T10:54:21	MARCELO VLADIMIR SALAZAR RUIZ	22796	\$ 27,5000	Cancelado	Sin facturar	Ver
2022-02-10T10:36:52	MARCELO VLADIMIR SALAZAR RUIZ	22795	\$ 23,0000	Cancelado	Sin facturar	Ver


Elaborado por: Karla Moyón, autora del presente proyecto

4. Visualizar los datos de la factura electrónica

Al seleccionar la acción de “Ver” en la interfaz “Mis facturas”, se despliega una nueva ventana indicando la vista previa del pedido seleccionado, datos de la factura, emisor, consumidor, detalle pedido y totales. Cabe mencionar, que a mediados del mes de Enero del 2022, el SRI impuso una nueva ley del régimen tributario interno, llamado Régimen Simplificado para Emprendedores y Negocios Populares (RIMPE), en el cual de manera de resumen, indica que los negocios catalogados como emprendedores, seguirán con la misma estructura de facturación, es decir, con impuestos, sin embargo si es Popular no deberá emitir facturas, únicamente podrá emitir notas de venta, sin embargo, en caso de facturar, los productos no tendrán IVA y se debe especificar este detalle en la factura en campos adicionales.

Figura 73

Vista previa pedido – RIMPE: Populares


KV - Facturación

MOYON RIVERA KARLA VANESSA
Dirección: Nuevo Sur - Pedro Capiro Oe5- 15 y Av. Teniente Hugo Ortiz
Obligado llevar contabilidad: NO

Vista Previa Pedido: 22722
R.U.C: 1724997950001
FACTURA No: 001-002-000000001
NÚMERO DE AUTORIZACIÓN:
AMBIENTE: PRUEBAS
CLAVE DE ACCESO: 1302202201172499795000110010020000000014769883311

Fecha de Emisión: 13/02/2022
Razón social/ Nombres y Apellidos: Gardenia Galarza
Correo electrónico: gardeniagalarza@hotmail.com
Dirección: Monjas
RUC/CI: 1709975674
Teléfono: 2322290
[EDITAR](#)

Cod Principal	Cant	Descripción	Precio Unitario	Descuento	Precio total
RA021	1	MicroSD Kingston - 64GB	12,0000	0.00	12,0000
RP007	1	Raspberry Pi 4 Model B 4GB	120,0000	0.00	120,0000
RA005	1	Disipadores Raspberry Pi	4,5000	0.00	4,5000
RA012	1	Modulo cámara Raspberry Pi visión nocturna	35,0000	0.00	35,0000
RA007	1	Case Armable Acrílico	6,5000	0.00	6,5000
FE012	1	Fuente 5V 3A USB Tipo C	12,0000	0.00	12,0000

Información Adicional
RIMPE: Populares

Subtotal IVA 0% \$190,0000
Subtotal sin impuesto \$190,0000
VALOR TOTAL \$190,0000
VALOR A PAGAR \$190,0000


[REGRESAR](#) [FACTURAR](#)

Elaborado por: Karla Moyón, autora del presente proyecto

Figura 74

Vista previa pedido – Sin RIMPE

Vista Previa Pedido: 22579



KV - Facturación

MOYON RIVERA KARLA VANESSA

Dirección: Nuevo Sur - Pedro Capiro OeS-15 y Av. Teniente Hugo Ortiz

Obligado llevar contabilidad: NO

R.U.C: 1724997950001

FACTURA No: 001 002 000000001

NÚMERO DE AUTORIZACIÓN

AMBIENTE: PRUEBAS

CLAVE DE ACCESO: 1302202201172499795000110010020000000017549794814

Fecha de Emisión: 13/02/2022

Razón social/ Nombres y Apellidos: Xavier Chasi

Correo electrónico: xavier.faq@gmail.com

Dirección: El Condado

RUC/CI: 1753244324

Teléfono: 0983934881

[EDITAR](#)

Cod Principal	Cant	Descripcion	Precio Unitario	Descuento	Precio total
AS024	1	YUN Shield V1.1.6	40,0000	0.00	40,0000

Información Adicional

Subtotal 12%	\$40.0000
Subtotal sin impuestos	\$40.0000
IVA 12.00%	\$4.8000
VALOR TOTAL	\$44.8000
VALOR A PAGAR	\$44.8000

[REGRESAR](#)

[FACTURAR](#)

Elaborado por: Karla Moyón, autora del presente proyecto

5. Editar los datos de la factura electrónica

En la interfaz donde se visualiza la vista preliminar de la factura del pedido seleccionado, tiene una opción de editar los campos del consumidor, que al dar clic el sistema muestra una nueva ventana con la información del mismo y se podrá modificar. Al seleccionar la opción “Guardar” los datos serán editados en la base de datos.

En la siguiente figura se muestra la interfaz de editar consumidor, con sus respectivas opciones.

Figura 75

Interfaz editar consumidor

Facturación Electrónica

MÓDULOS

- Mis Facturas
- Reporte General
- Mis Datos

ACTIVIDADES

- Cerrar Sesión

Logged in as: KarlaVanessa

Actualizar Consumidor

REGRESAR GUARDAR

Identificación*

1709975674

Nombre*

Gardenia Galarza

Correo*

gardeniagalarza@hotmail.com

Dirección*

Elaborado por: Karla Moyón, autora del presente proyecto

6. Búsquedas para filtrar pedidos

En la Interfaz “Mis facturas”, muestra una tabla donde se indica los pedidos realizados en la tienda virtual, los mismos que serán filtrados de acuerdo a la entrada en el campo de búsqueda, cabe recalcar, que el sistema indicará los campos que el usuario desee ingresar.

Figura 76

Interfaz Mis facturas – filtro de pedidos que estén cancelado

PEDIDOS:

10 entries per page

cancelado

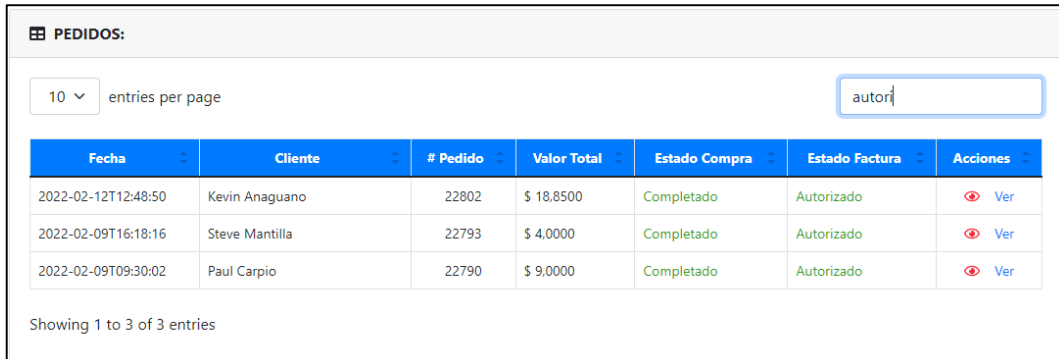
Fecha	Cliente	# Pedido	Valor Total	Estado Compra	Estado Factura	Acciones
2022-02-11T14:55:16	Francisco Valdez Lovato	22797	\$ 57,5000	Cancelado	No se factura	Ver
2022-02-10T10:54:21	MARCELO VLADIMIR SALAZAR RUIZ	22796	\$ 27,5000	Cancelado	No se factura	Ver
2022-02-10T10:36:52	MARCELO VLADIMIR SALAZAR RUIZ	22795	\$ 23,0000	Cancelado	No se factura	Ver
2022-02-07T11:39:36	Ines Navarrete	22782	\$ 22,0000	Cancelado	No se factura	Ver
2022-01-31T12:50:42	Luis Enrique Peñafiel Frias	22747	\$ 15,1700	Cancelado	No se factura	Ver
2022-01-23T21:19:39	Moises David Gutiérrez Peralta	22723	\$ 19,4000	Cancelado	No se factura	Ver
2022-01-07T21:45:53	Byron Palomeque	22570	\$ 3,3200	Cancelado	No se factura	Ver
2022-01-03T12:02:38	Fabian Rios Correa	22505	\$ 7,4600	Cancelado	No se factura	Ver

Showing 1 to 8 of 8 entries

Elaborado por: Karla Moyón, autora del presente proyecto

Figura 77

Interfaz Mis facturas – filtro de pedidos que estén autorizados



The screenshot shows a web interface titled 'PEDIDOS:'. It includes a search bar with the text 'autor' and a dropdown menu set to '10' entries per page. Below the search bar is a table with the following columns: Fecha, Cliente, # Pedido, Valor Total, Estado Compra, Estado Factura, and Acciones. The table contains three rows of data. At the bottom, it says 'Showing 1 to 3 of 3 entries'.

Fecha	Cliente	# Pedido	Valor Total	Estado Compra	Estado Factura	Acciones
2022-02-12T12:48:50	Kevin Anaguano	22802	\$ 18,8500	Completado	Autorizado	Ver
2022-02-09T16:18:16	Steve Mantilla	22793	\$ 4,0000	Completado	Autorizado	Ver
2022-02-09T09:30:02	Paul Carpio	22790	\$ 9,0000	Completado	Autorizado	Ver

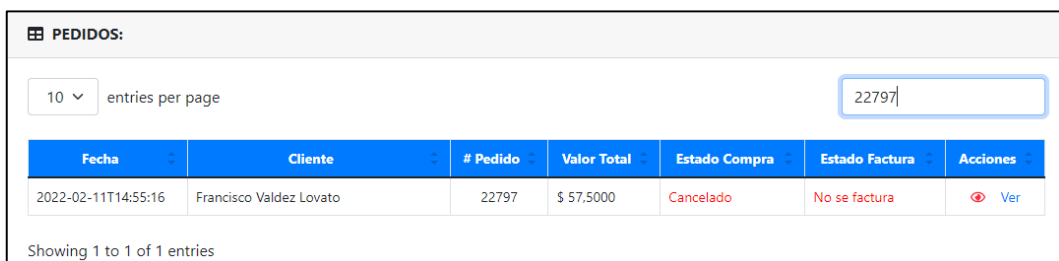
Elaborado por: Karla Moyón, autora del presente proyecto

7. Facturar pedido

Para cumplir con este requerimiento, se debe recordar que únicamente se podrá facturar los pedidos que tengan “Completado” el estado de la compra. Por ejemplo, en la figura 78 se muestra el pedido 22797, el cual tiene como estado de compra “Cancelado”

Figura 78

Interfaz Vista Previa Pedido con estado de compra cancelado



The screenshot shows the same 'PEDIDOS' interface, but the search bar now contains '22797'. The table now only contains one row of data for the order with ID 22797. The 'Estado Compra' is 'Cancelado' and the 'Estado Factura' is 'No se factura'. At the bottom, it says 'Showing 1 to 1 of 1 entries'.

Fecha	Cliente	# Pedido	Valor Total	Estado Compra	Estado Factura	Acciones
2022-02-11T14:55:16	Francisco Valdez Lovato	22797	\$ 57,5000	Cancelado	No se factura	Ver


En la siguiente figura se muestra el mensaje, indicando que el estado de la compra no es “Completado” y, además, no se muestra la opción de facturar.

Figura 79

Interfaz Vista Previa Pedido 22797 con estado de compra cancelado

Vista Previa Pedido: 22797

No se puede facturar. Estado del pedido no está completado



KV - Facturación

MOYON RIVERA KARLA VANESSA

Dirección: Nuevo Sur - Pedro Capiro Oe5-15 y Av. Teniente Hugo Ortiz

Obligado llevar contabilidad: NO

R.U.C	1724997950001
FACTURA No	001 002 000000001
NUMERO DE AUTORIZACIÓN	
AMBIENTE	PRUEBAS
CLAVE DE ACCESO	1402202201172499795000110010020000000015565588511

Fecha de Emisión: 14/02/2022

Razón social/ Nombres y Apellidos: Francisco Valdez Lovato

Correo electrónico: franciscovaldez96@gmail.com

Dirección: De los Almendros N54-09 Y Belladonas

RUC/CI: 1719131136

Teléfono: 0984574743

[EDITAR](#)

Cod Principal	Cant	Descripcion	Precio Unitario	Descuento	Precio total
AG001	1	UNO R3	17,5000	0.00	17,5000
AS024	1	YUN Shield V1.1.6	40,0000	0.00	40,0000

Información Adicional

RIMPE: Populares

Subtotal IVA 0%	\$57,5000
Subtotal sin impuesto	\$57,5000
VALOR TOTAL	\$57,5000
VALOR A PAGAR	\$57,5000

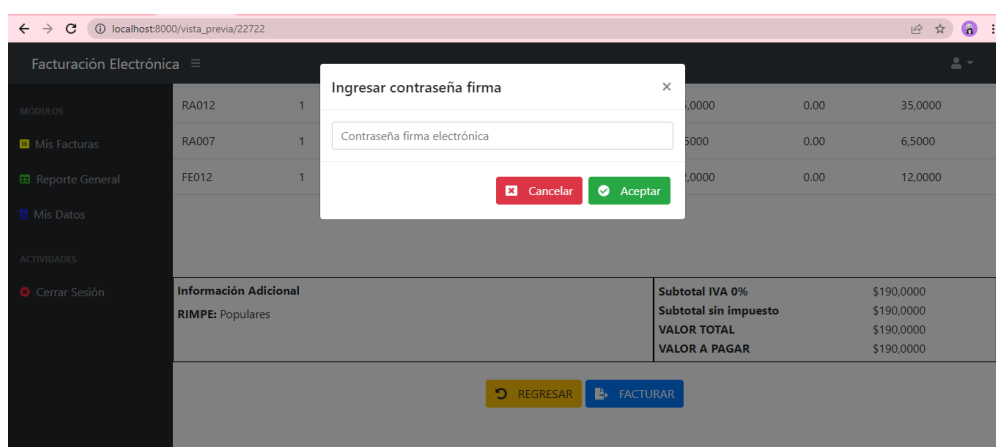
[REGRESAR](#)

Elaborado por: Karla Moyón, autora del presente proyecto

En la figura 73, se indica el Pedido 22722, al seleccionar la opción de facturar, el sistema muestra una ventana, en la cual solicita la contraseña de la firma electrónica como se indica en la siguiente figura.

Figura 80

Modal de ingreso de contraseña firma electrónica



The screenshot shows a web application interface for 'Facturación Electrónica'. A modal window titled 'Ingresar contraseña firma' is open, prompting the user to enter their 'Contraseña firma electrónica'. The modal has 'Cancelar' and 'Aceptar' buttons. In the background, a table lists items with codes (RA012, RA007, FE012), quantities (1), and prices. At the bottom, there are buttons for 'REGRESAR' and 'FACTURAR'.

Elaborado por: Karla Moyón, autora del presente proyecto

Por otro lado, como se indica en la figura 82, si la contraseña es correcta, se genera el documento XML, con su respectiva firma y es enviada al SRI para su validación y autorización. El sistema mostrará un mensaje indicado que el documento XML ha sido generado y el pedido se encuentra facturado. Además, como se puede observar, el usuario ya no podrá facturar y únicamente tendrá la opción de regresar a la interfaz “Mis facturas”. Finalmente, en la siguiente figura se muestra el estado de la factura del pedido 22722, el cual ha cambiado de “Sin facturar” a “Autorizado”.

Figura 83

Estado factura del pedido 22722



The screenshot shows a web interface titled "PEDIDOS:". It includes a search bar with "22722" entered and a dropdown for "entries per page" set to "10". Below is a table with the following data:

Fecha	Cliente	# Pedido	Valor Total	Estado Compra	Estado Factura	Acciones
2022-01-23T11:23:36	Gardenia Galarza	22722	\$ 190,0000	Completado	Autorizado	Ver

Showing 1 to 1 of 1 entries

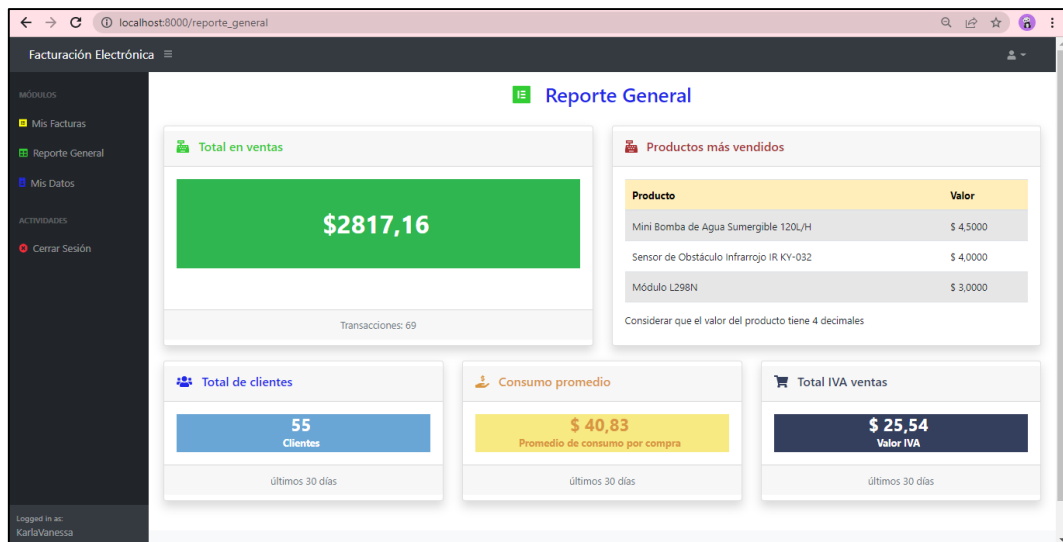
Elaborado por: Karla Moyón, autora del presente proyecto

8. Generar reporte general

En la interfaz de “Reporte General” se muestra, el total en ventas, número de transacciones, total de clientes, Consumo promedio y el total de IVA de acuerdo a las ventas realizadas, a través de la tienda virtual. Se tiene que considerar que los resultados presentados son de los últimos 30 días registradas en la base de datos. Además, muestra los 3 productos más vendidos.

Figura 84

Interfaz “Reporte General”



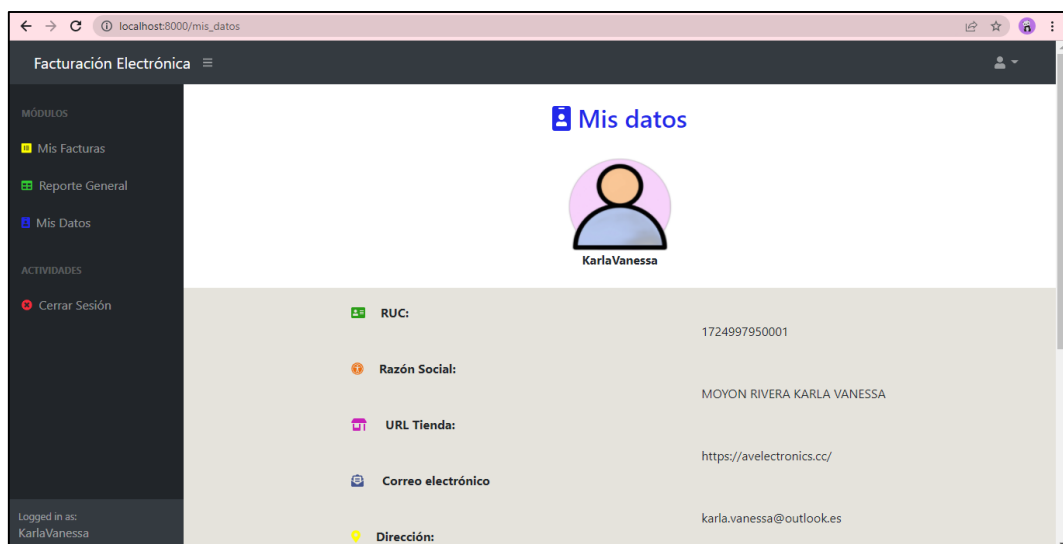
Elaborado por: Karla Moyón, autora del presente proyecto

9. Consultar datos usuario

En la figura 84 se muestra la interfaz “Mis datos”, en la cual se detalla la información del usuario contribuyente.

Figura 85

Interfaz “Mis datos”



Elaborado por: Karla Moyón, autora del presente proyecto

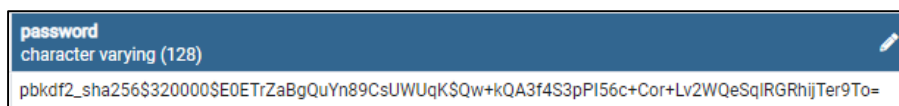
Requerimientos No Funcionales

Al finalizar con la construcción del sistema y de acuerdo a los requerimientos no funcionales descritos en el análisis de requerimientos se obtiene los siguientes resultados.

- ✓ Al tratarse de una facturación de pedidos en una tienda virtual, el sistema tiene las capacidades adecuadas para estar disponible las 24 horas al día, los 7 días de la semana.
- ✓ KV – Facturación cumple con el siguiente requerimiento No funcional: “El sistema deberá guardar la contraseña encriptada del usuario en la base de datos. Por otro lado, la contraseña de la firma electrónica no se almacena.”

Figura 86

Contraseña encriptada del usuario almacenada en la base de datos



Elaborado por: Karla Moyón, autora del presente proyecto

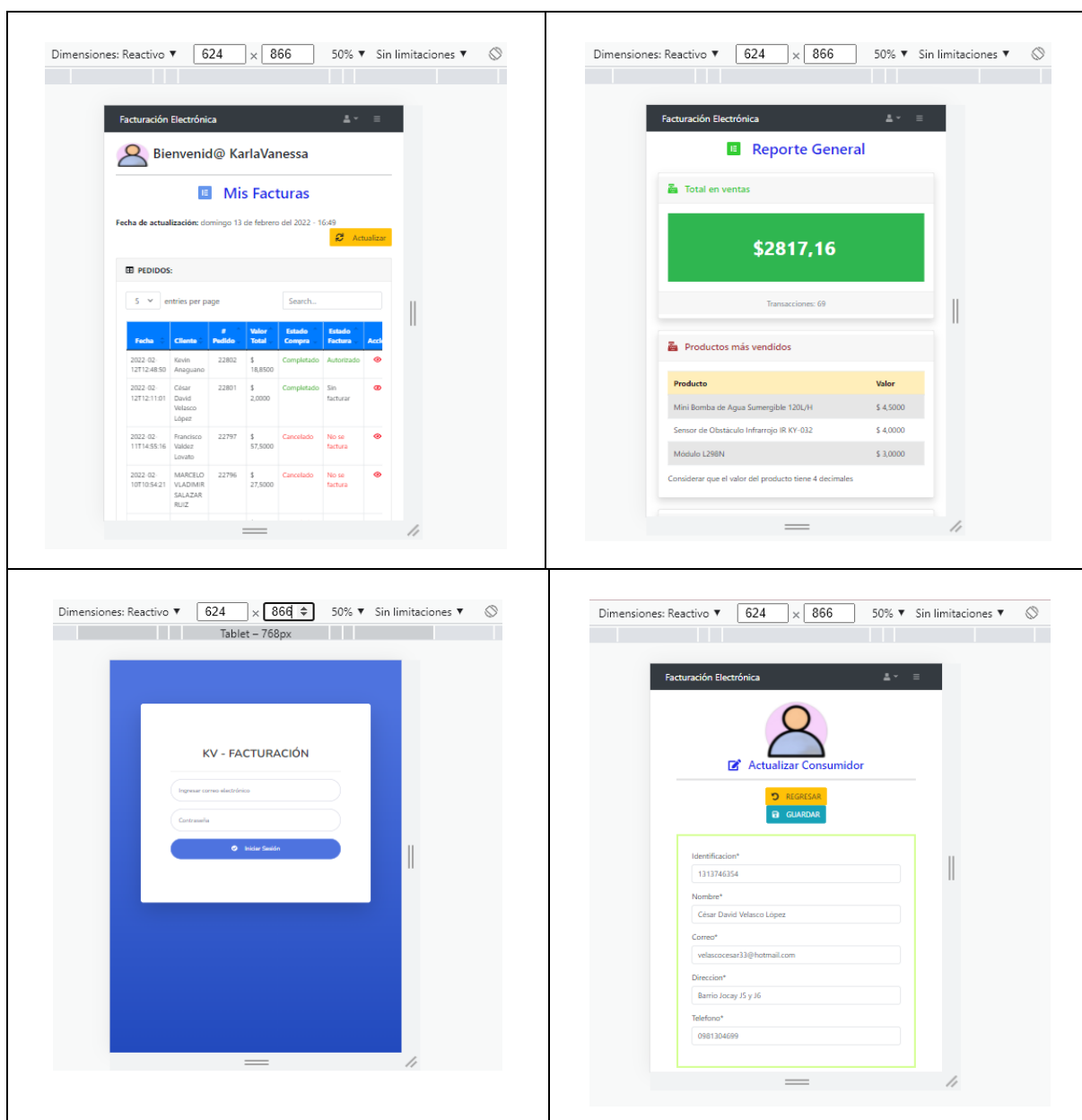
La contraseña de la firma electrónica no se encuentra almacenada en el sistema, por esta razón se solicita al momento de seleccionar la opción de “Facturar”

- ✓ Requerimiento No funcional: “El sistema debe proveer mecanismos de seguridad para que solo usuarios autorizados ingresen”. Para el cumplimiento de este requerimiento, se creó una interfaz de inicio de sesión para que solo tengan acceso los usuarios registrados anteriormente por el usuario administrador.
- ✓ El requerimiento No funcional: “El sistema debe tener un diseño de interfaces dinámicas e intuitivas para el usuario” al finalizar la construcción del sistema se realizó pruebas de funcionamiento, donde se obtuvo como resultado que el manejo de KV – facturación es intuitiva y fácil de manejar.

- ✓ Para el siguiente requerimiento no funcional “El sistema garantizará una adecuada visualización de las interfaces de usuario, es decir, debe contar con un diseño responsivo.” Se realizó las siguientes pruebas, donde en cada figura que se presenta a continuación cumple con el diseño responsivo, ya que, al modificar el tamaño de la ventana, los elementos de su interior se adaptan sin perder el estilo y formato planteado:

Figura 87

Interfaces principales con diseño responsivo



Elaborado por: Karla Moyón, autora del presente proyecto

4. CONCLUSIONES

- Se desarrolló un sistema con los requisitos iniciales del proyecto, dando como resultado la solución del problema mediante el análisis, diseño, construcción e implementación en el sistema, denominado KV - Facturación. Sin embargo, los cambios de políticas tributarias repentinos, afectaron significativamente al tiempo de construcción y desarrollo del sistema. Como es el caso del nuevo régimen, llamado RIMPE, aprobado a mediados del mes de Enero del 2022, el cual cambia la forma de facturar dependiendo del tipo de categoría asignado por el SRI al contribuyente.
- Las pruebas de funcionamientos demuestran que el sistema está en la capacidad de generar facturas electrónicas, a través de pedidos realizados a la tienda virtual del usuario.
- La aplicación web está adaptada para visualizar y utilizar desde cualquier navegador web actual, ya que, debido a su diseño responsivo, no afecta a la posición de los elementos del sistema.
- Django resultó ser la mejor opción para el desarrollo del proyecto, ya que permitió un rápido desarrollo y una implementación fácil, a través del editor de código, Visual Studio Code.
- Para la construcción del sistema se utilizó Bootstrap 4, ya que durante el desarrollo de las interfaces se pudo evidenciar que existen conflictos con las librerías utilizadas para Bootstrap 5. Como, por ejemplo, al momento de implementar modales y alertas.

5. RECOMENDACIONES

- Para versiones futuras del proyecto se recomienda generar una alerta temprana que notifique al usuario si la firma electrónica, está a punto de caducarse.
- Se recomienda crear una interfaz de registro de usuario para que cualquier persona que requiera del servicio de facturación electrónica pueda registrarse a sí mismo, sin la ayuda del administrador del sistema.
- Se recomienda crear una interfaz para que los usuarios puedan configurar su información tributaria, ya que, en la versión actual solo se muestra la información que ha sido almacenada por el administrador del sistema.
- En la interfaz de inicio de sesión se recomienda implementar una opción de recuperar la contraseña. En caso de que el usuario se olvide, se genere una nueva contraseña y se envíe mediante correo electrónico y se proceda al cambio de la misma.
- Al momento de enviar la factura electrónica al SRI, se recomienda que el usuario pueda descargar en formato pdf y se envíe mediante correo electrónico al consumidor.
- Se recomienda utilizar una versión jdk inferior a la 11, debido que, al momento de compilar el código de la firma electrónica con una versión actual, existe conflictos con las librerías de java.awt o java.swing. Solución encontrada (stackoverflow, 2020)

6. BIBLIOGRAFÍA

- AESA. (s.f.). *¿QUÉ ES Y CÓMO FUNCIONA LA FIRMA ELECTRÓNICA?* Obtenido de *¿QUÉ ES Y CÓMO FUNCIONA LA FIRMA ELECTRÓNICA?:*
https://sede.seguridadaerea.gob.es/SEDE_AESA/LANG_CASTELLANO/CONCEPTOS/QUE_FIRMA/
- Agencia de Tecnología y Certificación electrónica. (19 de Febrero de 2019). *Firma XAdES*. Obtenido de Firma XAdES:
https://www.accv.es/descargas/Setup/arangi/latest/arangi/firma_xades.html
- Banco Centra del Ecuador. (31 de 12 de 2020). *Art. 7 de la Ley Orgánica de Transparencia y Acceso a la Información Pública - LOTAIP*. Obtenido de Art. 7 de la Ley Orgánica de Transparencia y Acceso a la Información Pública - LOTAIP:
<https://www.bce.fin.ec/images/transparencia2021/literalf1enero2021.pdf>
- Barker, E., Barker, W., Burr, W., Polk, W., & Smid, M. (03 de 2007). *Recommendation for Key Management, Part 1: General (Revised)*. Obtenido de Recommendation for Key Management, Part 1: General (Revised):
<https://csrc.nist.gov/publications/detail/sp/800-57-part-1/revised/archive/2007-03-01>
- Cayambe Chicaiza, E. E. (2015). *Diseño e implementación de un sistema de facturación*. Obtenido de Diseño e implementación de un sistema de facturación:
<http://www.dspace.uce.edu.ec/bitstream/25000/5428/1/T-UCE-0011-221.pdf>
- Certificado digital*. (17 de 03 de 2008). Obtenido de Certificado digital:
http://enciclopedia.us.es/index.php/Certificado_digital
- Congreso Nacional. (17 de abril de 2002). *Ley de Comercio electrónico, firmas y mensajes de datos*. Obtenido de Ley de Comercio electrónico, firmas y mensajes de datos:
<https://www.telecomunicaciones.gob.ec/wp-content/uploads/downloads/2012/11/Ley-de-Comercio-Electronico-Firmas-y-Mensajes-de-Datos.pdf>
- Consejo de la Judicatura. (08 de 01 de 2019). *El Consejo de la Judicatura emitió más de 18 mil certificados de firma electrónica*. Obtenido de El Consejo de la Judicatura emitió más de 18 mil certificados de firma electrónica:
<https://www.funcionjudicial.gob.ec/index.php/es/saladeprensa/noticias/item/7074-el-consejo-de-la-judicatura-emitio%C3%B3-m%C3%A1s-de-18-mil-certificados-de-firma-electr%C3%B3nica.html>
- Datil. (2021). *Datil*. Obtenido de Datil: <https://datil.co/>
- De Souza, I. (12 de 07 de 2019). *XML: ¿qué es y para qué sirve este lenguaje de marcado?* Obtenido de XML: ¿qué es y para qué sirve este lenguaje de marcado?:
<https://rockcontent.com/es/blog/que-es-xml/>
- Especificación de requisitos de software*. (s.f.). Obtenido de Especificación de requisitos de software: https://www.ctr.unican.es/asignaturas/is1/plantilla_formato_ieee830.doc
- exes. (s.f.). *XML ¿Qué es?* Obtenido de XML ¿Qué es?: <https://www.mundolinux.info/que-es-xml.htm>

- Firma electrónica ¿Cómo obtenerla?* (2019). Obtenido de Firma electrónica ¿Cómo obtenerla?: <https://fabara.ec/blog-es/firma-electronica-como-obtenerla/>
- Guamanzara, C., & Guandinango, S. (2021). *Construcción e implementación de una solución libre para la automatización de procesos de importación de productos y la integración con Woocommerce. Caso de estudio: AV Electronics*. Quito: Universidad Politécnica Salesiana.
- Jaramillo, J. C. (2019). *ANF Autoridad de Certificación*. Obtenido de ANF Autoridad de Certificación: <https://citec.com.ec/socios/directorio/name/anf-autoridad-de-certificacion-1/>
- Jira Software. (s.f.). *¿Para qué sirve Jira?* Obtenido de ¿Para qué sirve Jira?: <https://www.atlassian.com/es/software/jira/guides/use-cases/what-is-jira-used-for#Jira-for-requirements-&-test-case-management>
- Naeem, T. (9 de Enero de 2020). *Qué es la integración de aplicaciones: definición, ventajas y desafíos*. Obtenido de Qué es la integración de aplicaciones: definición, ventajas y desafíos: <https://www.astera.com/es/type/blog/application-integration/>
- Pornin, T. (29 de Noviembre de 2011). *SHA, RSA and the relation between them*. Obtenido de SHA, RSA and the relation between them: <https://security.stackexchange.com/questions/9260/sha-rsa-and-the-relation-between-them>
- QualityConta. (2020). *¿Para qué se utiliza el Certificado Digital?* Obtenido de ¿Para qué se utiliza el Certificado Digital?: <https://www.qualityconta.com/actualidad/terminos-frecuentes/certificado-digital/>
- Red Hat. (08 de Mayo de 2020). *¿Qué es una API de REST?* Obtenido de ¿Qué es una API de REST?: <https://www.redhat.com/es/topics/api/what-is-a-rest-api>
- Registro Civil. (s.f.). *Firma Electrónica*. Obtenido de Firma Electrónica: <https://www.registrocivil.gob.ec/certificado-de-firma-electronica/>
- Scrum Org. (s.f.). *Scrum*. Obtenido de Scrum: <https://www.scrum.org/>
- Security Data. (2021). *Security Data*. Obtenido de Security Data: <https://www.securitydata.net.ec/conocenos/>
- SecurityData. (18 de Septiembre de 2019). *¿Qué es la firma electrónica y para qué sirve?* Obtenido de ¿Qué es la firma electrónica y para qué sirve?: <https://www.securitydata.net.ec/que-es-la-firma-electronica-y-para-que-sirve/>
- Sovos. (2021). *LEY DE FACTURA ELECTRÓNICA EN ECUADOR*. Obtenido de LEY DE FACTURA ELECTRÓNICA EN ECUADOR: <https://sovos.com/es/iva/reglas-fiscales/factura-electronica-ecuador/>
- SRI. (2020). *Ficha técnica comprobantes electrónicos esquema offline*. Obtenido de Ficha técnica comprobantes electrónicos esquema offline: <https://www.sri.gob.ec/o/sri-portlet-biblioteca-alfresco-internet/descargar/fb95cafc-a8ca-4a4c-afb6-12c4153165f0/FICHA%20TECNICA%20COMPROBANTES%20ELECTR%C3%93NICOS%20ESQUEMA%20OFFLINE.pdf>

- SRI. (s.f.). *Facturación electrónica*. Obtenido de Facturación electrónica:
<https://www.sri.gob.ec/facturacion-electronica>
- SRI. (s.f.). *Guía para contribuyentes*. Obtenido de Guía para contribuyentes:
[https://www.sri.gob.ec/o/sri-portlet-biblioteca-alfresco-internet/descargar/522a42ed-89c4-4ecd-978f-88c1afcd926c/Gu%C3%ADa%20para%20solicitar%20autorizaci%C3%B3n%20de%20emisi%C3%B3n%20de%20comprobantes%20electr%C3%B3nicos%2025042020%5B3052%5D%20\(1\)-1.pdf](https://www.sri.gob.ec/o/sri-portlet-biblioteca-alfresco-internet/descargar/522a42ed-89c4-4ecd-978f-88c1afcd926c/Gu%C3%ADa%20para%20solicitar%20autorizaci%C3%B3n%20de%20emisi%C3%B3n%20de%20comprobantes%20electr%C3%B3nicos%2025042020%5B3052%5D%20(1)-1.pdf)
- SRI. (s.f.). *IMPUESTO AL VALOR AGREGADO (IVA)*. Obtenido de IMPUESTO AL VALOR AGREGADO (IVA): <https://www.sri.gob.ec/impuesto-al-valor-agregado-iva/#%C2%BFcu%C3%A1l-es>
- SRI. (s.f.). *REGISTRO ÚNICO DE CONTRIBUYENTES*. Obtenido de REGISTRO ÚNICO DE CONTRIBUYENTES: <https://www.sri.gob.ec/RUC>
- stackoverflow*. (2020). Obtenido de stackoverflow:
<https://stackoverflow.com/questions/55571046/eclipse-is-confused-by-imports-accessible-from-more-than-one-module>
- startbootstrap. (2021). *SB Admin 2*. Obtenido de SB Admin 2:
<https://startbootstrap.com/theme/sb-admin-2>
- suds-community. (30 de Enero de 2022). *suds*. Obtenido de suds: <https://github.com/suds-community/suds/commits/master>
- Universidad Politécnica de Valencia. (2020). *¿Qué es un Certificado Digital?* Obtenido de ¿Qué es un Certificado Digital?:
<https://www.upv.es/contenidos/CD/info/711545normalc.html>
- Vega, R. M. (24 de Agosto de 2020). *Tipos de desarrollo de aplicaciones web: ejemplos y características*. Obtenido de Tipos de desarrollo de aplicaciones web: ejemplos y características: <https://profile.es/blog/desarrollo-aplicaciones-web/>
- Viafirma. (s.f.). *XAdES – Firma electrónica avanzada XML*. Obtenido de XAdES – Firma electrónica avanzada XML: <https://www.viafirma.com/faq/es/xades-firma-electronica-avanzada-xml/>
- WebEmpresa. (2019). *¿Qué es WordPress?* Obtenido de ¿Qué es WordPress?:
<https://www.webempresa.com/wordpress/que-es-wordpress.html>
- WordPress. (s.f.). *WooCommerce*. Obtenido de WooCommerce:
<https://es.wordpress.org/plugins/woocommerce/>
- WordPress. (s.f.). *WordPress*. Obtenido de WordPress: <https://es-es.wordpress.org/>

7. ANEXOS

ANEXO 1: SOLICITUD DE EMISIÓN PARA DOCUMENTOS ELECTRÓNICOS PARA PRUEBAS

SOLICITUD DE EMISIÓN PARA DOCUMENTOS ELECTRÓNICOS PARA PRUEBAS

SOLICITUD APROBADA

QUITO, 24 de enero del 2022

No. de Solicitud: 158995649

MOYON RIVERA KARLA VANESSA

RUC No: 1724997950001

Dirección: QUITO, PEDRO CAPIRO OE5-15 AV. TENIENTE HUGO ORTIZ, FRENTE AL CUERPO DE BOMBEROS, NUEVO SUR

De mi consideración, me permito informarle que la solicitud de **Emisión de Comprobantes Electrónicos para la ejecución en PRUEBAS** ha sido aprobada, para **Factura**.

El sujeto pasivo autorizado a emitir comprobantes electrónicos se compromete expresamente a proporcionar la información que sea requerida por el Servicio de Rentas Internas en dispositivos magnéticos o electrónicos, conforme a la ficha técnica que se encuentra disponible en la página Web del Servicio de Rentas Internas.

El uso incorrecto de los comprobantes de venta, retención y documentos complementarios firmados electrónicamente por parte del contribuyente, constituye una infracción tributaria, la misma que será sancionada conforme a las disposiciones legales vigentes.

El contribuyente deberá mantener al día sus obligaciones tributarias, actualizada la información en el RUC y la dirección de correo electrónico, la misma que servirá como medio de contacto; sus registros contables serán establecidos conforme la emisión de los documentos electrónicos emitidos. La información receptada y almacenada en la base del Servicio de Rentas Internas permanecerá por el periodo que determine la Administración Tributaria.

El sujeto pasivo es el responsable de la administración de los establecimientos, puntos de emisión y que la numeración sea secuencial para cada comprobante de venta, retención y documento complementario firmado electrónicamente.

Servicio de Rentas Internas

ANEXO 2: CÓDIGO FUENTE DEL PROYECTO EN REPOSITORIO EN GITHUB

El sistema KV – Facturación está publicado en el repositorio de GitHub, bajo licencia GNU, la cual es utilizada para software libre y código abierto. Además, garantiza a los usuarios la libertad de usar, modificar y compartir el software. Cabe aclarar que, si se realiza alguna modificación en el código, es obligatorio publicar los cambios como software libre.

En el siguiente enlace se puede acceder al código fuente, el mismo que contiene un README, donde se indica como descargar y utilizar el sistema.

<https://github.com/karlaVane/KV-Facturacion>