



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Desarrollo de una API para la emisión de facturas electrónicas usando el enfoque en DevOps que permite la implementación de un pipeline de Integración y Entrega Continua CI/CD.

Molina Naranjo, Karla Lizeth y Pachacama Cajamarca, Alex Fabricio

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Trabajo de Integración Curricular, previo a la obtención del Título de Ingeniero de Software

Ing. López Otañez, Edgar Rubén

16 de agosto del 2023

Latacunga

Reporte de verificación de contenido



Plagiarism report

TESINA MOLINA - PACHACAMA Versio...

Scan details

Scan time: August 15th, 2023 at 18:19 UTC
 Total Pages: 69
 Total Words: 17036

Plagiarism Detection

Types of plagiarism	Words
Identical	3% 518
Minor Changes	1.7% 287
Paraphrased	1.9% 331
Orphaned Words	0% 0

AI Content Detection

Text coverage
 AI text
 Human text

Plagiarism Results: (107)

- Características del producto Amazon CloudWatch - ...** 0.6%
<https://aws.amazon.com/es/monitoring/cloudwatch/>
 Saltar al contenido principal Haga clic aquí para volver a la página de inicio de Amazon Web S...
- CR031821560-Mares-Banuelos-Adopcion.pptx** 0.5%
<https://www.farbit.com/vep/contenido/adjuntos/2021/05/18/031821560...>
 ADICIÓN DE CLOUD COMPUTING COMO HERRAMIENTA DE RESPALDO DE TRABAJO E INFORMACIÓN EN LOS DESPACHOS CONSULARES DE LA CIUDAD ...
- CIELO VIRTUAL. HISTORIA | by Luisacontreras | Med...** 0.5%
<https://medium.com/@luisacontreras/cielo-virtual-80e5c0e50e5c>
 Luisacontreras
 Open in app Sign up Sign in Write Sign up Sign in CIELO VIRTUAL
 luisacontreras - Follow 3 min read ...

Certified by

About this report
<https://copyleaks.com>

copyleaks.com

Ing. López Otañez, Edgar Rubén

C. C. 0502002413



Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Certificación

Certifico que el trabajo de integración curricular: "Desarrollo de una API para la emisión de facturas electrónicas usando el enfoque en DevOps que permite la implementación de un pipeline de Integración y Entrega Continua CI/CD." fue realizado por los señores **Molina Naranjo, Karla Lizeth y Pachacama Cajamarca, Alex Fabricio**, el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizada en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Latacunga, 16 de agosto del 2023

Ing. López Otañez, Edgar Rubén

C. C. 0502002413



Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Responsabilidad de Autoría

Nosotros, **Molina Naranjo, Karla Lizeth y Pachacama Cajamarca, Alex Fabricio**, con cédulas de ciudadanía n° 1751812726 y 1727655944, declaramos que el contenido, ideas y criterios del trabajo de integración curricular: **Desarrollo de una API para la emisión de facturas electrónicas usando el enfoque en DevOps que permite la implementación de un pipeline de Integración y Entrega Continua CI/CD** es de mi/nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 16 de agosto del 2023

Molina Naranjo, Karla Lizeth

C.C.: 1751812726

Pachacama Cajamarca, Alex Fabricio

C.C.: 1727655944



Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Autorización de Publicación

Nosotros, **Molina Naranjo, Karla Lizeth y Pachacama Cajamarca, Alex Fabricio**, con cédulas de ciudadanía n° 1751812726 y 1727655944, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: **Desarrollo de una API para la emisión de facturas electrónicas usando el enfoque en DevOps que permite la implementación de un pipeline de Integración y Entrega Continua CI/CD en el Repositorio Institucional**, cuyo contenido, ideas y criterios son de mi/nuestra responsabilidad

Latacunga, 16 de agosto del 2023

Molina Naranjo, Karla Lizeth

C.C.: 1751812726

Pachacama Cajamarca, Alex Fabricio

C.C.: 1727655944

Dedicatoria

El presente trabajo quiero dedicar a mis padres Fausto y Mónica con todo mi cariño y gratitud, quienes han sido el pilar fundamental en mi vida y me han apoyado incondicionalmente durante el transcurso de mi formación profesional. Gracias a su amor incondicional, sus enseñanzas y sus esfuerzos por motivarme a alcanzar mis metas en mi vida.

A Mis abuelitos que, con su cariño, sus consejos y sus experiencias me han apoyado, a mi abuelita Etelvina, que no me acompaña en este momento, la llevo siempre en mi corazón.

A mi hermano Brayan y mi cuñada Pamela, que me han dado el privilegio de ser ejemplo de mi querida sobrina Doménica, a quien espero guiar y aconsejar, como lo han hecho sus padres conmigo en los buenos y malos momentos.

A mi compañero leal e incondicional Miko, quien ha estado acompañándome todas las noches en el transcurso de mi formación académica y ser una parte importante en cada logro de mi vida.

A Alex quien ha sido mi compañero de viaje, en mi vida y durante mi formación académica, por la paciencia, el compromiso, dedicación, conocimiento, esfuerzo y que realizamos para llegar a este punto de nuestras vidas, superando cada obstáculo que se nos presentó, dejándome grandes enseñanzas, experiencias y aventuras, gracias de todo corazón por este logro que alcanzamos juntos y esperando verle progresar en su vida profesional.

A Pamela quien ha estado ahí apoyándome desde mi infancia, brindándome su amistad y apoyo a pesar de las dificultades que se nos ha presentado.

Y, por último, pero no menos importantes para mis queridas amigas de colegio Alison y Mishell, con quienes he compartido este camino, sin importar la distancia hemos estado juntas, los momentos y recuerdos han sido una parte importante para completar este logro, porque no faltó las salidas, las risas y las conversaciones profundas. Apoyándonos mutuamente en cada paso que hemos dado y celebrándolos.

Molina Naranjo, Karla Lizeth

Dedicatoria

Dedico el presente trabajo:

A mis ñaños Pato, Maury, Byron, Vero y Jesica, por su inquebrantable apoyo, amor y paciencia a lo largo de este camino ya que, con sus consejos y experiencias me ayudan a fortalecer mi carácter como persona y profesional siendo así una fuente de inspiración constante para dar lo mejor de mí y superarme día tras día.

A mis abuelitos que sin importar la circunstancia siempre están para mí, con su amor y sabiduría apoyándome incondicional.

A Karla por ser mi compañera de vida, por el cariño, apoyo, risas, llantos y aventuras que hemos compartido durante todos estos años, por las noches de estudio interminables y sobre todo por ayudarme a crecer como persona y profesional superando cada adversidad que se nos presentaba.

A Jairo y Paola por siempre creer en mí y apoyarme en los momentos más difíciles de la vida, por las conversaciones, risas y experiencias compartidas.

Finalmente, pero no menos importante a mi madre Jeaneth que pese a las diferencias existentes siempre ha tratado estar para mí, por el esfuerzo que ha realizado al ayudarme a cumplir esta meta.

Esta tesis está dedicada a todos ustedes, quienes han sido los cimientos de mi éxito. Sin su presencia en mi vida, este logro carecería de sentido. ¡Muchas gracias!

Pachacama Cajamarca, Alex Fabricio

Agradecimiento

Nos gustaría expresar nuestros más sinceros agradecimientos:

A la Universidad de las Fuerzas Armadas ESPE y sus docentes, por impartirnos conocimientos como personas y profesionales, mediante sus experiencias y sus enseñanzas para crecer día tras día durante el transcurso de nuestra formación profesional.

A la empresa “Corporación Wolf S.A.”, por acogernos y brindarnos esta oportunidad, de colaborar y ayudar en el desarrollo de la API de facturación electrónica y formarnos como profesionales.

A nuestro asesor y colaborador por parte de la empresa Ing. José Ochoa, por estar siempre guiándonos con sus conocimientos y experiencias sin importar la hora o el día, enseñándonos a superar los retos que tiene un profesional, durante todo el proceso del desarrollo de la API de facturación electrónica.

A nuestra familia más cercana por apoyarnos incondicionalmente y creer en nosotros.

A nuestro director de tesis Ing. Rubén López por dedicarnos un poco de su tiempo para revisar avances y atender consultas sobre todo el proceso de titulación, para llevar a cabo satisfactoriamente el trabajo de integración curricular.

Molina Karla, Pachacama Alex

ÍNDICE DE CONTENIDOS

Carátula.....	1
Reporte de verificación de contenido.....	2
Certificación.....	3
Responsabilidad de Autoría	4
Autorización de Publicación.....	5
Dedicatoria.....	6
Dedicatoria.....	7
Agradecimiento.....	8
Índice de contenidos.....	9
Índice de tablas.....	12
Índice de figuras	13
Resumen	14
Abstract.....	15
Capítulo I: Introducción	16
Antecedentes.....	16
Justificación e importancia.....	17
Alcance.....	17
Planteamiento del problema.....	17
Formulación del problema a resolver	18
Objetivos	18
<i>Objetivo general.....</i>	18

	10
<i>Objetivos específicos.....</i>	<i>19</i>
Capítulo II: Marco Teórico.....	20
Facturación electrónica.....	20
<i>Comprobantes de venta.....</i>	<i>20</i>
<i>Facturación electrónica en el ecuador.....</i>	<i>21</i>
Backend.....	25
<i>Herramientas de desarrollo</i>	<i>25</i>
<i>Tecnologías y frameworks.....</i>	<i>26</i>
<i>Librerías</i>	<i>27</i>
Computación en la nube	29
<i>¿Qué es DevOps?</i>	<i>29</i>
<i>Pipeline CI/CD</i>	<i>30</i>
<i>GitHub actions</i>	<i>31</i>
<i>AWS.....</i>	<i>31</i>
Metodología.....	33
<i>SCRUM.....</i>	<i>33</i>
<i>Pair programming (Programación en Parejas).....</i>	<i>34</i>
Capítulo III: Implementación del Sistema	35
Fase de análisis y planificación	35
<i>Descripción del sistema.....</i>	<i>35</i>
<i>Descripción de la metodología</i>	<i>36</i>

<i>Especificación de requisitos</i>	37
Fase de diseño	40
<i>Arquitectura lógica</i>	41
<i>Descripción de la estructura</i>	41
<i>Diseño detallado</i>	42
Fase de desarrollo	45
Capítulo IV: Marco de Resultados	93
Rendimiento de despliegue.....	95
Análisis comparativo de costos.....	104
Conclusiones y recomendaciones	106
Conclusiones.....	106
Recomendaciones.....	108
Bibliografía	109
Anexos	115

ÍNDICE DE TABLAS

Tabla 1 <i>Roles designados</i>	36
Tabla 2 <i>Product Backlog</i>	37
Tabla 3 <i>Estándar de codificación Base de Datos</i>	44
Tabla 4 <i>Estándar de codificación Backend</i>	45
Tabla 5 <i>Sprint Backlog Global</i>	46
Tabla 6 <i>Sprint Backlog 02</i>	59
Tabla 7 <i>Sprint Backlog 03</i>	64
Tabla 8 <i>Sprint Backlog 04</i>	68
Tabla 9 <i>Sprint Backlog 05</i>	73
Tabla 10 <i>Sprint Backlog 06</i>	80
Tabla 11 <i>Sprint Backlog 07</i>	86
Tabla 12 <i>Métricas del Rendimiento de Entrega</i>	94
Tabla 13 <i>Rendimiento de Entrega del Proyecto Desarrollado</i>	94
Tabla 14 <i>Resultados del Caso 1</i>	98
Tabla 15 <i>Resultados del Caso 2</i>	98
Tabla 16 <i>Resultados del Caso 3</i>	99
Tabla 17 <i>Resultados del Caso 4</i>	99
Tabla 18 <i>Resultados del Caso 5</i>	100
Tabla 19 <i>Resultados del Caso 6</i>	100
Tabla 20 <i>Resultados del Caso 1 CD</i>	101
Tabla 21 <i>Resultados del Caso 2 CD</i>	101
Tabla 22 <i>Comparación de Tiempo entre el Despliegue Manual y Automatizado</i>	102

ÍNDICE DE FIGURAS

Figura 1 <i>Proceso de Emisión y Autorización de Comprobantes Electrónicos</i>	23
Figura 2 <i>Arquitectura Lógica del Backend</i>	41
Figura 3 <i>Diagrama de Despliegue</i>	43
Figura 4 <i>Entrevista con el cliente para definir los requerimientos</i>	51
Figura 5 <i>Implementación de la base de datos</i>	52
Figura 6 <i>Creación del servicio para crear la categoría</i>	61
Figura 7 <i>Creación del controlador para crear la categoría</i>	61
Figura 8 <i>Consumo del endpoint de crear la categoría</i>	62
Figura 9 <i>Creación del guardian para la ApiKey</i>	65
Figura 10 <i>Utilización del ApiKey</i>	66
Figura 11 <i>Creación del controlador para crear la factura</i>	69
Figura 12 <i>Consumo del endpoint para crear factura</i>	70
Figura 13 <i>Creación del controlador para crear la nota de crédito</i>	75
Figura 14 <i>Consumo del endpoint para crear la nota de crédito</i>	76
Figura 15 <i>Creación del controlador para el paginado y filtrado de los documentos</i>	77
Figura 16 <i>Consumo del endpoint para filtrar los documentos</i>	77
Figura 17 <i>Documentación generada con OpenApi</i>	82
Figura 18 <i>Validación de cálculos de la factura</i>	83
Figura 19 <i>Creación del test unitario del DNI</i>	88
Figura 20 <i>Reporte de test con el pipeline</i>	89
Figura 21 <i>Documentación desplegada en AWS</i>	90
Figura 22 <i>Flujo de trabajo de los pipelines de GitHub Actions</i>	91
Figura 23 <i>Pruebas de Aceptación de la API y sus múltiples endpoints</i>	91
Figura 24 <i>Casos del CI</i>	97
Figura 25 <i>Gráfico Despliegue Manual vs Automatizado</i>	103

Resumen

En el presente trabajo, se desarrolla una API con el uso de framework NestJS con TypeORM para un sistema de facturación electrónica, usando el enfoque DevOps que permite la implementación de pipelines de integración y entrega continua CI/CD para la empresa “Corporación Wolf S.A.”. Para el desarrollo del sistema se realiza un análisis del proceso de facturación electrónica en el Ecuador a través del SRI (Servicio de Rentas Internas). Para el desarrollo del backend se emplea una metodología ágil que está dividida en 3 fases: análisis y planificación, diseño y desarrollo. En el análisis y planificación se define los miembros del proyecto y el rol que va a cumplir cada uno, también se realiza la especificación de requisitos a través de historias de usuario, obteniendo el product backlog. En la fase de diseño se define la arquitectura lógica, descripción de la estructura, modelo entidad relación, diagrama de paquetes del sistema, diagrama de despliegue, estándares de codificación y modelo físico de la base de datos. En la fase de desarrollo se realizan 7 sprint con una duración de 315 horas en total. Por otro lado, se evalúa el rendimiento del desempeño de los pipelines de despliegue de forma empírica, tomando dos objetos de estudio: manualmente y automatizada mediante los pipelines, obteniendo como resultado una disminución de tiempo 325,02 s (5,25 min) por despliegue menor que la forma manual. Además, se hace un análisis comparativo de los costos en la empresa, tomando en cuenta al servicio externo de facturación electrónica actual y el desarrollado, obteniendo una reducción de egresos en la suscripción anual de \$340, por otro lado, se hace una proyección de 3 usuarios que podrían contratar el servicio de facturación, por lo cual se obtendría un ingreso neto de \$650 anualmente, concluyendo que el sistema desarrollado aporta a la economía, aumenta su competitividad y mejora los procesos de la empresa.

Palabras clave: backend, facturación electrónica, pipelines CI/CD, GitHub Actions, NestJS

Abstract

In this work, an API is developed with the use of NestJS framework with TypeORM for an electronic invoicing system, using the DevOps approach that allows the implementation of integration pipelines and continuous delivery CI/CD for the company "Corporation Wolf S.A.". For the development of the system, an analysis of the electronic invoicing process in Ecuador through the SRI (Internal Revenue Service) is performed. For the development of the backend, the agile methodology is used, which is divided into 3 phases: analysis and planning, design and development. In the analysis and planning phase, the members of the project and the role that each one will play are defined, as well as the specification of requirements through user stories, obtaining the product backlog. In the design phase, the logical architecture, structure description, entity-relationship model, system package diagram, deployment diagram, coding standards and physical model of the database are defined. In the development phase, 7 sprints are carried out with a total duration of 315 hours. On the other hand, the performance of the deployment pipelines is evaluated empirically, taking two objects of study: manually and automated by means of the pipelines, obtaining as a result a decrease of time 325.02 s (5.25 min) per deployment less than the manual way. In addition, a comparative analysis of the company's costs is made, taking into account the current external electronic invoicing service and the one developed, obtaining a reduction of expenses in the annual subscription of \$340, on the other hand, a projection is made of 3 users that could contract the invoicing service, which would result in a net income of \$650 annually, concluding that the developed system contributes to the economy, increases its competitiveness and improves the company's processes.

Keywords: backend, electronic invoicing, CI/CD pipelines, GitHub Actions, NestJS

Capítulo I

Introducción

Antecedentes

En este mundo globalizado, las entidades comerciales han implementado nuevas prácticas para el desarrollo de sus actividades correspondientes, debido a los avances tecnológicos que se están convirtiendo en un rol importante para las mismas, gracias a eso han podido modernizar sus procesos tradicionales de dos formas: mediante la adquisición de software de terceros, o la implementación de software desarrollado internamente.

En la actualidad, algunas entidades comerciales han optado la forma de desarrollar su propio software, el cual proporciona un control total, porque puede adaptarse a las exigencias específicas para agilizar de manera más efectiva sus procesos. Entre uno de sus procesos esta: la facturación electrónica, que es un componente fundamental para las entidades comerciales, en donde ha adquirido una importancia esencial, ya que agiliza los procesos de tributación que se lo realizan en conjunto con el Servicio de Rentas Internas (SRI), este busca eliminar el almacenamiento físico de certificados y preservar el medio ambiente al reducir el excesivo uso de papel (Vergara Sandoval, 2017).

En el año 2013, el Servicio de Rentas Internas (SRI) de Ecuador lanza oficialmente a la factura electrónica como una estrategia institucional para modernizar la administración tributaria a favor de reducir la evasión y simplificar el cumplimiento a los contribuyentes. En el año 2015, se da el primer ingreso masivo de contribuyentes a este nuevo sistema. Sin embargo, a pesar de que ya existían usuarios para el sistema, un 4% de las facturas que se emiten en el Ecuador aún son físicas, esto pese a que más del 96% de lo facturado se hace de manera virtual, por eso en noviembre del 2022, el SRI da a conocer que los contribuyentes están obligados a emitir facturas electrónicas (Merlo, 2006).

La manera en la que se ha venido agilizando el proceso de facturación electrónica en los últimos años, ha sido mediante el uso de prácticas y herramientas tecnológicas que han ido evolucionando en el mercado, aplicándolo para el desarrollo de APIs (Interfaz de Programación de Aplicaciones) que han permitido que las empresas integren la emisión de facturas electrónicas en sus procesos de negocios y sistemas informáticos.

Justificación e importancia

En la actualidad el SRI (Servicio de Rentas Internas) que gestiona la política tributaria en el Ecuador, conforme a la Resolución **NAC-DGERCGC22-00000024** a partir del 30 de noviembre del 2022, exige la emisión de facturas electrónicas, esto conlleva a que las empresas usen sistemas de facturación electrónica con el objetivo de proporcionar a las empresas una mejora en la eficiencia y productividad, reduciendo los errores, facilitando el cumplimiento fiscal, integrándose fácilmente con otros sistemas empresariales, aumentando la seguridad y reduciendo los costos asociados a la emisión de facturas en papel (Servicio de Rentas Internas, 2022).

Alcance

La finalidad del presente trabajo es el desarrollo del backend que implica crear los múltiples endpoints de APIs necesarios para realizar operaciones de facturación electrónica, el cual se utilizara para la emisión y autorización de: facturas, comprobantes de retención, guía de remisión, notas de crédito y débito. Basándose en el enfoque de DevOps con la implementación de pipelines CI/CD para automatizar los procesos de integración y entrega continua.

Planteamiento del problema

El proceso de facturación es una parte fundamental de la gestión financiera de cualquier empresa. Sin embargo, la facturación en papel puede ser un proceso costoso, lento y propenso a errores. Además, las facturas en papel no siempre son fáciles de rastrear y pueden

extraviarse o dañarse durante el proceso de envío, y algunos clientes pueden ser reacios a aceptar facturas electrónicas debido a preocupaciones sobre la seguridad de los datos y la privacidad. Es por eso que en Ecuador el SRI (Servicio de Rentas Internas) que gestiona la política tributaria, actualmente, exige a las entidades comerciales que cumplan con las normas establecidas que consiste en emitir facturas electrónicas, debido al crecimiento del mercado en nuestro país (Lara Freire, 2015).

Conforme a la Resolución **NAC-DGERCGC22-00000024**, a partir del 30 de noviembre del 2022, las empresas que están en la capacidad de desarrollar su propio sistema de facturación electrónica, pueden integrarle a su pila de aplicaciones, lo que eliminará la dependencia de un proveedor externo (Servicio de Rentas Internas, 2022).

Esto ha llevado a que la Corporación Wolf S.A. una empresa ecuatoriana de ingeniería electrónica y tecnologías de la información, que tiene 3 líneas de negocio: Robot Wolf, Wolf Electronics y Corporación Wolf S.A. (Ochoa & Bone, 2022), tenga la necesidad de reemplazar la facturación física, desarrollando su propio servicio de sistema de facturación electrónica

Formulación del problema a resolver

- ¿Cómo aporta el desarrollo de una API de facturación electrónica a la empresa “Corporación Wolf S.A.” mediante la implementación de pipelines de integración y entrega continua CI/CD para la automatización del despliegue?

Objetivos

Objetivo general

- Desarrollar una API para la emisión de documentos electrónicos usando el enfoque en DevOps que permite la implementación de un pipeline de integración y entrega continua CI/CD.

Objetivos específicos

- Conocer el estado del arte acerca del proceso de facturación electrónica, DevOps, CI/CD y computación en la nube.
- Implementar la API con sus múltiples endpoints.
- Validar los resultados, analizar errores de la API y sus múltiples endpoints.
- Generar tesina de grado.

Capítulo II

Marco Teórico

Facturación electrónica

La facturación electrónica es una forma de emitir comprobantes digitales legales por medio de sistemas, los cuales cumplen con los requisitos y reglamentos establecidos por parte de una institución legislativa tributaria vigente en cada país, garantizando su integridad y autenticidad (Mamian, 2021b), teniendo el mismo efecto y propósito que un documento en papel, por tanto, es una alternativa legal (Gobierno de España, s. f.) y autorizada para las entidades comerciales quienes quieran reemplazar su facturación tradicional.

Comprobantes de venta

Son documentos emitidos por las entidades comerciales que entregan cuando se transfieren bienes, se prestan bienes o realización de otras transacciones con otros tributos (Delgado, 2010), cumpliendo con los requisitos legales gravados por la Ley de Régimen Tributario Interno (Vergara Sandoval, 2017). Entre estos comprobantes se encuentran los siguientes:

Factura. Documento que está dirigido a empresas o personas naturales quienes cumplen con los requisitos necesarios para acceder al crédito tributario y realizar operaciones de exportación (Pazmiño, 2015).

Comprobantes de retención. Documento que acreditan las retenciones de impuestos realizadas por los agentes de retención en cumplimiento de lo dispuesto en la Ley de Régimen Tributario Interno (Delgado, 2010).

Guía de remisión. Documento que sustenta el traslado de mercaderías por cualquier motivo dentro del territorio nacional (Delgado, 2010).

Nota de débito. Documento que se emitirá para el cobro de intereses de mora y para recuperar costos y gastos, incurridos por el vendedor con posterioridad a la emisión de la factura (Delgado, 2010).

Nota de crédito. Documento que se emitirán para anular operaciones, aceptar devoluciones y conceder descuentos o bonificaciones (Delgado, 2010).

Facturación electrónica en el Ecuador

En el Ecuador se comienza a escuchar el termino de “Facturación Electrónica” desde el año 2002, debido al entorno en que se encontraba, ya les permitía hacer la realización del envío y recepción del documento para una mayor agilización del mismo (Mamian, 2021a).

Siete años después según la Resolución **NAC-DGERCGC09-00288** de Mayo en el 2012, el director general del servicio de Rentas Internas mediante resoluciones, circulares o disposiciones de carácter general y obligatorio, necesarias para la aplicación de las normas legales y reglamentarias que según el Art. 8 “Ley de Comercio Electrónico, Firmas Electrónicas y Mensajes de Datos”, se resuelve expedir las normas para la emisión de comprobantes de venta, documentos complementarios y comprobantes de retención como mensajes de datos, esto se lo realizará previo a la solicitud del sujeto pasivo, quien deberá contar con su firma electrónica vigente y válida para un período igual o mayor al de la indicada autorización (Servicio de Rentas Internas, 2009).

Posteriormente, en marzo del año 2012, según la Resolución **NAC-DGERCGC12-00105**, se resuelve expedir las normas para el nuevo esquema de emisión de comprobantes de venta, retención y documentos complementarios mediante mensajes de datos (Comprobantes Electrónicos). Además, el SRI mediante su página web institucional www.sri.gob.ec, proporciona un software libre donde cada sujeto pasivo puede realizar pruebas piloto, teniendo en cuenta que los comprobantes electrónicos que emitan en versión prueba no tienen validez tributaria y no sustentan costos y gastos, siendo así, una opción beneficiosa para las empresas

y negocios para que puedan entrar o estar inmersos con las nuevas tecnologías (Servicio de Rentas Internas, 2012).

Posteriormente, para el año 2013, el SRI determina según la Resolución **NAC-DGERCG13-00236**, un cronograma que establece la obligatoriedad a los sujetos pasivos que estén, obligados a emitir comprobantes electrónicos y que se aplicará a las transacciones que estén, sujetas a sustentación de crédito tributario del Impuesto al Valor Agregado para la emisión de comprobantes electrónicos (Servicio de Rentas Internas, 2013).

En el año 2014, conforme a la Resolución la **NAC-DGERCG14-00157**, el SRI determina que la facturación electrónica, debe ser implementada en la mayoría de los sectores económicos que se enlisten (Mamian, 2021a).

En el año 2018, se reforma la Resolución **NAC-DGERCGC18-00000191**, por la Resolución **NAC-DGERCGC18-00000431**, se explica cuáles, deben ser los contribuyentes quienes están obligados a emitir comprobantes electrónicos entre ellos esta (Servicio de Rentas Internas, 2018):

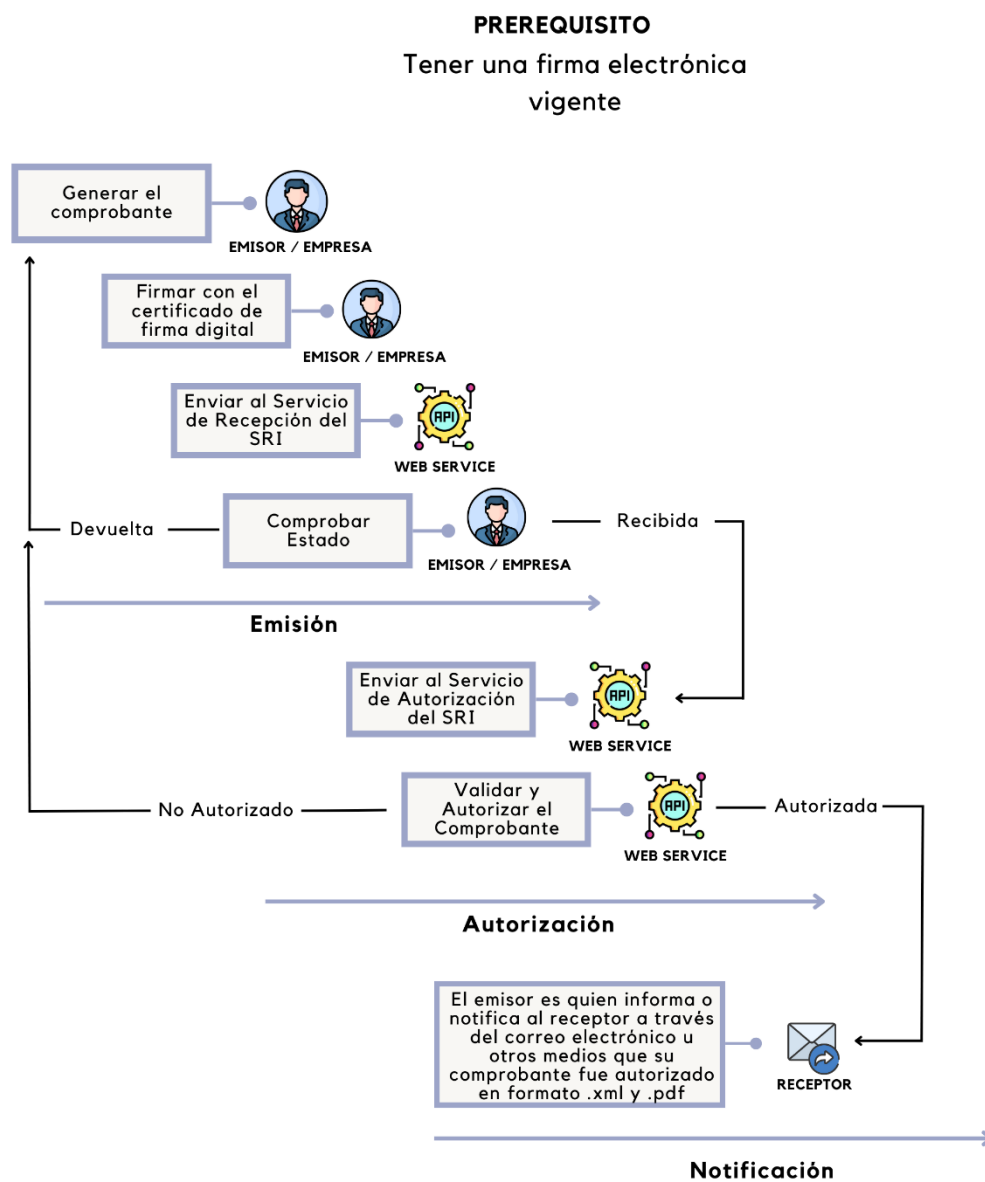
- Quienes realizan actividades de comercialización de maquinaria pesada
- Quienes sean personas naturales y declaren transacciones superiores a 1.000 dólares
- Quienes realicen actividades relacionadas con el petróleo o gas licuado. (Mamian, 2021a)

Mas adelante en el año 2022, conforme a la Resolución **NAC-DGERCGC22-00000024**, a partir del 30 de noviembre del 2022, exige la emisión de facturas electrónicas obligatorias a todos los sujetos pasivos y sectores económicos quienes están obligados a emitirlos, por lo que conlleva a muchas de las empresas que están en la capacidad de hacerlo, desarrollen su propio sistema de facturación electrónica que se integre a su pila de aplicaciones, lo que eliminará la dependencia de un proveedor externo (Servicio de Rentas Internas, 2022).

Proceso para la emisión y autorización de comprobantes.

Figura 1

Proceso de Emisión y Autorización de Comprobantes Electrónicos



Nota. El proceso del SRI, quien recibe y autoriza los comprobantes electrónicos.

Formato del comprobante (XML). El SRI proporciona una estructura específica y detallada para cada uno de los comprobantes electrónicos, que cada emisor deberá cumplir para que el SRI pueda hacer el proceso descrito anteriormente. A continuación, se muestran los formatos:

- Anexo 1: Formato XML Factura
- Anexo 2: Formato XML Comprobante de Retención
- Anexo 3: Formato XML Guía de Remisión
- Anexo 4: Formato XML Nota de Crédito
- Anexo 5: Formato XML Nota de Débito (Servicio de Rentas Internas, 2023).

Servicios del SRI para la emisión y autorización. El SRI proporciona Web Services para los dos ambientes disponibles que son: Pruebas y Producción, los cuales son estandarizados con protocolos de seguridad y certificados SSL. Estos tienen la función para la recepción y autorización de los comprobantes electrónicos, sea individual o por lotes (Servicio de Rentas Internas, 2023).

Para la utilización del Web Services de pruebas el Sri proporciona los siguientes:

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

Para la utilización del Web Services de producción el Sri proporciona los siguientes:

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

Los Web Services se utilizan de manera asíncrona, una vez que se haya receptado el comprobante se obtendrá dos estados: recibida y rechazada, cuando este en recibida, se debe esperar un tiempo determinado para que consuma la otra WS de autorización mediante la clave de acceso y se podrá obtener 3 estados: procesamiento (PPR), autorizado (AUT), no autorizado (NAT) (Servicio de Rentas Internas, 2023).

Backend

Se hace referencia al código fuente que no puede ser visualizado por parte del usuario o cliente, se ejecuta en un servidor de aplicaciones. Se encarga de la lógica del negocio, conexión a la base de datos, creación de servicios web y seguridad. Es decir, accede a la información que el cliente solicita por medio de la aplicación mediante la utilización de los servicios web (APIs) (Ochoa & Bone, 2022).

Herramientas de desarrollo

Visual Studio Code. Es un editor de código que combina varios lenguajes de programación, permitiéndole ser un potente y robusto IDE (Integrated Development Environment) de desarrollo (Microsoft, 2023) Open Source con una compatibilidad en los sistemas operativos más usados, siendo así un software muy versátil, debido a las infinidad de extensiones que se adaptan a cada campo y ámbito dentro del desarrollo de software.

Dbeaver. Es un software universal de Open Source, que ayuda a la gestión de bases de datos de manera más usable y profesional para los usuarios, siendo multiplataforma y compatible con una gran variedad de base de datos. Teniendo un potente editor SQL, que proporcionan muchas funcionalidades como la administración, monitoreo de base de datos, migración de datos y esquemas (DBeaver Documentation, 2020).

PgAdmin4. Es un software Open Source, que gestiona la base de datos de código PostgreSQL, proporcionando una usabilidad e interfaz gráfica simple y entendible (Page, s. f.).

Insomnia. Es un software de Open Source que facilita la interacción, el diseño, la depuración y prueba de APIs. Presenta características que le hacen intuitiva para el usuario, como las variables de entorno, generación de código y los asistentes de autenticación (Kong Inc., 2021).

GitHub. Es una plataforma de alojamiento de código, que permite la colaboración y control de versiones del proyecto desde cualquier lugar del mundo. Entre sus elementos más esenciales están los repositorios, ramas, confirmaciones y solicitudes de extracción (GitHub Inc., 2023a).

Excalidraw. Es una aplicación web Open Source de una pizarra virtual donde se puede dibujar diagramas, bocetos, etc., con una interfaz sencilla, adaptable e intuitiva para el usuario (*excalidraw*, 2020/2023).

Draw.io. Es una aplicación de editor en línea donde se puede crear diagramas, bocetos con una integración a varias plataformas y aplicaciones (JGraph Ltd, 2005).

Tecnologías y frameworks

PostgreSQL. Es un sistema de base de datos Open Source y relacional de objetos que utiliza el lenguaje SQL, teniendo una sólida arquitectura para la integridad de datos, es extensible, tolerante a fallas, ofreciendo soluciones innovadoras (The PostgreSQL Global Development Group, 1996).

NestJS. Ayuda a construir aplicaciones del lado del servidor de manera eficiente y escalable Node.js, utilizando JavaScript siendo compatible con TypeScript combinando con elementos de POO (Programación Orientada a Objetos), PF (Programación Funcional) y PRF (Programación Reactiva Funcional). Además, utiliza servicios robustos HTTP (Mysliwiec, 2017). Siendo un framework fácil de utilizarlo en cualquier proyecto, porque es fácil de aprender y dominar, además, proporciona una extensa documentación y tiene su propia comunidad.

TypeORM. Es un ORM (Mapeo Relacional de Objetos), es decir, se refiere al mapeado de entidades con tablas de bases de datos, automatizando la conversión de objeto a tabla y tabla a objeto, proporcionando comandos CLI simples para las migraciones. TypeORM es muy flexible porque permite actualizar, mantener y reutilizar el código, abstrayendo el sistema DB de la aplicación. Es compatible con múltiples bases de datos usando TypeScript y JavaScript (Tutorials Point (India), s. f.).

Typescript. Se encuentra como “en una relación inusual con JavaScript, ofrece todas las características de JavaScript y una capa adicional sobre estas: el sistema de tipo de TypeScript” (Microsoft, 2012). Es decir, podemos utilizar conocimientos de JavaScript y aplicarlo en TypeScript, este “añade características únicas como la tipificación estática, las interfaces, las clases y mucho más” (Rolfo, 2022).

Librerías

bcrypt. Su función principal es hashear las contraseñas según varios estándares criptográficos (NPM Inc., 2022c).

bwip-js. Según (NPM Inc., 2023b) se define como un escritor de código de barras en JavaScript puro que utiliza módulos de codificación para varios estándares de códigos de barras diferentes, generando la imagen que contiene el código de barra de tipo png.

cookie-parser. Permite guardar credenciales dentro de las cookies del navegador (NPM Inc., 2021c).

crypto-js. La página oficial de CryptoJS (NPM Inc., 2021a) define como “la colección de algoritmos criptográficos estándar y seguros implementados en JavaScript utilizando las mejores prácticas y patrones”

dayjs. Permite formatear la fecha en diferentes formatos de acuerdo a la necesidad del cliente (NPM Inc., 2023c).

exceljs. Permite el diseño, creación y lectura de archivos de hoja de cálculo xlsx y json (NPM Inc., 2021b).

hashids. La página oficial de Hashids (Brzóska, s. f.) define como “una pequeña biblioteca de código abierto que genera identificadores cortos, únicos y no secuenciales a partir de números”

node-forge. NPM (NPM Inc., 2022b) define como la “implementación totalmente nativa del protocolo TLS en JavaScript, un conjunto de utilidades criptográficas y un conjunto de herramientas para desarrollar aplicaciones web que utilicen muchos recursos de red”.

nodemailer. Es una librería Open Source desarrollada por la comunidad que permite el envío de correos electrónicos fácilmente, está integrada con Office365 y Google Workspace (Reinman, s. f.).

passport-jwt. Permite el almacenamiento de un json web token para el control de autorización y acceso a las solicitudes (NPM Inc., 2022d).

passport-local. Permite autenticar usando un nombre de usuario y contraseña a la aplicación, este va de la mano con el passport-jwt (NPM Inc., 2022a).

pg. Permite realizar una conexión directa a la base de datos PostgreSQL de una manera sencilla.

puppeteer. Permite la creación de PDF a través de capturas de pantalla de páginas de un navegador virtual basado en Chrome (Google Inc., 2023).

Redocly CLI. La página oficial (Redocly Inc., 2023) se define como “una herramienta de línea de comandos de código abierto para trabajar con descripciones de Open API, portales de desarrolladoras y otras operaciones de ciclo de vida de API”

swagger-jsdoc. Permite generar la documentación de las APIs en un estándar OpenAPI en formato json (NPM Inc., 2023a).

xml2js. Permite la transformación de objetos XML a objetos JavaScript y viceversa (Kubica, 2023).

Computación en la nube

Según (Arias, 2015) define a la computación en la nube como “un sistema de computación distribuido orientado al consumidor, que consiste en una colección de ordenadores virtualizados e interconectados que son suministrados dinámicamente y presentados como uno o más recursos computacionales unificados, conforme acuerdo de nivel de servicio negociado entre el proveedor de servicios y el consumidor”. Por otro lado, (Google Inc., s. f.) define a la computación en la nube como “la disponibilidad bajo demanda de recursos de computación como servicios a través de Internet. Esta tecnología evita que las empresas tengan que encargarse de aprovisionar, configurar o gestionar los recursos y permiten que paguen únicamente por lo que usen”.

¿Qué es DevOps?

GitLab (GitLab B.V., 2022) define a DevOps como “una combinación de desarrollo de software (dev) y operaciones (ops). Se define como una metodología de ingeniería de software cuyo objetivo es integrar el trabajo de los equipos de desarrollo y los equipos de operaciones facilitando una cultura de colaboración y responsabilidad compartida”. En cambio (López, 2021) define como “un enfoque de colaboración entre todos los miembros de los equipos de TI para

solucionar los problemas que surgen para conseguir despliegues de software frecuentes y confiables en entornos de producción, a través de la cultura, la automatización, la medición y el intercambio”.

Pipeline CI/CD

La Integración continua permite hacer cambios al código de un proyecto, luego esos cambios se deben integrar en su rama respectiva, mediante las herramientas de automatización se integra directamente los cambios sin tener que hacerlo manualmente, debido a que hay cambios diariamente por lo que este proceso ahorra mucho tiempo. Todo el proceso se realiza a través de un servidor de automatización en la nube (Vinueza, s. f.).

La entrega continua es una extensión del CI que integra automáticamente los cambios del proyecto en el repositorio, pueden ser implementados directamente en el servidor de prueba o producción mediante instrucciones manuales, donde la funcionalidad y la interfaz son visibles cada que se realiza la implementación en la rama principal (Vinueza, s. f.).

El despliegue continuo viene después de la integración continua y la entrega continua, aquí la automatización se utiliza para el despliegue, pero debe tomar medidas de seguridad similares a las de un operador de DevOps. Esta fase de implementación automática no es utilizada por todos los equipos de desarrollo o empresas que utilizan integración y entrega continua, porque depende en gran medida de la lógica empresarial si esta decide llevar el proceso de implementación de forma automática o no. Esto se lo realiza, para que si llegan nuevos cambios que puedan romper o presente fallos en la aplicación, se realice con éxito el despliegue (Vinueza, s. f.).

Un pipeline de CI/CD es un conjunto de pasos que se inician cuando se activa el trabajo en un proyecto. El archivo que se configura en el proyecto define el flujo de trabajo, que debe ser revisado por un software antes de ser desplegado de manera automatizada (Devops Latam, 2021; Vinueza, s. f.).

GitHub actions

Es una plataforma de integración continua y entrega continua que permite automatizar y ejecutar el flujo de trabajo de compilación, prueba y despliegue mediante cada solicitud o evento que se haga en el repositorio, además, proporciona máquinas virtuales en los sistemas operativos más conocidos que pueden correr o ejecutar su flujo de trabajo, o en la infraestructura en la nube (GitHub Inc., 2023b).

Según (Huerlo, 2021) se dice que “automatiza, personaliza y ejecuta flujos de trabajo de desarrollo de software directamente en tu repositorio con GitHub Actions. Puede descubrir, crear y compartir acciones para realizar cualquier trabajo que quiera, incluido CI/CD, y combinar acciones en un flujo de trabajo completamente personalizado”

AWS

La página oficial de AWS (Amazon Web Services Inc., 2023c) define que “Amazon Web Services (AWS) incluye más de 200 servicios integrales de centros de datos a nivel global, cada uno de los cuales se ocupa de un área de funcionalidad, por lo que hace que, las empresas emergentes que crecen más rápido, las compañías más grandes y los organismos gubernamentales líderes, estén usando AWS para reducir costos, aumentar su agilidad e innovar de forma más rápida. AWS brinda una gran variedad de servicios y flexibilidad necesaria para decidir cómo quiere administrar la infraestructura de AWS, sin embargo, puede resultar complicado saber qué servicios deben usarse y como aprovisionarlos” en cada proyecto.

Elastic Beanstalk. Es un servicio de gestión de infraestructura web. Gestiona e implementa las aplicaciones fácilmente sin preocuparse por su infraestructura subyacente, reduciendo la complejidad, sin limitar las opciones, ni el control, gestionando automáticamente la información de aprovisionamiento, equilibrio de carga, escalado y estado de la aplicación. Admite varios lenguajes para las aplicaciones y crea varios recursos como instancias de Amazon EC2 para ejecutar la aplicación dependiendo de la cantidad de solicitudes que exista en el momento para mantener un buen rendimiento constante (Amazon Web Services Inc., 2023d).

Relational Database Service. Es un servicio web que proporciona una base de datos relacional estándar con capacidad asequible, escalable y maneja tareas comunes que gestiona a la base de datos. Es sencillo de configurar, ejecutar, escalar y es compatible con varios gestores de base de datos (Amazon Web Services Inc., 2023b).

S3 Bucket. Un bucket es un contenedor de objetos almacenados en Amazon S3, puede almacenar cualquier cantidad de objetos en un bucket y puede tener hasta 100 en la cuenta. Ofrece escalabilidad en las industrias, disponibilidad de datos, seguridad y rendimiento. Además, proporciona funciones de administración para que puedan optimizar, organizar y configurar el acceso a sus datos para cumplir con su negocio específico, organizativo y, requisitos de cumplimiento (Amazon Web Services Inc., 2023e).

CloudWatch. Según AWS (Amazon Web Services Inc., 2023a) define como “un servicio de monitorización y administración que proporciona datos e información procesable para AWS, aplicaciones locales, híbridas, otras aplicaciones en la nube y recursos de infraestructura. Puede recopilar y acceder a todos los datos operativos y de rendimiento en formato de registros y métricas desde una sola plataforma en lugar de monitorearlos en silos (servidor, red o base de datos). Permite monitorear toda la pila (pila, aplicaciones, infraestructura, red y servicios) y utilizar alarmas, registros y datos de evento para tomar medidas automatizadas y reducir el tiempo medio de resolución, liberando recursos importantes y permite centrarse en la creación de aplicaciones y valor empresarial”.

Metodología

Según (Trigás, 2012) define a la metodología como “aquella disciplina que indicara métodos y técnicas que hay que usar en cada fase del ciclo de vida de desarrollo del proyecto. Los elementos que componen a una metodología son: fases, documentación, técnicas y herramientas, métodos, control y evaluación”

SCRUM

(Atlassian, 2023) define como “un marco ágil de gestión de proyectos que ayuda a los equipos a estructurar y administrar su trabajo a través de un conjunto de valores, principios y prácticas. Scrum alienta a los equipos a aprender a través de experiencia, autoorganizarse mientras trabaja en un problema, y reflexionar sobre sus victorias y pérdidas para mejorar continuamente”.

Roles de scrum. Se asigna y se define la responsabilidad de cada miembro del equipo de desarrollo, para que la coordinación y la realización de las tareas no sean complicadas, según Scrum sugiere que el equipo de desarrollo sea entre 3 a 9 miembros. Según (SCRUMstudy™, 2013) define tres principales roles que son:

- **Product Owner:** Es el responsable de establecer los requisitos y la visión del producto. Es quien realiza el Product Backlog y coloca en prioridad los requisitos.
- **Scrum Master:** Es el jefe del equipo de desarrollo, por lo que tiene que encargarse de seguir el proceso de Scrum. Es quien interactuará con el Product Owner.
- **Equipo de Desarrollo:** Son encargadas de crear los entregables del producto

Pair programming (Programación en Parejas)

La programación en parejas implica principalmente que dos personas trabajen juntas en la misma máquina, para escribir todo el código del proyecto que se enviara a producción cuando este sea finalizado. Esta técnica puede ayudar a mejorar la calidad del producto sin reducir el tiempo de entrega. Pueden tomar medidas cuando sea necesario porque están atentos y analizan el código. El nivel de los desarrolladores, es un factor crucial que siempre debe tenerse en cuenta a lo largo del desarrollo cuando se utiliza esta técnica, además, debe tener un horario suficiente y flexible para ambas partes (Zamudio Domínguez et al., 2023).

Capítulo III

Implementación del Sistema

Este capítulo se enfoca en el desarrollo del sistema propuesto, un sistema de servicios web basado en APIs para la emisión y autorización de comprobantes electrónicos requeridos por el SRI (Servicio de Rentas Internas) utilizando tecnologías como NestJS, TypeORM y TypeScript, etc., los servicios web implementados en este sistema, serán consumidos por un Sistema de Punto de Venta, que es desarrollada por la otra tesina complementaria al proyecto, ambas incorporadas a la empresa “Corporación Wolf S.A.”.

Fase de análisis y planificación

En esta fase se analiza detalladamente el problema a resolver y se define específicamente los roles involucrados y requisitos del sistema por parte del cliente, para lo cual, se recolecta la información a través de reuniones con los involucrados.

Descripción del sistema

Actualmente la empresa “Corporación Wolf S.A.”, realiza el proceso de emisión de facturas electrónicas a los clientes, mediante la utilización de servicios externos de facturación electrónica. Para esto es necesario tener una licencia para utilizar los servicios, el cual, es Dátil, que es un proveedor autorizado de facturación electrónica dentro del Ecuador y Perú (Dátil, 2021).

Mediante una reunión con el gerente (representante legal) de la empresa, explicó que se requiere realizar el mismo proceso para la emisión de facturas electrónicas a los clientes, pero sin utilizar sistemas externos, sino desarrollando uno propio para la empresa, que les resultará más factible con menor costo a largo plazo.

Descripción de la metodología

Para realizar el desarrollo el sistema se utilizará la metodología ágil, debido a las circunstancias que se desarrollaran, ya que tenemos, una comunicación directa con los involucrados del proyecto, además de una flexibilidad para la entrega continua del producto final (sistema) mediante los Sprint, que se generan cierto tiempo con sus respectivos entregables. A continuación, se detalla el proceso que se utilizara para la entrega de cada Sprint:

Roles. Para el proceso del desarrollo con la metodología primero se identificó las personas involucradas en el proyecto y se designó sus respectivos roles que se encuentran en la Tabla 1, que se puede visualizar a continuación:

Tabla 1

Roles designados

N°	Rol	Integrante	Descripción
01	Scrum Master	Ing. José Vicente Ochoa Valarezo	Líder del equipo de Scrum
02	Product Owner	Ing. José Alexander Villaruel	Representa a las partes interesadas
03	Team Development	Karla Lizeth Molina Naranjo, Alex Fabricio Pachacama Cajamarca	Desarrollo y diseño de la aplicación
04	Stakeholder	Ing. Edgar Rubén López Otañez	Representa a las partes interesadas

Nota. Se puede visualizar la distribución de los integrantes del equipo, el rol y una pequeña descripción que desempeña cada uno en el proyecto.

Reuniones. Con el equipo se determinó que sería factible tres tipos de reuniones:

- **Reunión de Planificación del Sprint:** Esta reunión se llevaba a cabo 7 veces durante el período de desarrollo, donde se mostraba los avances y se realizaba una retrospectiva para las mejoras del proyecto.
- **Reunión de Revisión:** Está reunión se llevaba a cabo con el Scrum Master del Equipo, donde se estableció que sería cada lunes a las 5 p.m. durante el período de desarrollo del proyecto.
- **Reuniones Diarias:** Esta reunión se llevaba a cabo entre los desarrolladores, con el uso de la técnica del pair programming, donde se retroalimentaban y se trabajó a la par mediante la plataforma Meet y Live Share del Visual Studio Code cada uno en su máquina, durante el período de desarrollo del proyecto.

Especificación de requisitos

Para poder obtener los requisitos, se estableció una reunión con todos los miembros del equipo de desarrollo, en la cual el Product Owner definió los requerimientos del sistema, se creó el Product Backlog para determinar las historias de usuario, dando así, una estimación al esfuerzo total y priorizando los requerimientos planteados, debido a que unos tienen mayor relevancia que otros. Una vez analizado estos requerimientos se obtuvieron 23 historias de usuario, para el desarrollo del sistema que se especifican en la Tabla 2, a continuación:

Tabla 2

Product Backlog

ID	Historias de usuario/actividades	Puntos Estimados	Prioridad
AC-01	Entrevista con el cliente para la definición de requisitos	3	Alta

ID	Historias de usuario/actividades	Puntos Estimados	Prioridad
AC-02	Diseñar el diagrama: entidad-relación, paquetes, despliegue y la estructura lógica para la definición de la arquitectura del sistema	9	Media
AC-03	Diseñar e implementar la base de datos	18	Alta
AC-04	Definir el estándar de codificación	3	Baja
AC-05	Instalar entorno de desarrollo (VS Code, NestJS, TypeORM, NodeJS, PgAdmin)	3	Media
AC-06	Revisar la ficha técnica de los documentos requeridos por el SRI de febrero 2023. Versión 2.23	9	Alta
AC-07	Investigar librerías para crear pdf y xml para el envío de correos electrónicos	12	Alta
AC-08	Investigar librerías para firmar electrónicamente documentos con el archivo .p12 usando el estándar XAdES	9	Alta
AC-09	Investigar para crear los pipelines para la documentación y el despliegue del sistema	18	Media
HU-01	Gestionar los productos	4	Media
HU-02	Gestionar las categorías	4	Media
HU-03	Gestionar los precios	4	Media

ID	Historias de usuario/actividades	Puntos Estimados	Prioridad
HU-04	Gestionar las empresas	6	Media
HU-05	Gestionar las sucursales	3	Media
HU-06	Gestionar los clientes	3	Media
HU-07	Gestionar los transportistas	3	Media
HU-08	Gestionar usuarios internos (vendedores) de la empresa	6	Media
HU-09	Gestionar usuarios (administradores) de la empresa	12	Alta
HU-10	Gestionar la autenticación de los usuarios de la empresa	18	Alta
HU-11	Crear el componente para el envío de correos electrónicos	15	Baja
HU-12	Crear el comprobante electrónico: Factura	27	Alta
HU-13	Generar reporte de Excel para facturas	9	Baja
HU-14	Crear los comprobantes electrónicos: Comprobantes de Retención, Guía de Remisión, Nota de Débito y Crédito	36	Alta
HU-15	Agregar paginación y filtrado en la búsqueda de documentos	6	Baja

ID	Historias de usuario/actividades	Puntos Estimados	Prioridad
HU-16	Guardar los reportes y pdf en un buffer	3	Media
HU-17	Generar la documentación con el estándar OpenApi utilizando redocly	18	Media
HU-18	Crear el histórico del inventario y proveedores	3	Baja
HU-19	Agregar la validación de los cálculos de la factura	6	Media
HU-20	Crear test unitarios de utilidades	12	Baja
HU-21	Implementar el pipeline de integración y entrega continua CI/CD del sistema	12	Alta
HU-22	Configurar el servidor para el despliegue en AWS	12	Alta
HU-23	Realizar pruebas de aceptación de los múltiples EndPoint del sistema	9	Media

Nota. En esta tabla se detallan las historias de usuario y actividades que se utilizarán para desarrollar los Sprints con su prioridad debida y sus puntos de historia estimados.

Fase de diseño

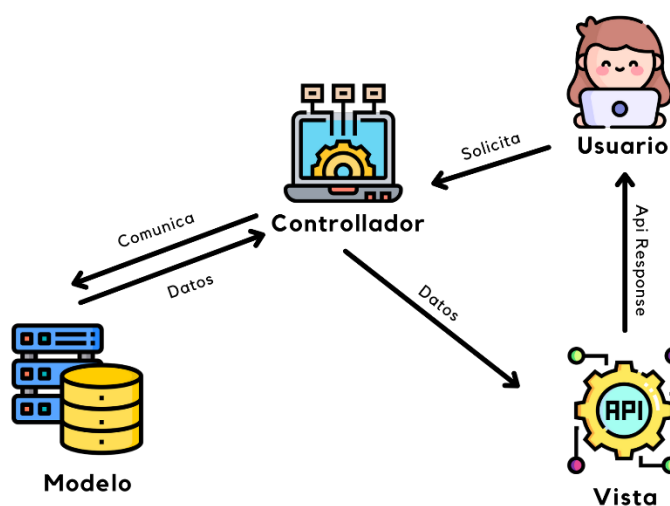
En esta fase se realizan los componentes de arquitectura para el diseño y modelado de la base de datos, definición de estándares, despliegue del sistema, que nos proporcionará el marco de referencia necesario para poder desarrollar el sistema.

Arquitectura lógica

El patrón de arquitectura que se utiliza para el desarrollo del sistema es MVC (Model View Controller / Modelo Vista Controlador), debido al framework de NestJs que usa este patrón, que nos ayuda a separar la lógica del negocio de los modelos permitiendo su implementación y su modificación sea más flexible (Zulian, 2011). A continuación, en la Figura 2 se muestra la estructura del patrón:

Figura 2

Arquitectura Lógica del Backend



Nota. En la figura se muestra la arquitectura MVC utilizada en el proyecto. Imagen autoría propia.

Descripción de la estructura

La estructura de la arquitectura de MVC se divide en tres partes que se describen a continuación:

Modelo. (Datos y reglas del Negocio), representa y gestiona los datos del programa, el modelo funciona independientemente a la vista y el controlador (Díaz & Fernández, 2012). En el proyecto el modelo serían las entidades para la base de datos de todo el proyecto.

Vista. Crea y gestiona una visualización al usuario interactuando con los controladores (Díaz & Fernández, 2012). En el sistema se definen a las vistas como las respuestas que se obtienen de los controladores a través de los formatos json.

Controlador. (Gestiona las entradas del usuario), es un objeto que comprende los comandos del usuario y actúa sobre los modelos y la vista (Díaz & Fernández, 2012). En el sistema se define a los controladores como los múltiples endpoints que creamos a través de los servicios.

Diseño detallado

Modelo entidad – relación. Se conoce como una técnica para generar un modelo de datos de alta calidad, que proporciona a los sistemas de información ofreciendo una forma de estándar de definir los datos y las relaciones entre estos, mejorando en gran medida la calidad del sistema y aumentando la productividad del software (Barker, 1994). Se representa el modelo entidad relación de la base de datos que se utilizó en el desarrollo del proyecto en el Anexo 6.

Diagrama de paquetes del sistema. Crea un modelo de la vista estática del sistema y muestran como los componentes están organizados y son interdependientes (Scolari, 2010). Se puede visualizar el diagrama de paquetes que se utilizó para el backend en el Anexo 7.

Los paquetes que se utilizan se describen a continuación:

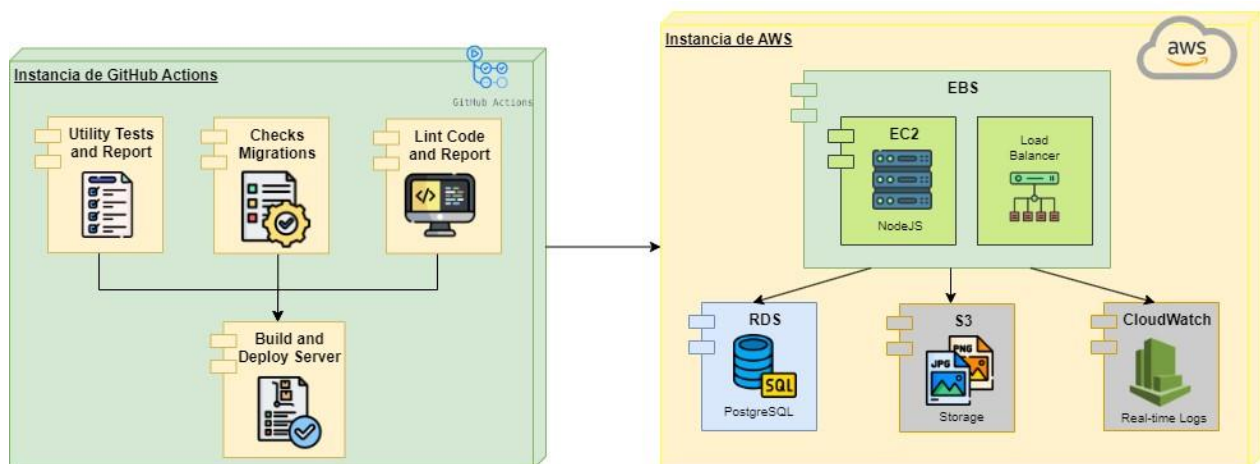
- **Paquete Auth:** Es el encargado de realizar las acciones de inicio de sesión, registro de los negocios, recuperar cuenta y cambiar contraseña, etc.

- **Paquete Business:** Es el encargado de gestionar nuevos negocios, sucursales y reportes de las facturas.
- **Paquete Business Users:** Es el encargado de realizar la gestión de los clientes y transportistas.
- **Paquete Documents:** Es el responsable de la lógica para emitir, autorizar y mostrar información de los comprobantes electrónicos como la factura, comprobante de retención, guía de remisión, nota de crédito y débito.
- **Paquete Email:** Es el responsable de generar los emails con cada documento emitido y autorizado al cliente.
- **Paquete Products:** Es el responsable de gestionar inventarios, categorías y productos de los negocios.

Diagrama de despliegue. Es un tipo de diagrama UML, que se detalla la arquitectura en la que el sistema se va a ejecutar tanto de hardware o software, para poder comprender de mejor manera (Siriwardhana, 2020). En la Figura 3 se muestra el diagrama de despliegue detalladamente.

Figura 3

Diagrama de Despliegue



Nota. En la figura se muestra el diagrama de despliegue especificando la utilización de las GitHub Actions y el AWS.

Estándar de codificación. Son un conjunto de buenas prácticas de desarrollo que ayudan a mantener el mismo estilo de codificación, facilitando la lectura y su comprensión, reduciendo la cantidad de decisiones que toman los desarrolladores durante la programación (Ochoa & Bone, 2022). A continuación, en la Tabla 3 se detalla el estándar de codificación que se utilizó para el desarrollo del proyecto:

Tabla 3

Estándar de codificación Base de Datos

Tipo	Estándar	Ejemplo
Primary Key	snake_case	debit_note_id
Tablas	snake_case	debit_notes
Atributos	snake_case	document_code

Nota. La tabla muestra la utilización de estándar de codificación para la base de datos como las tablas, atributos y llaves primarias.

Tabla 4*Estándar de codificación Backend*

Tipo	Estándar	Ejemplo
Clase	PascalCase	Invoice
Método	camelCase	authorizeDocuments
Variable	camelCase	invoiceService

Nota. La tabla muestra la utilización de estándar de codificación para el Backend como sus clases, métodos y variables.

Modelo físico de la base de datos. Para una mejor comprensión, manejo e integridad de los datos que se utilizaran en el desarrollo del proyecto, se ha diseñado un modelo físico de la base de datos, que contiene tablas, columnas, claves primarias, secundarias y sus respectivas relaciones entre sí. El modelo físico se encuentra en el Anexo 8.

Fase de desarrollo

En esta fase se presenta el desarrollo del sistema mediante las actividades designadas para los Sprints mediante las historias de usuarios.

Cada Sprint comienza con una reunión de planificación del Sprint donde el Product Owner y el equipo de desarrollo planifican el desarrollo del siguiente Sprint. En la reunión el Scrum Master se obtiene o presenta al equipo el Product Backlog donde indica la estimación y la prioridad que tiene cada requerimiento que se mostró en la Tabla 1. Para poder llevar a cabo las reuniones, se realizaron a través de la plataforma Google Meet.

Con las reuniones se obtuvieron 23 historias de usuario, que se distribuyeron en 7 Sprints, cada Sprint tendrá una duración de 3 semanas, sin contar los fines de semana (sábado y domingo), los cuales tendrán máximo 45 puntos de historia estimados, que equivaldría a 45 horas de trabajo, donde cada día se trabajó 3 horas y a la semana un total de 15 horas. Dando como resultado el Sprint Backlog Global que se detalla a continuación:

Tabla 5

Sprint Backlog Global

ID	Historias de usuario/actividades	Puntos	Fecha	Puntos
		Estimados	Inicio-Fin	Totales
Sprint 01				
AC-01	Entrevista con el cliente para la definición de requisitos	3		
AC-02	Diseñar el diagrama: entidad-relación, paquetes, despliegue y la estructura lógica para la definición de la arquitectura del sistema	9	27/02/2023	
AC-03	Diseñar e implementar la base de datos	18	- 17/03/2023	45
AC-04	Definir el estándar de codificación	3		
AC-05	Instalar entorno de desarrollo (VS Code, NestJS, TypeORM, NodeJS, PgAdmin)	3		

ID	Historias de usuario/actividades	Puntos Estimados	Fecha Inicio-Fin	Puntos Totales
AC-06	Revisar la ficha técnica SRI febrero 2023. Versión 2.23	9		
Sprint 02				
HU-01	Gestionar los productos	4		
HU-02	Gestionar las categorías	4		
HU-03	Gestionar los precios	4		
HU-04	Gestionar las empresas	6	20/03/2023	
HU-05	Gestionar las sucursales	3	-	45
			07/04/2023	
HU-06	Gestionar los clientes	3		
HU-07	Gestionar los transportistas	3		
HU-08	Gestionar usuarios internos (vendedores) de la empresa	6		
HU-09	Gestionar usuarios (administradores) de la empresa	12		

ID	Historias de usuario/actividades	Puntos Estimados	Fecha Inicio-Fin	Puntos Totales
Sprint 03				
HU-10	Gestionar la autenticación de los usuarios de la empresa	18	10/04/2023	
AC-07	Investigar librerías para crear pdf y xml para el envío de correos electrónicos	12	- 28/04/2023	45
HU-11	Crear el componente para el envío de correos electrónicos	15		
Sprint 04				
AC-08	Investigar librerías para firmar electrónicamente documentos con el archivo .p12 usando el estándar XAdES	9	01/05/2023 -	45
HU-12	Crear el comprobante electrónico: Factura	27	19/05/2023	
HU-13	Generar reporte de Excel para facturas	9		
Sprint 05				
HU-14	Crear los comprobantes electrónicos: Comprobantes de Retención, Guía de Remisión, Nota de Débito y Crédito	36	22/05/2023 - 09/06/2023	45

ID	Historias de usuario/actividades	Puntos Estimados	Fecha Inicio-Fin	Puntos Totales
HU-15	Agregar paginación y filtrado en la búsqueda de documentos	6		
HU-16	Guardar los reportes y pdf en un buffer	3		
Sprint 06				
HU-17	Generar la documentación con el estándar OpenApi utilizando redocly	18		
HU-18	Crear el histórico del inventario y proveedores	3	12/06/2023	
HU-19	Agregar la validación de los cálculos de la factura	6	- 30/06/2023	45
AC-09	Investigar para crear los pipelines para la documentación y el despliegue del sistema	18		
Sprint 07				
HU-20	Crear test unitarios de utilidades	12		
HU-21	Implementar el pipeline de integración y entrega continua CI/CD del sistema	15	03/07/2023 - 21/07/2023	45
HU-22	Configurar el servidor para el despliegue en AWS	9		

ID	Historias de usuario/actividades	Puntos Estimados	Fecha Inicio-Fin	Puntos Totales
HU-23	Realizar pruebas de aceptación de los múltiples endpoints del sistema	9		

Nota. En la tabla se observa los 7 Sprints que se obtuvo con sus debidas historias técnicas y de usuario detalladamente.

Sprint 01: Análisis y diseño del sistema

Para el desarrollo del Sprint 01, las actividades 01-02-03-04-05-06 que se detallaron en la Tabla1 fueron referencias, para el comienzo del desarrollo del sistema como la entrevista con el Product Owner para la definición de los requerimientos, la cual se llevó por zoom con todos los involucrados, una vez obtenido los requerimientos se empezó a especificar los estándares de codificación e instalar las herramientas definidas para el diseño e implementación de la base de datos, además se realiza una revisión de la Guía para Contribuyentes – Ficha Técnica: Manual de Usuario, catálogo y especificaciones técnicas “Emisión de Comprobantes Electrónicos”.

Resultados del Sprint

El Sprint 01 se llevó a cabo con normalidad cumpliendo las actividades definidas en el tiempo establecido. A continuación, se detallan los resultados finalizados:

Figura 4

Entrevista con el cliente para definir los requerimientos

The image shows a Zoom meeting window. On the left, a web application interface is visible, displaying a table of 'Comprobantes Emitidos' (Issued Receipts) for the period 'Ene 1, 2023 - ene 18, 2023'. The table has columns for 'CLIENTE', 'FECHA', 'NÚMERO', 'TOTAL', and 'ESTADO'. The data is as follows:

CLIENTE	FECHA	NÚMERO	TOTAL	ESTADO
José Villarruel	Ene 7, 2023	002-002-000001125	580.00	Enviado
Francisco Salazar	Ene 6, 2023	002-002-000000114	\$5.000.00	Enviado
Edison Rodrigo Ramos Zarza	Ene 6, 2023	002-002-000000113	\$24.40	Enviado

On the right side of the Zoom window, a list of participants is shown:

- José Villarruel
- Karla Molina
- José Ochoa
- Alex Pachacama
- Christian Guevara

Nota. En la figura se observa la entrevista realizada mediante la plataforma zoom con el representante legal de la empresa (Product Owner) y el equipo de desarrollo para establecer los requerimientos que tendrá el sistema a desarrollar.

Figura 5

Implementación de la base de datos

```

1  @Entity({ name: 'roles' })
2  export class Role {
3    @PrimaryGeneratedColumn('increment', { name: 'role_id', type: 'int' })
4    roleId: number;
5
6    @Column({ name: 'name', type: 'varchar', length: 32 })
7    name: string;
8
9    @Column({ name: 'description', type: 'varchar', length: 64 })
10   description: string;
11
12   // @Exclude()
13   @CreateDateColumn({
14     name: 'created_at',
15     type: 'timestampz',
16     default: () => 'CURRENT_TIMESTAMP',
17   })
18   createdAt: Date;
19
20   // @Exclude()
21   @UpdateDateColumn({
22     name: 'updated_at',
23     type: 'timestampz',
24     default: () => 'CURRENT_TIMESTAMP',
25   })
26   updatedAt: Date;
27
28   /* relations */
29   @OneToMany(() => UserToRole, (userToRole) => userToRole.role)
30   userToRoles: UserToRole[];
31 }

```

Nota. El Sprint 01 se desarrolló sin ningún inconveniente las actividades del Sprint respectivo.

Sprint 02: Gestión del sistema

Para el desarrollo del Sprint 02, las historias de usuarios 01-02-03-04-05-06-07-08-09 que se detallan a continuación fueron referencias para implementar la gestión del sistema.

Historias de Usuario detallada

Historias de Usuario

Número: HU-01

Usuario: Administrador

Nombre historia: Gestionar los productos

Prioridad: Media

Riesgo en desarrollo: Baja

Puntos estimados (horas): 4

Interacción asignada: 1

Programadores responsables: Karla Molina, Alex Pachacama

Descripción: Como administrador necesito crear, leer, actualizar y eliminar la información de los productos para tener un inventario actualizado

Validación (Criterios de aceptación):

- El sistema permitirá leer productos a través de un endpoint específico
- El sistema permitirá crear productos a través de un endpoint específico
- El sistema permitirá actualizar productos a través de un endpoint específico
- El sistema permitirá eliminar productos a través de un endpoint específico

Historias de Usuario

Número: HU-02

Usuario: Administrador

Nombre historia: Gestionar las categorías

Prioridad: Media

Riesgo en desarrollo: Baja

Puntos estimados (horas): 4

Interacción asignada: 1

Programadores responsables: Karla Molina, Alex Pachacama

Descripción: Como administrador necesito crear, leer, actualizar y eliminar la información de las categorías para tener la información actualizada y ordenada.

Validación (Criterios de aceptación):

Historias de Usuario

- El sistema permitirá leer categorías a través de un endpoint específico
 - El sistema permitirá crear categorías a través de un endpoint específico
 - El sistema permitirá actualizar categorías a través de un endpoint específico
 - El sistema permitirá eliminar categorías a través de un endpoint específico
-

Historias de Usuario

Número: HU-03

Usuario: Administrador

Nombre historia: Gestionar los precios

Prioridad: Media

Riesgo en desarrollo: Baja

Puntos estimados (horas): 4

Interacción asignada: 1

Programadores responsables: Karla Molina, Alex Pachacama

Descripción: Como administrador necesito crear, leer y eliminar la información de los precios para mantener actualizado cada producto

Validación (Criterios de aceptación):

- El sistema permitirá leer precios a través de un endpoint específico
 - El sistema permitirá crear precios a través de un endpoint específico
 - El sistema permitirá eliminar precios a través de un endpoint específico
-

Historias de Usuario

Número: HU-04

Usuario: Administrador

Nombre historia: Gestionar las empresas

Prioridad: Media

Riesgo en desarrollo: Baja

Puntos estimados (horas): 6

Interacción asignada: 1

Historias de Usuario

Programadores responsables: Karla Molina, Alex Pachacama

Descripción: Como administrador necesito crear, leer, actualizar y eliminar la información de la empresa para mantener la información actualizada

Validación (Criterios de aceptación):

- El sistema permitirá leer información de la empresa a través de un endpoint específico
 - El sistema permitirá crear empresas a través de un endpoint específico
 - El sistema permitirá actualizar información de la empresa a través de un endpoint específico
 - El sistema permitirá eliminar empresas a través de un endpoint específico
-

Historias de Usuario

Número: HU-05

Usuario: Administrador

Nombre historia: Gestionar las sucursales

Prioridad: Media

Riesgo en desarrollo: Baja

Puntos estimados (horas): 3

Interacción asignada: 1

Programadores responsables: Karla Molina, Alex Pachacama

Descripción: Como administrador necesito crear, leer, actualizar y eliminar la información de las sucursales de mi empresa, para mantener la información real y actualizada.

Validación (Criterios de aceptación):

- El sistema permitirá leer información de la sucursal a través de un endpoint específico
 - El sistema permitirá crear sucursales a través de un endpoint específico
-

Historias de Usuario

- El sistema permitirá actualizar información de la sucursal a través de un endpoint específico
 - El sistema permitirá eliminar sucursales a través de un endpoint específico
-

Historias de Usuario

Número: HU-06

Usuario: Administrador

Nombre historia: Gestionar los clientes

Prioridad: Media

Riesgo en desarrollo: Baja

Puntos estimados (horas): 3

Interacción asignada: 1

Programadores responsables: Karla Molina, Alex Pachacama

Descripción: Como administrador necesito crear, leer, actualizar y eliminar la información de los clientes de la empresa, para utilizar la información al momento de facturar.

Validación (Criterios de aceptación):

- El sistema permitirá leer información del cliente a través de un endpoint específico
 - El sistema permitirá crear clientes a través de un endpoint específico
 - El sistema permitirá actualizar información del cliente a través de un endpoint específico
 - El sistema permitirá eliminar clientes a través de un endpoint específico
-

Historias de Usuario

Número: HU-07

Usuario: Administrador

Nombre historia: Gestionar los transportistas

Prioridad: Media

Riesgo en desarrollo: Baja

Puntos estimados (horas): 3

Interacción asignada: 1

Programadores responsables: Karla Molina, Alex Pachacama

Descripción: Como administrador necesito crear, leer, actualizar y eliminar la información de los transportistas de la empresa, para utilizar la información al momento de crear una guía de remisión.

Validación (Criterios de aceptación):

- El sistema permitirá leer información del transportista a través de un endpoint específico
 - El sistema permitirá crear transportistas a través de un endpoint específico
 - El sistema permitirá actualizar información del transportista a través de un endpoint específico
 - El sistema permitirá eliminar transportistas a través de un endpoint específico
-

Historias de Usuario

Número: HU-08

Usuario: Administrador

Nombre historia: Gestionar usuarios internos (vendedores) de la empresa

Prioridad: Media

Riesgo en desarrollo: Baja

Puntos estimados (horas): 6

Interacción asignada: 1

Programadores responsables: Karla Molina, Alex Pachacama

Historias de Usuario

Descripción: Como administrador necesito crear, leer, actualizar y eliminar la información de los usuarios (vendedores) de la empresa, para tener varios usuarios (vendedores) en las diversas sucursales de la empresa.

Validación (Criterios de aceptación):

- El sistema permitirá leer usuarios a través de un endpoint específico
 - El sistema permitirá crear usuarios a través de un endpoint específico con el rol de vendedores.
 - El sistema permitirá actualizar usuarios a través de un endpoint específico.
 - El sistema permitirá eliminar usuarios a través de un endpoint específico.
 - El sistema permitirá la limitación de acceso en los endpoints dependiendo de su rol asignado.
-

Historias de Usuario

Número: HU-09

Usuario: Administrador

Nombre historia: Gestionar los usuarios (administradores) de la empresa

Prioridad: Alta

Riesgo en desarrollo: Baja

Puntos estimados (horas): 12

Interacción asignada: 1

Programadores responsables: Karla Molina, Alex Pachacama

Descripción: Como administrador necesito crear, leer y actualizar la información de los usuarios (administradores) de la empresa, para tener varias empresas y generar más ingresos

Validación (Criterios de aceptación):

- El sistema permitirá leer usuarios a través de un endpoint específico
-

Historias de Usuario

- El sistema permitirá crear usuarios a través de un endpoint específico con el rol de administrador de empresa.
 - El sistema permitirá actualizar usuarios a través de un endpoint específico.
-

Tabla 6*Sprint Backlog 02*

Sprint 02						
Fecha Inicio:			Fecha Fin:		Jornada:	
20/03/2023			07/04/2023		3 horas diarias	
H.U	Tareas	Horas	Fecha Inicio	Fecha Fin	Encargados	Estado
01	Gestionar los productos	4	20/03/2023	21/03/2023	Karla Molina, Alex Pachacama	Finalizado
02	Gestionar las categorías	4	21/03/2023	22/03/2023	Karla Molina, Alex Pachacama	Finalizado
03	Gestionar los precios	4	22/03/2023	23/04/2023	Karla Molina, Alex Pachacama	Finalizado
04	Gestionar las empresas	6	24/03/2023	27/03/2023	Karla Molina, Alex Pachacama	Finalizado

Sprint 02							
05	Gestionar las sucursales	3	28/03/2023	28/03/2023	Molina, Alex	Finalizado	Karla Pachacama
06	Gestionar los clientes	3	29/03/2023	29/03/2023	Molina, Alex	Finalizado	Karla Pachacama
07	Gestionar los transportistas	3	30/03/2023	30/03/2023	Molina, Alex	Finalizado	Karla Pachacama
08	Gestionar usuarios internos (vendedores) de la empresa	6	31/03/2023	03/04/2023	Molina, Alex	Finalizado	Karla Pachacama
09	Gestionar usuarios (administradores) de la empresa	12	04/04/2023	07/04/2023	Molina, Alex	Finalizado	Karla Pachacama

Nota. En la tabla se muestran detalles de las tareas que se realizarán durante el Sprint, quien es el responsable, horas, fecha de inicio y final de cada historia de usuario y del sprint, además del estado si ha sido finalizada, completando las 45 horas durante las tres semanas sin contar fines de semana.

Resultados del Sprint

A continuación, se detallan un ejemplo de los resultados de las historias de usuario finalizadas:

Figura 6

Creación del servicio para crear la categoría

```

1  async create(data: CreateCategoryDto, file: Express.Multer.File) {
2    const newCategory = this.categoryRepo.create(data);
3    try {
4      const { url } = await this.publicFilesService.savePublicFile(
5        file,
6        `businesses/${data.businessId}/categories/`,
7      );
8      this.categoryRepo.merge(newCategory, { image: url });
9      await this.categoryRepo.save(newCategory);
10   } catch (error) {
11     console.log('error', error);
12     return ResponseClient.failed(
13       'Error al guardar las credenciales de facturación',
14       HttpStatus.BAD_REQUEST,
15     );
16   }
17   return ResponseClient.success(newCategory);
18 }

```

Nota. Se crea el servicio de crear categoría de un producto con sus campos establecidos, para guardar en la base de datos.

Figura 7

Creación del controlador para crear la categoría

```

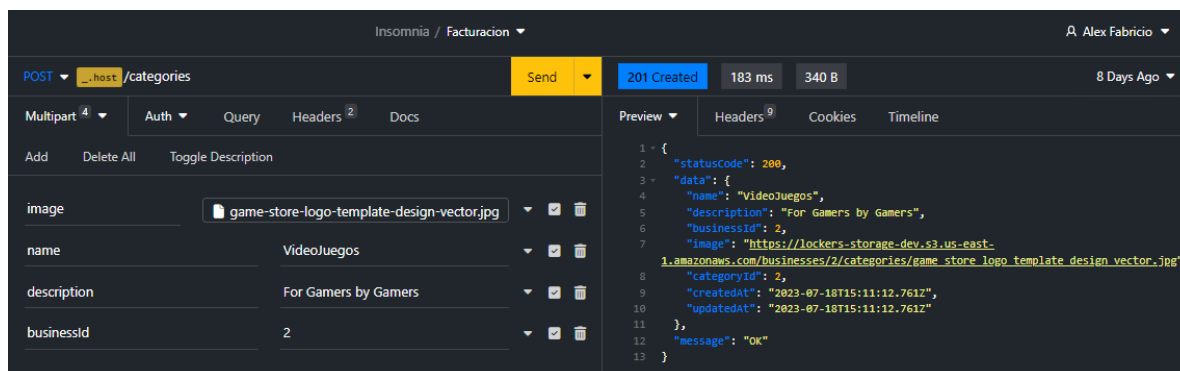
1  @Roles(RoleEnum.SUPER_ADMIN, RoleEnum.BUSINESS_ADMIN)
2  @ApiFile()
3  @Post()
4  async create(
5    @Req() req: Request,
6    @Body() payload: CreateCategoryDto,
7    @UploadedFile() file: Express.Multer.File,
8  ) {
9    const user = req.user as PayloadToken;
10   await this.businessService.validateUserScope(
11     user.sub,
12     payload.businessId,
13     [RoleEnum.BUSINESS_ADMIN],
14     user.roles.includes(RoleEnum.SUPER_ADMIN),
15   );
16   return this.categoriesService.create(payload, file);
17 }

```

Nota. Se crea el controlador para generar la Api de tipo POST para consumir y crear categoría de un producto.

Figura 8

Consumo del endpoint de crear la categoría



Nota. En la figura se muestra como consumir la Api a través de la herramienta Insomnia para comprobar la funcionalidad al momento de crear la categoría de un producto.

Nota. El Sprint 02 Se desarrollo sin ningún inconveniente cumpliendo las historias de usuario designadas en el Sprint respectivo.

Sprint 03: Implementación de Autenticación y envío de correos

Para el desarrollo del Sprint 03, se realizó la actividad de investigar librerías al crear pdf y xml, para él envió de correos electrónicos, para los comprobantes electrónicos, junto con las historias de usuario 10-11 que se realizaron durante el Sprint, se detallan a continuación:

Historias de Usuario detalladas

Historias de Usuario

Número: HU-10

Usuario: Cliente

Nombre historia: Generar la autenticación de los usuarios de la empresa

Prioridad: Alta

Riesgo en desarrollo: Baja

Puntos estimados (horas): 18

Interacción asignada: 1

Programadores responsables: Karla Molina, Alex Pachacama

Historias de Usuario

Descripción: Como cliente necesito la autorización y autenticación de mi cuenta para usar el sistema y sus múltiples endpoints.

Validación (Criterios de aceptación):

- El sistema permitirá restringir y controlar el acceso a los endpoints mediante una apikey alfanumérico.
 - El sistema permitirá el inicio de sesión con Passport y JWT a través de un guardian.
 - El sistema permitirá actualizar la contraseña de la cuenta a través de un endpoint específico.
-

Historias de Usuario

Número: HU-11

Usuario: Cliente

Nombre historia: Crear el componente para el envío de correos electrónicos

Prioridad: Baja

Riesgo en desarrollo: Baja

Puntos estimados (horas): 15

Interacción asignada: 1

Programadores responsables: Karla Molina, Alex Pachacama

Descripción: Como cliente quiero que me lleguen notificaciones al email para recuperar la cuenta, en caso de pérdida de datos.

Validación (Criterios de aceptación):

- El sistema permitirá el envío de correos electrónicos
 - El sistema permitirá recuperar la cuenta mediante un correo electrónico a través de un endpoint específico.
-

Historias de Usuario

- El sistema permitirá cambiar la contraseña de la cuenta mediante un correo electrónico personalizado a través de un endpoint específico.
-

Tabla 7

Sprint Backlog 03

Sprint 03						
Fecha Inicio:		Fecha Fin:		Jornada:		
10/04/2023		28/04/2023		3 horas diarias		
H.U	Tareas	Horas	Fecha Inicio	Fecha Fin	Encargados	Estado
AC	Gestionar la				Karla	
10	autenticación de los usuarios de la empresa	18	10/04/2023	17/04/2023	Molina, Alex Pachacama	Finalizado
07	Investigar librerías para crear pdf y xml para el envío de correos electrónicos	12	18/04/2023	21/04/2023	Molina, Alex Pachacama	Finalizado
11	Crear el componente para el envío de correos electrónicos	15	24/04/2023	28/04/2023	Molina, Alex Pachacama	Finalizado

Nota. En la tabla se muestran detalles de las tareas que se realizarán durante el Sprint, quien es el responsable, horas, fecha de inicio y final de cada historia de usuario y del sprint, además

del estado si ha sido finalizada, completando las 45 horas durante las tres semanas sin contar fines de semana.

Resultados

En el Sprint 03 se representa completamente la implementación de la autenticación y autorización para poder acceder al sistema y gestionar sus cuentas, además, se implementó el uso de la apikey para restringir la integridad y seguridad de los datos de cada endpoint endpoints. A continuación, se detallan un ejemplo del resultado de una historia de usuario finalizada:

Figura 9

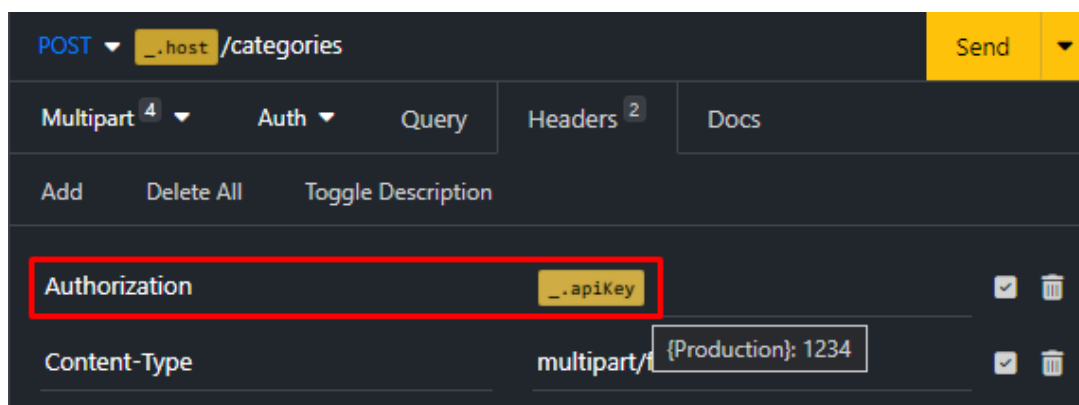
Creación del guardian para la ApiKey

```
1  canActivate(  
2    context: ExecutionContext,  
3  ): boolean | Promise<boolean> | Observable<boolean> {  
4    const isPublic = this.reflector.get(IS_PUBLIC_KEY, context.getHandler());  
5    if (isPublic) return true;  
6  
7    const request = context.switchToHttp().getRequest<Request>();  
8    const authHeader = request.header('Authorization');  
9    const isAuth = authHeader === this.configService.apiKey;  
10   if (!isAuth)  
11     return ResponseClient.failed('No Autorizado', HttpStatus.FORBIDDEN);  
12  
13   return isAuth;
```

Nota. Se crea el guardian para restringir el acceso y tener seguridad al momento de consumir las APIs.

Figura 10

Utilización del ApiKey



Nota. Se muestra un ejemplo de cómo debe ir el header de la petición con la apikey al momento de consumir la Api.

Nota. En el Sprint 03 se desarrolló sin ningún inconveniente las actividades del Sprint respectivo.

Sprint 04: Implementación de factura electrónica

Para el desarrollo del Sprint 04, se realizó la actividad de investigar librerías, para firmar electrónicamente documentos con el archivo .p12 usando el estándar XAdES y las historias de usuarios 12-13, que se detallan a continuación muestran las actividades al momento de generar una factura de una empresa al cliente.

Historias de Usuario detalladas

Historias de Usuario

Número: HU-12

Usuario: Cliente

Nombre historia: Crear el comprobante electrónico: Factura

Prioridad: Alta

Riesgo en desarrollo: Baja

Historias de Usuario

Puntos estimados (horas): 27

Interacción asignada: 1

Programadores responsables: Karla Molina, Alex Pachacama

Descripción: Como cliente quiero emitir una factura electrónica, para que el SRI autorice y tenga una validez legal siguiendo la estructura definida.

Validación (Criterios de aceptación):

- El sistema permitirá firmar electrónicamente los documentos utilizando el estándar XAdES
 - El sistema permitirá la emisión y autorización de facturas usando la estructura del xml en la versión 1.0.0 a través de un endpoint específico.
 - El sistema permitirá enviar un correo electrónico al cliente notificando la factura generada con el xml y pdf autorizado.
-

Historias de Usuario

Número: HU-13

Usuario: Cliente

Nombre historia: Generar reporte de Excel para facturas

Prioridad: Baja

Riesgo en desarrollo: Baja

Puntos estimados (horas): 9

Interacción asignada: 1

Programadores responsables: Karla Molina, Alex Pachacama

Descripción: Como cliente quiero generar un reporte de facturas en formato xlsx en un rango de fecha determinado, para el control y recopilación de los datos

Validación (Criterios de aceptación):

- El sistema permitirá generar un reporte de facturas en formato xlsx a través de un endpoint específico.
-

Tabla 8*Sprint Backlog 04*

Sprint 04						
Fecha Inicio:		Fecha Fin:		Jornada:		
04/05/2023		19/05/2023		3 horas diarias		
H.U	Tareas	Horas	Fecha Inicio	Fecha Fin	Encargados	Estado
AC						
	Investigar librerías para firmar electrónicamente				Karla	
08	documentos con el archivo .p12 usando el estándar XAdES	9	01/05/2023	19/05/2023	Molina, Alex Pachacama	Finalizado
12	Crear el comprobante electrónico: Factura	27	24/05/2023	16/05/2023	Molina, Alex Pachacama	Finalizado
13	Generar reporte de Excel para facturas	9	17/05/2023	19/05/2023	Molina, Alex Pachacama	Finalizado

Nota. En la tabla se muestran detalles de las tareas que se realizaron durante el Sprint, quien es el responsable, horas, fecha de inicio y final de cada historia de usuario y del sprint, además del estado si ha sido finalizada, completando las 45 horas durante las tres semanas sin contar fines de semana.

Resultado.

En el Sprint 04 se logró implementar el primer documento electrónico autorizado por el SRI y notificarlo al cliente respectivo mediante el correo electrónico que es la factura. A continuación, se detallan los resultados las historias de usuario realizadas y finalizadas:

Figura 11

Creación del controlador para crear la factura

```
1 @Roles(RoleEnum.BUSINESS_ADMIN, RoleEnum.SELLER)
2 @Post()
3 async create(@Req() req: Request, @Body() payload: CreateInvoiceDto) {
4     // recovery business values
5     const business = await this.businessService.recoverySignatureByBranch(
6         payload.document.branchId,
7     );
8     // check scopes
9     const user = req.user as PayloadToken;
10    await this.businessService.validateUserScope(
11        user.sub,
12        business.businessId,
13        [RoleEnum.BUSINESS_ADMIN],
14        user.roles.includes(RoleEnum.SUPER_ADMIN),
15    );
16    return this.invoicesServices.create(payload, business.sign);
17 }
```

Nota. Se crea el controlador para generar la Api de tipo POST para consumir y crear la factura.

Figura 12

Consumo del endpoint para crear factura

```

POST http://localhost:3000/invoices
Send
201 Created 99.8 ms 92 B 19 Hours Ago

JSON Auth Query Headers 2 Docs
1 - {
2 -   "document": { ++ 8 ++ },
21 -   "invoice": {
22     "currency": "dolar",
23     "tip": 0,
24     "totalDiscount": 0,
25     "amount": 22.40,
26     "totalWithoutDiscount": 20
27   },
28   "taxesInvoices": [ ++ 1 ++ ],
37   "payment": [ ++ 1 ++ ],
43   "details": [
44     {
45       "description": "Flash",
46       "productId": 1,
47       "code": "G-0001",
48       "quantity": 2,
49       "totalPrice": 20,
50       "discount": 0.00,
51       "unitPrice": 10,
52       "unitMeasure": "unidad",
53       "taxes": [
54         {
55           "code": 2,
56           "taxBases": 20,
57           "rate": 12,
58           "value": 2.40,
59           "percentcode": 2
60         }
61       ]
62     }
63   ]
64 }

Preview Headers 9 Cookies Timeline
1 - {
2   "statusCode": 200,
3   "data": "1707302301172765594400110010010000000017567651519",
4   "message": "Ok"
5 }

```

Nota. En la figura se muestra como consumir la Api a través de la herramienta Insomnia para crear, emitir y autorizar una factura

Nota. En el Sprint 04, fue complejo tratar e implementar la firma electrónica en formato (.p12), pero se logró implementarlo a través de la librería node-forge.

Sprint 05: Implementación de Comprobantes electrónicos

Para el desarrollo del Sprint 05, las historias de usuarios 14-15-16 que se detallan a continuación, ayudan a completar la creación, emisión y autorización de documentos generando los comprobantes electrónicos faltantes, con su paginado y filtración respectiva.

Historias de Usuario detalladas

Historias de Usuario

Número: HU-14

Usuario: Cliente

Nombre historia: Crear los comprobantes electrónicos: Comprobantes de Retención, Guía de Remisión, Nota de Débito y Crédito.

Prioridad: Alta

Riesgo en desarrollo: Baja

Puntos estimados (horas): 36

Interacción asignada: 1

Programadores responsables: Karla Molina, Alex Pachacama

Descripción: Como cliente quiero emitir los comprobantes electrónico restantes, para que el SRI autorice y tenga una validez legal siguiendo la estructura definida.

Validación (Criterios de aceptación):

- El sistema permitirá la emisión y autorización de comprobantes de retención, usando la estructura del xml en la versión 1.0.0 a través de un endpoint específico.
 - El sistema permitirá la emisión y autorización de la guía de remisión, usando la estructura del xml en la versión 1.0.0 a través de un endpoint específico.
 - El sistema permitirá la emisión y autorización de una nota de crédito, usando la estructura del xml en la versión 1.0.0 a través de un endpoint específico.
 - El sistema permitirá la emisión y autorización de una nota de débito, usando la estructura del xml en la versión 1.0.0 a través de un endpoint específico.
 - El sistema permitirá enviar un correo electrónico al cliente, notificando que el comprobante electrónico se ha generado con el xml y pdf autorizado.
-

Historias de Usuario

Número: HU-15

Usuario: Cliente

Nombre historia: Agregar paginación y filtrado en la búsqueda de documentos

Prioridad: Baja

Riesgo en desarrollo: Baja

Puntos estimados (horas): 6

Interacción asignada: 1

Programadores responsables: Karla Molina, Alex Pachacama

Descripción: Como cliente quiero poder paginar y filtrar los documentos, para la búsqueda y el acceso a la información de manera más optimizada.

Validación (Criterios de aceptación):

- El sistema permitirá el filtrado de los documentos mediante la sucursal.
 - El sistema permitirá el filtrado de los documentos mediante el ambiente de producción o pruebas.
 - El sistema permitirá el filtrado de los documentos mediante el estado del documento (autorizado, no autorizado, rechazado, en proceso y recibido).
 - El sistema permitirá el filtrado de los documentos mediante el punto de emisión de la empresa o sucursal.
 - El sistema permitirá el filtrado de los documentos mediante el código identificador del documento.
 - El sistema permitirá el filtrado de los documentos mediante fechas de la emisión del documento (antes y después).
 - El sistema permitirá el filtrado de los documentos mediante parámetros de búsqueda (nombre e identificación del cliente y secuencial del documento).
 - El sistema permitirá el paginado de los documentos mediante los parámetros limit y offset.
-

Historias de Usuario

Número: HU-16**Usuario:** Cliente**Nombre historia:** Guardar los reportes y pdf en un buffer**Prioridad:** Media**Riesgo en desarrollo:** Baja**Puntos estimados (horas):** 3**Interacción asignada:** 1**Programadores responsables:** Karla Molina, Alex Pachacama

Descripción: Como cliente necesito almacenar los comprobantes electrónicos autorizados en formato xml, pdf y los reportes en xlsx directamente en la base de datos, para una mayor facilidad de acceso.

Validación (Criterios de aceptación):

- El sistema permitirá almacenar el xml, pdf y los reportes en tipo de dato buffer
-

Tabla 9*Sprint Backlog 05*

Sprint 05						
Fecha Inicio:			Fecha Fin:		Jornada:	
22/05/2023			09/06/2023		3 horas diarias	
H.U	Tareas	Horas	Fecha Inicio	Fecha Fin	Encargados	Estado
14	Crear los comprobantes electrónicos: Comprobantes de Retención, Guía de	36	22/05/2023	06/06/2023	Karla Molina, Alex Pachacama	Finalizado

Sprint 05

	Remisión, Nota de Débito y Crédito						
	Agregar paginación y filtrado en la búsqueda de documentos	6	07/06/2023	08/06/2023	Molina, Alex	Finalizado	Karla Pachacama
15	Guardar los reportes y pdf en un buffer	3	09/06/2023	09/06/2023	Molina, Alex	Finalizado	Karla Pachacama
16							

Nota. En la tabla se muestran detalles de las tareas que se realizaron durante el Sprint, quien es el responsable, horas, fecha de inicio y final de cada historia de usuario y del sprint, además del estado si ha sido finalizada, completando las 45 horas durante las tres semanas sin contar fines de semana.

Resultado

En el Sprint 05 se logró implementar los comprobantes electrónicos restantes con su estructura para su respectiva emisión, autorización por parte del SRI y la notificación al cliente respectivo mediante el correo electrónico. A continuación, se detallan los resultados de las historias de usuario realizadas y finalizadas

Figura 13

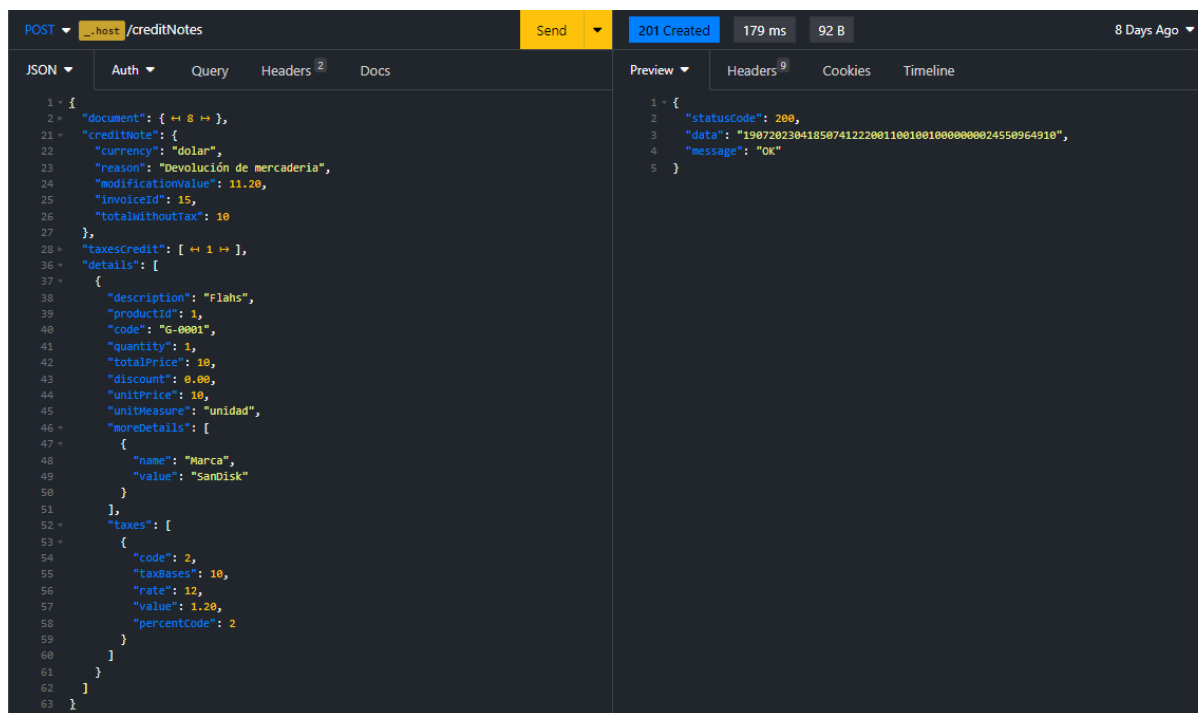
Creación del controlador para crear la nota de crédito

```
1 @Roles(RoleEnum.BUSINESS_ADMIN, RoleEnum.SELLER)
2 @Post()
3 async create(@Req() req: Request, @Body() payload: CreateCreditNoteDto) {
4     // recovery business values
5     const business = await this.businessService.recoverySignatureByBranch(
6         payload.document.branchId,
7     );
8     // check scopes
9     const user = req.user as PayloadToken;
10    await this.businessService.validateUserScope(
11        user.sub,
12        business.businessId,
13        [RoleEnum.BUSINESS_ADMIN],
14        user.roles.includes(RoleEnum.SUPER_ADMIN),
15    );
16    return this.creditServices.create(payload, business.sign);
17 }
```

Nota. Se crea el controlador para generar la Api de tipo POST para consumir y crear la factura.

Figura 14

Consumo del endpoint para crear la nota de crédito



```
POST ...host /creditNotes 201 Created 179 ms 92 B 8 Days Ago
JSON Auth Query Headers 2 Docs Preview Headers 9 Cookies Timeline
1 {
2   "document": { ↵ ↵ },
21  "creditnote": {
22    "currency": "dolar",
23    "reason": "Devolución de mercadería",
24    "modificationValue": 11.20,
25    "invoiceId": 15,
26    "totalWithoutTax": 10
27  },
28  "taxesCredit": [ ↵ ↵ ],
36  "details": [
37    {
38      "description": "Flahs",
39      "productId": 1,
40      "code": "G-0001",
41      "quantity": 1,
42      "totalPrice": 10,
43      "discount": 0.00,
44      "unitPrice": 10,
45      "unitMeasure": "unidad",
46      "moreDetails": [
47        {
48          "name": "Marca",
49          "value": "SanDisk"
50        }
51      ],
52      "taxes": [
53        {
54          "code": 2,
55          "taxBases": 10,
56          "rate": 12,
57          "value": 1.20,
58          "percentcode": 2
59        }
60      ]
61    }
62  ]
63 }
```

Nota. En la figura se muestra como consumir la Api a través de la herramienta Insomnia para crear, emitir y autorizar una nota de crédito

Figura 15

Creación del controlador para el paginado y filtrado de los documentos

```

1 @Roles(RoleEnum.SUPER_ADMIN, RoleEnum.BUSINESS_ADMIN)
2 @Get(':businessId')
3 async findAllByBusiness(
4   @Req() req: Request,
5   @Param('businessId') businessId: number,
6   @Query() params: FilterParamsDocumentDto,
7 ) {
8   const user = req.user as PayloadToken;
9   await this.businessService.validateUserScope(
10    user.sub,
11    businessId,
12    [RoleEnum.BUSINESS_ADMIN],
13    user.roles.includes(RoleEnum.SUPER_ADMIN),
14  );
15  return this.findService.findDocuments(businessId, params);
16 }

```

Nota. Se crea el controlador para generar la Api de tipo GET para consumir y filtrar el módulo de documentos.

Figura 16

Consumo del endpoint para filtrar los documentos

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** `https://api.invoicing.quibox.co/documents/2?q=alex&codDoc=06&state=authorized`
- Status:** 200 OK
- Response Time:** 139 ms
- Response Size:** 8.5 KB
- Query Parameters:**
 - `q`: alex
 - `codDoc`: 06
 - `state`: authorized
 - `ptoEmi`: 001
 - `environment`: 1
 - `dateAfter`: 17-05-2023
 - `dateBefore`: 27-07-2023
 - `limit`: 5
 - `offset`: 5
- Response Body (JSON):**

```

1 {
2   "statusCode": 200,
3   "data": {
4     "items": [
5       {
6         "documentId": 25,
7         "documentCode": "06",
8         "environment": "1",
9         "sequential": 7,
10        "keyAccess": "210720230618507412220011001001000000075607976515",
11        "typeEmission": "1",
12        "dateEmission": "2023-07-21",
13        "ptoEmi": "001",
14        "establishment": "001",
15        "state": "authorized",
16        "authorizationDate": "2023-07-21",
17        "authorizationNum": "210720230618507412220011001001000000075607976515",
18        "moreInformation": [
19          {
20            "name": "email",
21            "value": "fabricio10.pc@gmail.com"
22          }
23        ],
24        "createdAt": "2023-07-21T16:17:07.970Z",
25        "updatedAt": "2023-07-21T16:17:32.889Z",
26        "branchId": 2,
27        "customerId": 3,
28        "branch": {
29          "branchId": 2,
30          "ptoEmi": "001",
31          "establishment": "001",
32          "address": "BAÑOS DE AGUA SANTA, VIA A BAÑOS SIN, A DOS CUADRAS DEL CENTRO DE SALUD, CASA DE UN PISO, COLOR CREMA, CASERIO RIO BLANCO YUNGUILLA",
33          "name": "Ambato",
34          "province": "Cotacachi",
35          "city": "Ambato",
36          "createdAt": "2023-07-17T17:00:35.045Z",
37          "updatedAt": "2023-07-18T17:38:52.345Z",
38          "businessId": 2,
39          "business": {
40            "businessId": 2

```

Nota. En la figura se muestra como consumir la Api utilizando query para el filtrado a través de la herramienta Insomnia en la búsqueda de documentos

Nota. En el Sprint 05 se desarrolló sin ningún inconveniente las actividades del Sprint respectivo.

Sprint 06: Generación de Documentación de los Comprobantes Electrónicos e introducción a pipelines CI/CD

Para el desarrollo del Sprint 06, se realizó la actividad de investigar, cómo crear los pipelines para la documentación y el despliegue del sistema mediante el uso de GitHub Actions y las historias de usuarios 17-18-19 que se detallan a continuación nos ayuda a crear una documentación intuitiva de los endpoints para la emisión y autorización de comprobantes electrónicos.

Historias de Usuario detalladas

Historias de Usuario

Número: HU-17

Usuario: Desarrollador

Nombre historia: Generar la documentación con el estándar OpenApi utilizando redocly

Prioridad: Media

Riesgo en desarrollo: Baja

Puntos estimados (horas): 18

Interacción asignada: 1

Programadores responsables: Karla Molina, Alex Pachacama

Descripción: Como desarrollador se requiere una documentación de los comprobantes electrónicos utilizando el formato OpenApi utilizando redocly, para proporcionar una información detallada de la estructura de las APIs para la emisión y autorización de comprobantes electrónicos

Validación (Criterios de aceptación):

Historias de Usuario

- El sistema permitirá visualizar la documentación mediante el uso de un servidor estático localmente.
 - El sistema permitirá obtener un archivo en formato yml para generar la documentación utilizando el estándar OpenApi
-

Historias de Usuario

Número: HU-18

Usuario: Cliente

Nombre historia: Crear el histórico del inventario y proveedores

Prioridad: Baja

Riesgo en desarrollo: Baja

Puntos estimados (horas): 3

Interacción asignada: 1

Programadores responsables: Karla Molina, Alex Pachacama

Descripción: Como cliente quiero tener un historial del inventario y del proveedor del producto para el control y análisis del comportamiento de datos dentro de la empresa.

Validación (Criterios de aceptación):

- El sistema permitirá visualizar el historial del producto a través del endpoint que consulta el producto
 - El sistema permitirá visualizar el historial del inventario a través del endpoint que consulta el producto
-

Historias de Usuario

Número: HU-19

Usuario: Cliente

Nombre historia: Agregar la validación de los cálculos de la factura

Prioridad: Media

Riesgo en desarrollo: Baja

Historias de Usuario

Puntos estimados (horas): 6

Interacción asignada: 1

Programadores responsables: Karla Molina, Alex Pachacama

Descripción: Como cliente quiero validar los cálculos de los detalles de una factura, para poder crear una factura y controlar posibles errores que presenta el SRI.

Validación (Criterios de aceptación):

- El sistema permitirá crear facturas, solo si los cálculos están correctos, caso contrario saldrá el error.
-

Tabla 10

Sprint Backlog 06

Sprint 06						
Fecha Inicio:		Fecha Fin:		Jornada:		
12/06/2023		30/06/2023		3 horas diarias		
H.U	Tareas	Horas	Fecha Inicio	Fecha Fin	Encargados	Estado
AC						
17	Generar la documentación con el estándar OpenApi utilizando redocly	18	12/06/2023	19/06/2023	Karla Molina, Alex Pachacama	Finalizado
18	Crear el histórico del inventario y proveedores	3	20/06/2023	20/06/2023	Karla Molina, Alex Pachacama	Finalizado

Sprint 06						
	Agregar la validación				Karla	
19	de los cálculos de la factura	6	21/06/2023	22/06/2023	Molina, Alex	Finalizado
	Investigar para crear				Pachacama	
09	los pipelines para la documentación y el despliegue del sistema	18	23/06/2023	30/06/2023	Molina, Alex Pachacama	Finalizado

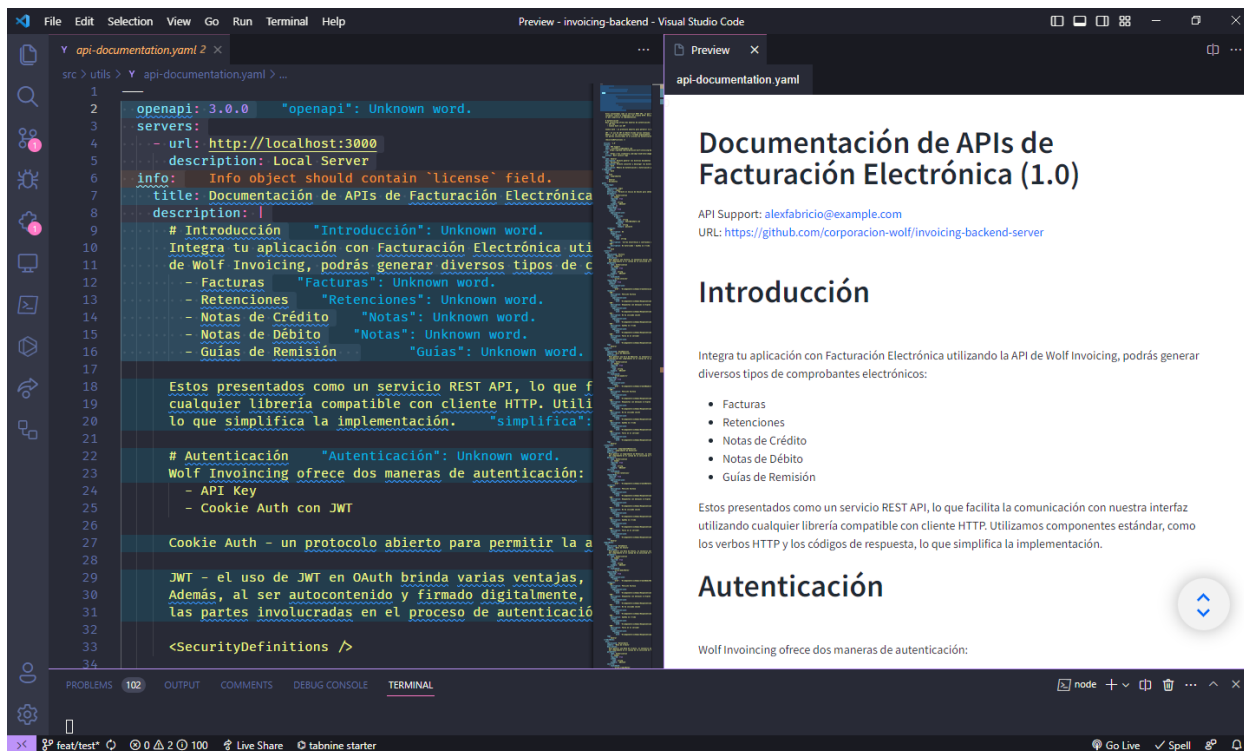
Nota. En la tabla se muestran detalles de las tareas que se realizaron durante el Sprint, quien es el responsable, horas, fecha de inicio y final de cada historia de usuario y del sprint, además del estado si ha sido finalizada, completando las 45 horas durante las tres semanas sin contar fines de semana.

Resultados

En el Sprint 06 se logró implementar los pipelines para generar la documentación mediante el servidor estático que se utilizara al momento del despliegue con AWS, validar si los cálculos de la factura son verdaderos se crea caso contrario salta error. A continuación, se detallan los resultados las historias de usuario realizadas y finalizadas:

Figura 17

Documentación generada con OpenApi



Nota. En la figura se muestra la documentación generada para los comprobantes electrónicos mediante el archivo en formato yml que será implementada en los pipelines

Figura 18

Validación de cálculos de la factura

```

POST ..._host /invoices
Send 400 Bad Request 68.6 ms 72 B Just Now

JSON Auth Query Headers Docs
26 "totalWithoutDiscount": 20
27 },
28 "taxesInvoices": [
29 {
30 "code": 2,
31 "percentCode": 2,
32 "taxBases": 20,
33 "additionalDiscount": 0,
34 "value": 2.40
35 }
36 ],
37 "payment": [
38 {
39 "total": 22.40,
40 "paymentForm": "01"
41 }
42 ],
43 "details": [
44 {
45 "description": "Flash",
46 "productId": 1,
47 "code": "G-0001",
48 "quantity": 3,
49 "totalPrice": 20,
50 "discount": 0.00,
51 "unitPrice": 10,
52 "unitMeasure": "unidad",
53 "taxes": [
54 {
55 "code": 2,
56 "taxBases": 20,
57 "rate": 12,
58 "value": 2.40,
59 "percentCode": 2
60 }
61 ]
62 }
63 ]
64 }

Preview Headers Cookies Timeline
1 {
2 "statusCode": 400,
3 "message": "El precio total de un producto es erroneo"
4 }

```

Nota. En la figura se muestra un ejemplo del error que puede saltar si algún dato esta mal al momento de crear la factura.

Nota. En el Sprint 06 se desarrolló sin ningún inconveniente las actividades del Sprint respectivo, los pipelines de la documentación se generaron dentro de un servidor de pruebas con GitHub Actions antes de ser desplegada en AWS.

Sprint 07: Implementación de Test Unitarios y Despliegue del Servidor AWS

Para el desarrollo del Sprint 06, las historias de usuarios 20-21-22-23 que se detallan a continuación, se hace la implementación con los pipelines para el despliegue de la aplicación, documentación en AWS, además, se genera test unitarios a utilidades que se ha usado dentro del desarrollo del sistema.

Historias de Usuario detalladas

Historias de Usuario

Número: HU-20

Usuario: Desarrollador

Nombre historia: Crear test unitarios de utilidades

Prioridad: Media

Riesgo en desarrollo: Baja

Puntos estimados (horas): 12

Interacción asignada: 1

Programadores responsables: Karla Molina, Alex Pachacama

Descripción: Como desarrollador quiero testear las utilidades, para validación de la funcionalidad.

Validación (Criterios de aceptación):

- El sistema permitirá realizar los test a la función de validar el DNI
- El sistema permitirá realizar los test a la función de generar el clave de acceso del comprobante electrónico
- El sistema permitirá realizar los test a la función de generar el código de barras de la clave de acceso
- El sistema permitirá realizar los test a la función de los cálculos de la factura

Historias de Usuario

Número: HU-21

Usuario: Desarrollador

Nombre historia: Implementar el pipeline de integración y entrega continua CI/CD del sistema

Prioridad: Alta

Riesgo en desarrollo: Baja

Puntos estimados (horas): 15

Interacción asignada: 1

Programadores responsables: Karla Molina, Alex Pachacama

Historias de Usuario

Descripción: Como desarrollador quiero automatizar el proceso del despliegue en AWS mediante pipelines de GitHub Actions, para la reducción del tiempo en el proceso.

Validación (Criterios de aceptación):

- El sistema permitirá la integración de un pipeline mediante GitHub Actions, para verificar si existen nuevas migraciones del proyecto, mediante un comentario en las pull request
 - El sistema permitirá la integración de un pipeline mediante GitHub Actions para validar los test Unitarios, mediante un comentario en las pull request
 - El sistema permitirá la integración de un pipeline mediante GitHub Actions para verificar la sintaxis del código, mediante un comentario en las pull request
 - El sistema permitirá la integración de un pipeline mediante GitHub Actions para el despliegue del proyecto en AWS
-

Historias de Usuario

Número: HU-22

Usuario: Desarrollador

Nombre historia: Configurar el servidor para el despliegue en AWS

Prioridad: Alta

Riesgo en desarrollo: Baja

Puntos estimados (horas): 9

Interacción asignada: 1

Programadores responsables: José Ochoa

Descripción: Como desarrollador quiero configurar el AWS para el despliegue del proyecto y puesta en producción.

Validación (Criterios de aceptación):

- N/A
-

Historias de Usuario

Número: HU-23**Usuario:** Desarrollador**Nombre historia:** Realizar pruebas de aceptación de los múltiples EndPoint del sistema**Prioridad:** Media**Riesgo en desarrollo:** Baja**Puntos estimados (horas):** 9**Interacción asignada:** 1**Programadores responsables:** Karla Molina, Alex Pachacama

Descripción: Como desarrollador quiero comprobar las funcionalidades de mi sistema en producción para comprobar su funcionalidad y resolver posibles errores encontrados.

Validación (Criterios de aceptación):

- El sistema permitirá consumir las APIs en un ambiente de producción.
-

Tabla 11*Sprint Backlog 07*

Sprint 07						
Fecha Inicio:			Fecha Fin:		Jornada:	
03/07/2023			21/07/2023		3 horas diarias	
H.U	Tareas	Horas	Fecha Inicio	Fecha Fin	Encargados	Estado
20	Crear test unitarios de utilidades	12	03/07/2023	06/07/2023	Karla Molina, Alex Pachacama	Finalizado
21	Implementar el pipeline de integración y	15	07/07/2023	13/07/2023	Karla Molina, Alex Pachacama	Finalizado

Sprint 07

	entrega continua CI/CD del sistema						
	Configurar el servidor para el despliegue en AWS	9	14/07/2023	18/07/2023	José Ochoa	Finalizado	
22	Realizar pruebas de aceptación de los múltiples EndPoint del sistema	9	19/07/2023	21/07/2023	Karla Molina, Alex Pachacama	Finalizado	

Nota. En la tabla se muestran detalles de las tareas que se realizaron durante el Sprint, quien es el responsable, horas, fecha de inicio y final de cada historia de usuario y del sprint, además del estado si ha sido finalizada, completando las 45 horas durante las tres semanas sin contar fines de semana.

Resultados

En el Sprint 07 se logró desplegar el sistema con los pipelines de GitHub Actions y se comprobó la funcionalidad de cada API creada en un entorno de producción, se detectó errores y se los corrigió, mediante los pipelines creados se automatizó el proceso para su despliegue gracias a los, incluyendo la documentación del sistema. A continuación, se detallan los resultados las historias de usuario realizadas y finalizadas:

Figura 19

Creación del test unitario del DNI

```
1 test('DNI of more than 10 numbers', () => {
2   expect(
3     IsDniMessageConstraint({
4       value: '1727655944001',
5       constraints: [],
6       targetName: '',
7       object: {},
8       property: '',
9     }),
10  ).toBe('La cedula debe tener 10 digitos');
11 });
```

Nota. En la figura se muestra la creación del test unitario para comprobar el DNI o CI no tenga más de 10 números.

Figura 21

Documentación desplegada en AWS

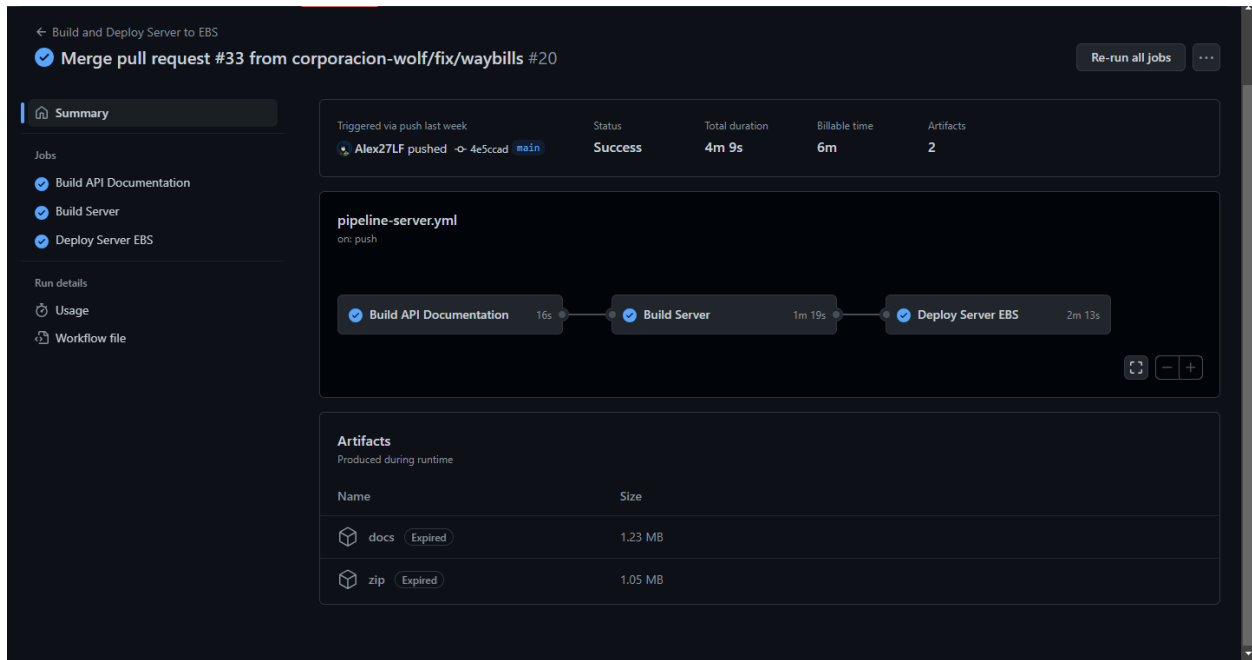
The screenshot displays the OpenAPI documentation for the 'Factura' endpoint. The page is titled 'Factura' and includes a search bar and a navigation menu. The main content area shows the following details:

- Endpoint:** POST /invoices
- Authorization:** CookieAuth
- Header Parameters:** Authorization (string, required, Example: ABC1234)
- Request Body Schema:** application/json
- Request Body Schema Details:**
 - document (object, required, CreateDocumentDto)
 - invoice (object, required, InvoiceDto)
 - payment (Array of objects, required, CreatePaymentDto) - Se define el Método de Pago a realizar en la Factura
 - taxesInvoices (Array of objects, required, CreateTaxInvoiceDto)
 - details (Array of objects, required, CreateDetailDto)
- Request Samples:** A sample JSON payload for the POST request, including fields like environment, sequential, typeEmission, dateEmission, moreInformation, ptoEmi, establishment, branchId, customerId, invoice, wmybillId, currency, ivaValue, rentValue, tip, totalDiscount, and amount.

Nota. En la figura se muestra la documentación que se generó con el OpenApi de los comprobantes electrónicos ya desplegado con el dominio determinado.

Figura 22

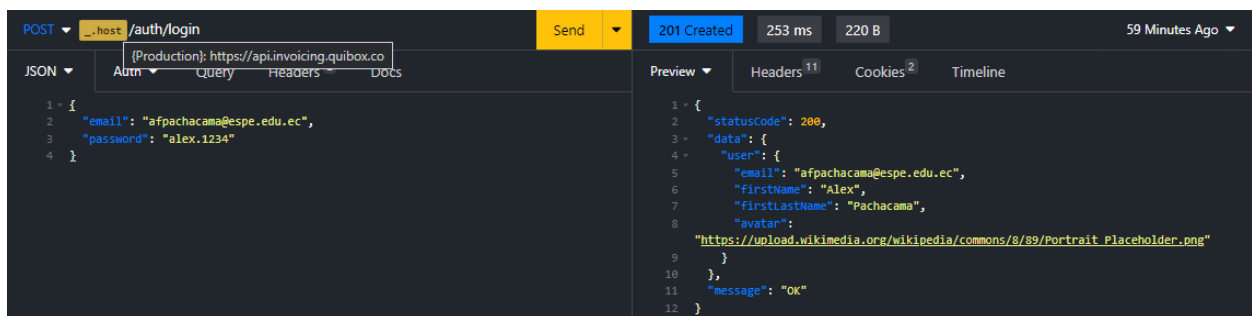
Flujo de trabajo de los pipelines de GitHub Actions



Nota. En la figura se muestra la Actions del GitHub con el flujo de trabajo que se implementó en el pipeline para el despliegue del sistema con la funcionalidad correcta.

Figura 23

Pruebas de Aceptación de la API y sus múltiples endpoints



Nota. En la figura se muestra un ejemplo de pruebas de aceptación comprobando la funcionalidad de la Api de inicio de sesión en el ambiente de producción a través de Insomnia

debido a que el proyecto se enfocó solo en el desarrollo del backend. Además, en el Anexo 9 se adjunta el resto de pruebas de aceptación de las APIs implementadas.

Nota. En el Sprint 07 se desarrolló sin ningún inconveniente las actividades del Sprint respectivo.

Capítulo IV

Marco de Resultados

En este capítulo se detallan los resultados que se obtuvieron al culminar con el desarrollo del backend para el sistema de facturación electrónica para la empresa “Corporación Wolf S.A.”.

Para medir la eficiencia del despliegue del sistema mediante el uso de los pipelines y los costos, se ha utilizado métodos tradicionales o empíricos.

Según (PhD et al., 2018) señalan que hay cuatro buenas prácticas que pueden ayudar a ver el rendimiento al momento de utilizar el enfoque DevOps con pipelines CI/CD en un proyecto, entre ellas son:

- **Frecuencia de Implementación (Deployment Frequency):** Consiste en monitorear la frecuencia de las implementaciones dentro del proyecto.
- **Plazo de Ejecución para Cambios (Lead Time For Changes):** El tiempo que lleva la pull request y la confirmación que el encargado de revisar lo realiza.
- **Tiempo Medio de Restauración (MTTR):** Mide la capacidad de respuesta para resolver o implementar las soluciones dentro del proyecto en un ambiente de producción.
- **Tasa de Fallo de Cambio (Change Failure Rate):** Es la tasa de errores del proyecto cuando existe un fallo o una implementación a través de los pipelines.

A continuación, en la Tabla 12 se muestran las métricas o buenas prácticas al momento de utilizar un enfoque DevOps para la entrega de un sistema:

Tabla 12*Métricas del Rendimiento de Entrega*

	Rendimiento Alto	Rendimiento Medio	Rendimiento Bajo
Frecuencia de Implementación	Bajo demanda (Varios despliegues al día)	Entre una vez a la semana o al mes	Entre una vez por semana y una vez por mes *
Tiempo de Espera para Cambios	Menos de una hora	Entre una semana o un mes	Entre una semana y un mes *
MTTR	Menos de una hora	Menos de un día	Entre un día y una semana
Tasa de Fallo de Cambio	0 – 15 %	0 – 15 %	31 - 45 % *

Nota. Tabla: *Los resultados bajos son inferiores por término medio, pero tenían la misma mediana que los de rendimiento medio. Tomada (PhD et al., 2018).

En el proyecto que se ha desarrollado del Backend se ha utilizado la Tabla 12 para ver si cumple con las métricas o buenas prácticas y ver si tiene un buen rendimiento al momento de realizar el despliegue.

Tabla 13*Rendimiento de Entrega del Proyecto Desarrollado*

	Rendimiento Alto	Rendimiento Medio	Rendimiento Bajo
Frecuencia de Implementación	✓		
Tiempo de Espera para Cambios	✓		

	Rendimiento Alto	Rendimiento Medio	Rendimiento Bajo
MTTR		✓	
Tasa de Fallo de Cambio		✓	

Nota. Los resultados del proyecto siguiendo las buenas prácticas cumplen con un rendimiento esperado con la entrega y su despliegue continuo CI/CD.

Rendimiento de despliegue

Al terminar de desarrollar el software y hacer el despliegue se construyó pipelines CI/CD que nos permite automatizar el proceso de compilación, prueba y despliegue del sistema, mediante la utilización de GitHub Actions, debido a que el código está alojado en GitHub y mientras se realiza un push a la rama principal, las máquinas virtuales de GitHub construyen la aplicación mediante el pipeline CD que es la encargada de desplegar en AWS, previamente debe pasar por una pull request donde a cada commit se le realiza el escaneo del código, migraciones y los test unitarios a las utilidades del sistema mediante pipelines CI, mientras este no este validado, no se puede realizar el despliegue, por lo que se debe corregir los errores que existan.

Para medir el rendimiento del despliegue del sistema desarrollado utilizando pipelines con GitHub Actions, se ha utilizado el método empírico cuantitativo que tiene como objetivo obtener conocimiento a través de la observación, experiencia, etc., generando resultados que son expresados matemáticamente.

Se definió dos objetos de estudio:

- Desplegar el proyecto manualmente
- Desplegar el proyecto utilizando pipeline CI/CD mediante GitHub Actions

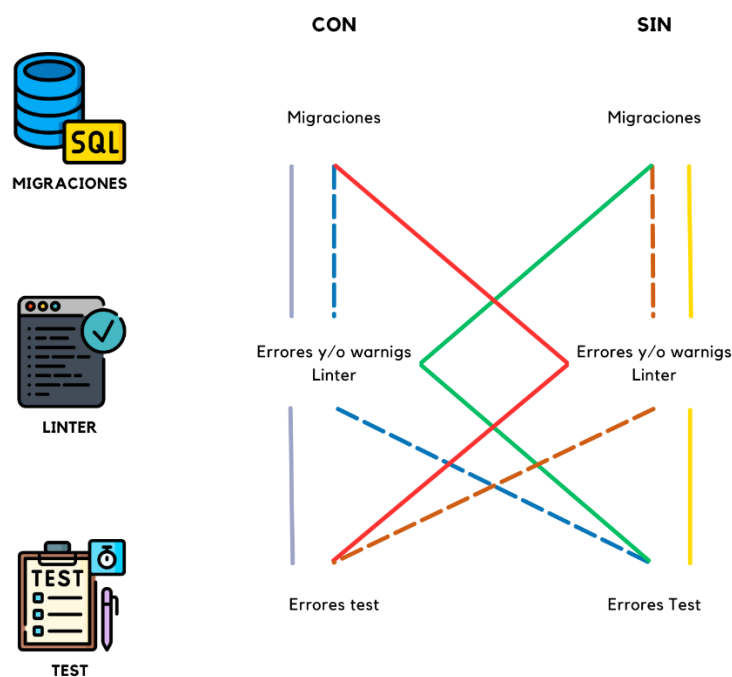
Para poder medir los objetos de estudio definidos anteriormente se ha propuesto hacerlo de manera tradicional, es decir, con un cronómetro que medirá el tiempo de ejecución al ejecutar cada acción, obteniendo el mismo resultado en los dos objetos de estudio. La población que se utilizará será:

- **Integración Continua (CI):** El proceso que se realizan dentro del CI mediante los pipelines son independientemente uno del otro, para lo cual, no es necesario que si una termina de ejecutarse empieza la otra, mientras que, si lo hacemos manualmente se debe hacer uno por uno, ya que su proceso no es automatizado.
 - CI con Migraciones
 - CI con errores y/o advertencias (warnings) en el Código por Linter
 - CI con errores en los Test
 - CI sin Migraciones
 - CI sin errores en el Código por Linter
 - CI sin errores en los Test
- **Entrega Continua (CD):** El proceso que se realiza dentro del CD tanto en los pipelines como manualmente son dependientes uno del otro, es decir, si hay un error no se puede pasar a la siguiente acción para su despliegue, por lo que la persona que esté a cargo del despliegue del proyecto debe revisar y solucionarlo.
 - Construir la documentación (Build Documentation)
 - Construir el proyecto (Build Server)
 - Despliegue al EBS

Una vez definido la población tanto para el CI como el CD, se ha definido la muestra para los CI con todas las combinaciones o casos que se pueden dar. A continuación, en la Figura se muestra las 6 combinaciones del CI.

Figura 24

Casos del CI



Nota. En la figura se muestra los posibles casos que se pueden dar en CI en los dos objetos de estudio cada uno representado por un color diferente, haciéndoles grupos de 3 sin repetir,

Caso 1: El primer caso especifica que no existen migraciones, que el escaneo de código por medio del Linter y los Tests no tienen errores, obteniendo los resultados que se muestran en la Tabla 14.

Tabla 14*Resultados del Caso 1*

	Integración Manual	Integración Automatizada
Escaneo de Código	79,16 s	21 s
Detectar Migraciones	10,35 s	0 s
Test Unitarios	76,20 s	15 s
Total	165,71 s	21 s

Caso 2: El segundo caso especifica que existen migraciones, que el escaneo de código por medio del Linter y los Tests tienen errores, obteniendo los resultados que se muestran en la Tabla 15.

Tabla 15*Resultados del Caso 2*

	Integración Manual	Integración Automatizada
Escaneo de Código	95,80 s	24 s
Detectar Migraciones	32,02 s	6 s
Test Unitarios	96,20 s	21 s
Total	224,02	24 s

Caso 3: El tercer caso especifica que no existen migraciones, que el escaneo de código por medio del Linter da errores y los Tests no tiene errores, obteniendo los resultados que se muestran en la Tabla 16.

Tabla 16*Resultados del Caso 3*

	Integración Manual	Integración Automatizada
Escaneo de Código	96,70 s	23 s
Detectar Migraciones	9,28 s	0 s
Test Unitarios	97,27 s	21 s
Total	203,25 s	23 s

Caso 4: El cuarto caso especifica que no existen migraciones, que el escaneo de código por medio del Linter no da errores y los Tests tiene errores, obteniendo los resultados que se muestran en la Tabla 17.

Tabla 17*Resultados del Caso 4*

	Integración Manual	Integración Automatizada
Escaneo de Código	78,93 s	22 s
Detectar Migraciones	8,20 s	0 s
Test Unitarios	93,85 s	21 s
Total	109,98	22 s

Caso 5: El quinto especifica que existen migraciones, que el escaneo de código por medio del Linter da errores y los Tests no tiene errores, obteniendo los resultados que se muestran en la Tabla 18.

Tabla 18*Resultados del Caso 5*

	Integración Manual	Integración Automatizada
Escaneo de Código	93,64 s	24 s
Detectar Migraciones	28,47 s	4 s
Test Unitarios	85,16 s	18 s
Total	207,27 s	24 s

Caso 6: El sexto caso especifica que existen migraciones, que el escaneo de código por medio del Linter no da errores y los Tests tiene errores, obteniendo los resultados que se muestran en la Tabla 19.

Tabla 19*Resultados del Caso 6*

	Integración Manual	Integración Automatizada
Escaneo de Código	77,16 s	21 s
Detectar Migraciones	8,97 s	5 s
Test Unitarios	94,93 s	24 s
Total	181,06	24 s

Nota. En la Tabla 14, Tabla 15, Tabla 16, Tabla 17, Tabla 18, Tabla 19: Se muestran los resultados obtenidos a cada caso respectivo omitiendo el tiempo en que tarda en instalar las dependencias, con los dos objetos de estudio que son manual y automatizado o pipelines.

En el CD hay dos casos, en las que todo está correcto o que tenga un error, estos deben ser corregidos o resueltos, para que este correcto y se realice el despliegue.

Caso 1: El primer caso especifica que la documentación, la construcción del servidor (Build Server) y el despliegue al EBS (Deploy EBS) están correctamente, obteniendo los resultados que se muestran en la Tabla 20.

Tabla 20

Resultados del Caso 1 CD

	Despliegue Manual	Despliegue automatizado
Build de la Documentación	73,33 s	19 s
Build del Proyecto	109,29 s	37 s
Deploy EBS	145,33 s	122 s
Total	327,95 s	178 s

Caso 2: El segundo caso especifica que la documentación, la construcción del servidor (Build Server) y el despliegue al EBS (Deploy EBS) no están correctamente, obteniendo los resultados que se muestran en la Tabla 21.

Tabla 21

Resultados del Caso 2 CD

	Despliegue Manual	Despliegue automatizado
Build de la Documentación	71,05 s	17 s
Build del Proyecto	110,32 s	32 s
Deploy EBS	33,33 s detecta el error	8 s detecta el error
Total	214,70 s	57 s

Nota. En la Tabla 20, Tabla 21: Se muestran los resultados obtenidos a cada caso respectivo del CD, con los dos objetos de estudio que son manual y automatizado o pipelines. Cuando da error en cualquier acción no se realiza el CD manteniendo la versión anterior.

Una vez obtenido todos los resultados en la Tabla 22 se muestra una media del tiempo de los resultados que se obtuvo, un tiempo de 0 s al momento de hacerlo manualmente y de 0 s al utilizar de manera automatizada mediante los pipelines, quienes tienen las acciones definidas de lo que deben hacer, comparando los resultados de tiempo se llega a la conclusión de que si lo hacemos de manera automatizada mediante los pipelines nos ahorra mucho tiempo porque no hay necesidad de escribir cada comando ya que las Actions de GitHub ejecutan de manera paralela, teniendo un buen rendimiento al momento de hacer un despliegue.

Tabla 22

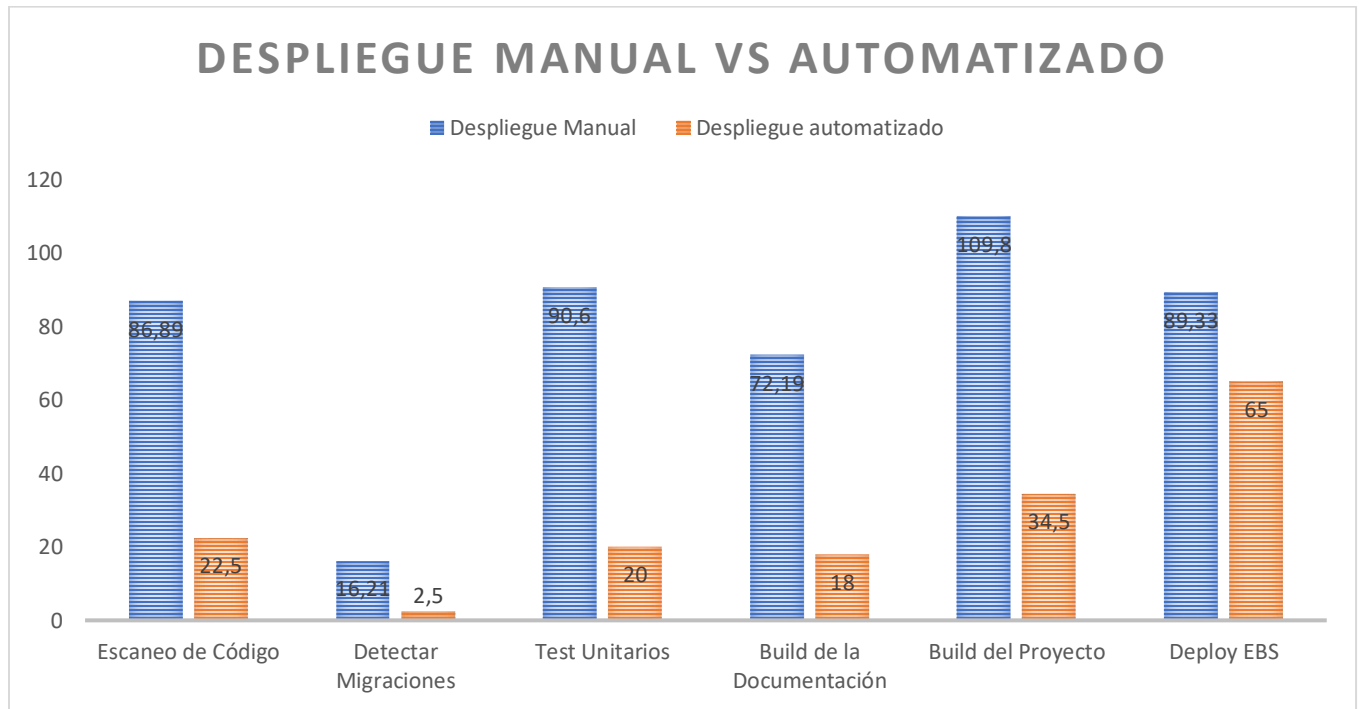
Comparación de Tiempo entre el Despliegue Manual y Automatizado

	Despliegue Manual	Despliegue automatizado
Escaneo de Código	86,89 s	22,50 s
Detectar Migraciones	16,21 s	2,50 s
Test Unitarios	90,60 s	20 s
Build de la Documentación	72,19 s	18 s
Build del Proyecto	109,80 s	34,50 s
Deploy EBS	89,33 s	65 s

Nota. Se muestran los resultados obtenidos del promedio de los casos que se ha realizado para el CI/CD

Figura 25

Gráfico Despliegue Manual vs Automatizado



Nota. Gráfico que compara el rendimiento del CI/CD manualmente y automatizado.

En el Gráfico y en la Tabla anterior se evidencia una reducción en el tiempo sumamente considerable al desplegar mediante los pipelines de integración y entrega continua CI/CD a través de las GitHub Actions de manera automatizada, que hacerlo manualmente, por lo que, existe un buen rendimiento al desplegar el sistema, además, no es requerido que el responsable de llevar a cabo el despliegue tenga que memorizar los comandos y procedimientos, los cuales, están propensos a errores. Esto disminuye las horas de trabajo necesarias y, en cambio incrementa su eficiencia al enfocarse en otras actividades.

Análisis comparativo de costos

Se realiza el análisis de costos entre el servicio externo de facturación electrónica de terceros que utiliza la empresa actualmente y el sistema desarrollado en la empresa, para determinar cuál es más rentable. A continuación, se detallan los dos tipos de servicios para poder realizar el análisis:

- **Servicio Externo de Facturación Electrónica**
 - Tarifa anual: \$340
 - Tarifa mensual: \$28.33
 - Número de Documentos Electrónicos: 1200

- **Servicio Propio**
 - Tarifa anual: \$300
 - Tarifa mensual: \$25
 - Número de Documentos Electrónicos: 1300
 - Costo Mensual de proveedor en la nube: \$20 mínimo

Tomando en cuenta el precio del servicio externo de facturación electrónica antes mencionada, se concluye que, la empresa reduciría un egreso anual de \$340, ya que pasarían a pagar el costo del proveedor de la nube mensualmente \$20 mínimo, además, el número de documentos electrónicos pasaría de 1200 a ser ilimitado para la empresa.

La empresa “Corporación Wolf S.A.”, tiene una proyección a futuro para el servicio propio de facturación electrónica estimado en un costo anual de \$300 dólares a terceros, teniendo así, un ingreso neto mensual de \$25 por empresa con 1300 documentos que incluye facturas, guías de remisión, comprobantes de retención, nota de crédito y débito, con la gestión de usuarios (clientes, transportistas y sucursales de la empresa) e inventarios de productos. Además, como es un servicio propio se adapta a las necesidades de la empresa.

Es importante considerar que este servicio de facturación electrónica interno, contribuye a los ingresos de la empresa. En otras palabras, si contamos con tres usuarios que contratan el servicio durante el año, esto generaría un ingreso total de \$900. Mientras tanto, los gastos anuales de la empresa ascenderían a \$250 al pagar el proveedor de servicios en la nube que se encuentra alojado el servicio, lo que resultaría en una ganancia de \$650.

Conclusiones y recomendaciones

Conclusiones

Al finalizar el trabajo de integración curricular y después de haber analizado los resultados del rendimiento del despliegue y la proyección de costos del backend de facturación electrónica de la empresa “Corporación Wolf S.A.”, se llegaron a las siguientes conclusiones:

- Tras revisar el estado de arte se concluye que hay sistemas implementados para la facturación electrónica, sin embargo, la documentación al respecto es limitada. Al igual que no hay documentación o evidencias que en el sistema se aplique el enfoque DevOps mediante la utilización de pipelines y computación en la nube, generando un apoyo de investigación para estudios próximos.
- Se desarrollo el backend de facturación electrónica para la empresa “Corporación Wolf S.A.” usando el enfoque DevOps a través de la utilización de pipelines CI/CD, para lo cual, se tomó en cuenta los requisitos y estructuras que el SRI solicita para hacer todo el proceso de recepción y autorización de cada comprobante electrónico que se implementó dentro del sistema
- Se concluye que la utilización de GitHub Actions, permite hacer el despliegue de manera sencilla y automatizada al servidor en la nube de AWS.
- Se utilizó tecnologías de última generación para implementar en un sistema de facturación como NestJS con TypeORM, debido a que es muy conocida por el enfoque ha aplicaciones de servidor y API Rest en la nube, teniendo una escalabilidad, adaptabilidad y facilidad de implementar funcionalidades dentro de cada proyecto.
- Para la medición del rendimiento de los pipelines CI/CD no se evidencia un método o métricas, más bien, solo hay buenas prácticas que han ido apareciendo a través de la experiencia, para lo cual, se optó medir de forma

empírica a través de la observación cuantitativa de los dos objetos de estudio que se definieron.

- Tras analizar los resultados en segundos que se obtuvo para los 6 casos del CI y los dos casos del CD en el despliegue manual y automatizado, a través de los pipelines que se crearon con GitHub Actions, se obtiene un mejor rendimiento cuando se hace de manera automatizada disminuyendo una considerable cantidad de tiempo con el proceso manual que esta propenso a errores, lo que hace que la implementación de los pipelines CI/CD sea muy eficiente dentro del proyecto.

Recomendaciones

- Se recomienda conocer bien el proceso de facturación electrónica y emisión de los comprobantes, además, de su estructura xml.
- Se recomienda considerar el uso del framework NestJS con TypeORM en proyectos posteriores, dado que simplifica la escritura de código con un enfoque en buenas prácticas, esto hace que el mantenimiento del código fuente sea más fácil y sencillo.
- Es aconsejable implementar los pipelines CI/CD, ya que esto agilizará la automatización de los procesos de integración y entrega del proyecto.

Bibliografía

- Amazon Web Services Inc. (2023a). *Características del producto Amazon CloudWatch – Amazon Web Services (AWS)*. Amazon Web Services, Inc.
<https://aws.amazon.com/es/cloudwatch/features/>
- Amazon Web Services Inc. (2023b). *¿Qué es Amazon Relational Database Service (Amazon RDS)? - Amazon Relational Database Service*.
https://docs.aws.amazon.com/es_es/AmazonRDS/latest/UserGuide/Welcome.html
- Amazon Web Services Inc. (2023c). *¿Qué es AWS?* Amazon Web Services, Inc.
<https://aws.amazon.com/es/what-is-aws/>
- Amazon Web Services Inc. (2023d). *¿Qué es AWS Elastic Beanstalk? - AWS Elastic Beanstalk*.
https://docs.aws.amazon.com/es_es/elasticbeanstalk/latest/dg/Welcome.html
- Amazon Web Services Inc. (2023e). *What is Amazon S3? - Amazon Simple Storage Service*.
<https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html#BasicsBucket>
- Arias, Á. (2015). *Computación en la Nube: 2ª Edición*. IT Campus Academy.
- Atlassian. (2023). *Scrum—What is it, how it works, & how to start*. Atlassian.
<https://www.atlassian.com/agile/scrum>
- Barker, R. (1994). *El modelo entidad-relación CASE*methodtm*. Ediciones Díaz de Santos.
- Brzóska, B. (s. f.). *Hashids*. Hashids. Recuperado 13 de julio de 2023, de <https://hashids.org/>
- Dátil. (2021). *Dátil*. Dátil. <https://datil.com>
- DBeaver Documentation. (2020, septiembre 16). DBeaver Documentation. *DBeaver*.
<https://dbeaver.com/docs/wiki/>
- Delgado, R. C. (2010). *REGLAMENTO DE COMPROBANTES DE VENTA, RETENCIÓN Y DOCUMENTOS COMPLEMENTARIOS*. 38.
- Devops Latam. (2021, abril 29). *¿Qué es un Pipeline de CI / CD?* *DevOps Latam*.
<https://devopslatam.com/que-es-un-pipeline-de-ci-cd/>

- Díaz, Y., & Fernández, Y. (2012, junio 12). *Patrón Modelo-Vista-Controlador*. | *Telemática*.
<https://revistatelematica.cujae.edu.cu/index.php/tele/article/view/15>
- Excalidraw*. (2023). [TypeScript]. Excalidraw. <https://github.com/excalidraw/excalidraw> (Obra original publicada en 2020)
- GitHub Inc. (2023a). *Hello World—GitHub*. GitHub Docs. <https://ghdocs-prod.azurewebsites.net/en/get-started/quickstart/hello-world>
- GitHub Inc. (2023b). *Understanding GitHub Actions*. GitHub Docs. <https://ghdocs-prod.azurewebsites.net/en/actions/learn-github-actions/understanding-github-actions>
- GitLab B.V. (2022, febrero 10). *What is DevOps?* <https://about.gitlab.com/topics/devops/>
- Gobierno de España. (s. f.). *Factura Electrónica—¿Qué es la factura electrónica ?* Recuperado 13 de julio de 2023, de <https://www.facturae.gob.es/factura-electronica/Paginas/factura-electronica.aspx>
- Google Inc. (s. f.). *¿Qué es cloud computing?* Google Cloud. Recuperado 13 de julio de 2023, de <https://cloud.google.com/learn/what-is-cloud-computing?hl=es>
- Google Inc. (2023, julio 13). *Puppeteer*. <https://pptr.dev/>
- Huerlo, J. R. (2021). *Terraform Como Herramienta Para Automatizar La Creación De Infraestructuras Siguiendo El Concepto “Infraestructura Como Código”* [Thesis, Ecuador - PUCESE - Escuela de Sistemas y Computación].
<http://localhost/xmlui/handle/123456789/2602>
- JGraph Ltd. (2005). *Introduction to diagrams—Draw.io*. <https://www.drawio.com/doc/getting-started-diagram-types>
- Kong Inc. (2021). *Introduction to Insomnia | Insomnia Docs*.
<https://docs.insomnia.rest/insomnia/get-started>
- Kubica, M. (2023, mayo 25). *Xml2js*. Npm. <https://www.npmjs.com/package/xml2js>
- Lara Freire, C. L. (2015). *La facturación electrónica y la rentabilidad de la compañía importadora ALVARADO VÁSCONEZ Cía. Ltda., de la ciudad de Ambato*.

- [BachelorThesis, Universidad Técnica de Ambato. Facultad de Contabilidad y Auditoría. Carrera de Contabilidad y Auditoría. Semipresencial.].
<https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/18531>
- López, M. A. (2021). *DevOps Aplicado a Sistemas IoT: Definición e Implementación de Procesos Continuos de Monitorización y Retroalimentación* [Phd, E.T.S.I de Sistemas Informáticos (UPM)]. <https://oa.upm.es/68043/>
- Mamian, M. (2021a, febrero 3). Conoce quiénes obligados facturar electrónicamente en Ecuador. *Contífico*. <https://contifico.com/obligados-a-facturar-electronicamente/>
- Mamian, M. (2021b, junio 11). Descubre qué es la Facturación Electrónica en Ecuador. *Contífico*. <https://contifico.com/que-es-la-facturacion-electronica-y-como-impulsa-tu-negocio/>
- Merlo, P. C. (2006). *La factura electrónica en el Ecuador* .
- Microsoft. (2012). *Documentation—TypeScript for JavaScript Programmers*.
<https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html>
- Microsoft. (2023, junio 7). *Code Navigation in Visual Studio Code*.
<https://code.visualstudio.com/docs/editor/editingevolved>
- Mysliwiec, K. (2017). *Documentation | NestJS - A progressive Node.js framework*.
Documentation | NestJS - A Progressive Node.Js Framework. <https://docs.nestjs.com>
- NPM Inc. (2021a, julio 22). *Crypto-js*. Npm. <https://www.npmjs.com/package/crypto-js>
- NPM Inc. (2021b, agosto 21). *Exceljs*. Npm. <https://www.npmjs.com/package/exceljs>
- NPM Inc. (2021c, noviembre 16). *Cookie-parser*. Npm. <https://www.npmjs.com/package/cookie-parser>
- NPM Inc. (2022a). *Passport-local*. <https://www.npmjs.com/package/passport-local>
- NPM Inc. (2022b, marzo 30). *Node-forge NPM*. Npm.io. <https://npm.io/package/node-forge>
- NPM Inc. (2022c, octubre 6). *Bcrypt*. Npm. <https://www.npmjs.com/package/bcrypt>

NPM Inc. (2022d, diciembre 24). *Passport-jwt*. Npm. <https://www.npmjs.com/package/passport-jwt>

NPM Inc. (2023a, enero 16). *Swagger-jsdoc*. Npm. <https://www.npmjs.com/package/swagger-jsdoc>

NPM Inc. (2023b, junio 4). *Bwip-js*. Npm. <https://www.npmjs.com/package/bwip-js>

NPM Inc. (2023c, julio 1). *Dayjs*. Npm. <https://www.npmjs.com/package/dayjs>

Ochoa, J. V., & Bone, E. L. (2022). *Desarrollo de una aplicación móvil y backend para apoyar a los conductores en la gestión de reparto de paquetería en la empresa «Corporación Wolf S.A.»*

Page, D. (s. f.). *pgAdmin 4—PgAdmin 4 7.4 documentation*. Recuperado 13 de julio de 2023, de <https://www.pgadmin.org/docs/pgadmin4/development/index.html>

Pazmiño, V. A. (2015). *Análisis de la Implementación de Facturación Electrónica en el Ecuador- Ventajas y Desventajas frente a la Facturación Física*.

PhD, N. F., Humble, J., & Kim, G. (2018). *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. IT Revolution.

Redocly Inc. (2023, junio 4). *Redocly CLI*. Redocly. <https://redocly.com/docs/cli/>

Reinman, A. (s. f.). *Node mailer*. Recuperado 13 de julio de 2023, de <https://nodemailer.com/about/>

Rolfo, M. (2022, mayo 16). *Tutorial de TypeScript: Una guía paso a paso para aprender TypeScript*. Codigoencasa.com. <https://codigoencasa.com/tutorial-de-typescript-una-guia-paso-a-paso-para-aprender-typescript/>

Scolari, S. (2010). *Impacto de reglas de refactorización en diagramas UML con restricciones OCL* [Tesis, Universidad Nacional de La Plata]. <http://sedici.unlp.edu.ar/handle/10915/4011>

SCRUMstudy™. (2013). *A guide to the Scrum Body of knowledge (SBOK Guide)* (2013 edition).

SCRUMstudy, A brand of VMEdu, Inc.

Servicio de Rentas Internas. (2009). *NORMATIVA TRIBUTARIA GENERAL*.

Servicio de Rentas Internas. (2012). *Resolución No. NAC-DGERCGC12-00105*.

https://www.sri.gob.ec/o/sri-portlet-biblioteca-alfresco-internet/descargar/aced2484-8422-45d6-a342-cd414ea26783/Resoluci_n+No.+NAC-DGERCGC12-00105%2C+publicada+en+R.O.+666+de+21-03-2012.pdf

Servicio de Rentas Internas. (2013). *Resolución No. NAC-DGERCGC13-00236*.

https://www.sri.gob.ec/o/sri-portlet-biblioteca-alfresco-internet/descargar/e4350b9d-1d1f-44e9-9b77-37de683a5be1/Resoluci_n+No.+NAC-DGERCGC13-00236%2C+publicada+en+R.O.+956+de+17-05-2013.pdf

Servicio de Rentas Internas. (2018). *Resolución No. NAC-DGERCGC18-00000431*.

<https://www.sri.gob.ec/o/sri-portlet-biblioteca-alfresco-internet/descargar/364f9544-5c4e-4b74-9b28-e4a208a8174a/NAC-DGERCGC18-00000431.pdf>

Servicio de Rentas Internas. (2022). *Resolución No. NAC-DGERCGC22-00000024*.

<https://www.sri.gob.ec/o/sri-portlet-biblioteca-alfresco-internet/descargar?id=c508d69a-4ea4-4940-8777-fbe89fef2fac&nombre=NAC-DGERCGC22-00000024.pdf>

Servicio de Rentas Internas. (2023). *FICHA TÉCNICA COMPROBANTES ELECTRÓNICOS*

ESQUEMA OFFLINE Versión 2.23. <https://www.sri.gob.ec/o/sri-portlet-biblioteca-alfresco-internet/descargar/c562dc7d-de50-4e1c-8e87-b3020d7d8c41/FICHA%20TE%CC%81CNICA%20COMPROBANTES%20ELECTRO%CC%81NICOS%20ESQUEMA%20OFFLINE%20Versio%CC%81n%202.23.pdf>

Siriwardhana, S. (2020, octubre 26). Tutorial de Diagrama de Despliegue | ¿Qué es un Diagrama de Despliegue? *Blog de Creately*.

<https://creately.com/blog/es/diagramas/tutorial-de-diagrama-de-despliegue/>

The PostgreSQL Global Development Group. (1996). *PostgreSQL: About*.

<https://www.postgresql.org/about/>

Trigás, M. (2012). *Metodología Scrum*. <https://openaccess.uoc.edu/handle/10609/17885>

Tutorials Point (India). (s. f.). *TypeORM - Quick Guide*. Recuperado 13 de julio de 2023, de

https://www.tutorialspoint.com/typeorm/typeorm_quick_guide.htm

Vergara Sandoval, M. P. (2017). *Problemática en la implementación de facturación electrónica en el Ecuador* [MasterThesis, Universidad Andina Simón Bolívar, Sede Ecuador].

<http://repositorio.uasb.edu.ec/handle/10644/6031>

Vinueza, S. O. (s. f.). *SOFTWARE Y TÉCNOLOGÍA*.

<https://repositorio.pucese.edu.ec/bitstream/123456789/2418/1/VINUEZA%20CELI%20SCAR%20STALIN.pdf>

Zamudio Domínguez, H. M., Galván García, H. A., Herrera Saucedo, A., & Murilo Macías, J. P.

(2023). *Plataforma colaborativa para el pair programming distribuido*.

<http://tesis.ipn.mx/xmlui/handle/123456789/31158>

Zulian, E. (2011). *Implementación de un framework para el desarrollo de aplicaciones web*

utilizando patrones de diseño y arquitectura MVC/REST [Thesis, Universidad de Belgrano. Facultad de Tecnología Informática.].

<http://repositorio.ub.edu.ar/handle/123456789/640>

Anexos