



! POSGRADOS !

MAESTRÍA EN SOFTWARE CON MENCIÓN EN DESARROLLO WEB Y MÓVIL

RPC-SO-34-NO.778-2021

OPCIÓN DE TITULACIÓN:

PROYECTO DE TITULACIÓN CON
COMPONENTES DE INVESTIGACIÓN
APLICADA Y/O DE DESARROLLO

TEMA:

ANÁLISIS, DESARROLLO Y MIGRACIÓN DE LOS
MÓDULOS DE PROVEEDORES Y CLIENTES DEL
SISTEMA CONTABLE DE LA EMPRESA ACCESCONT
PARA AGILIZAR LOS PROCESOS RELACIONADOS
AL MANEJO DE LA FACTURACIÓN PARA EL
DEPARTAMENTO FINANCIERO DE LA EMPRESA

AUTORES:

BRYAM FERNANDO CABRERA SARMIENTO
WILLIAM ARIOLFO LITUMA MOSCOSO

DIRECTORA:

MARÍA DEL PILAR MORQUECHO YUNGA

CUENCA – ECUADOR
2025

Autores:



Bryam Fernando Cabrera Sarmiento

Ingeniero Mecatrónico.

Candidato a Magíster en Software con Mención en Desarrollo Web y Móvil por la Universidad Politécnica Salesiana – Sede Cuenca.

bcabrer1@est.ups.edu.ec



William Ariolfo Lituma Moscoso

Ingeniero Mecatrónico.

Candidato a Magíster en Software con Mención en Desarrollo Web y Móvil por la Universidad Politécnica Salesiana – Sede Cuenca.

wlituma@est.ups.edu.ec

Dirigido por:



María Del Pilar Morquecho Yunga

Ingeniera en Sistemas.

Magister en Evaluación y Auditoría de Sistemas Tecnológicos.

Máster en Ingeniería Matemática y Computación.

mmorquechoy@ups.edu.ec

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

DERECHOS RESERVADOS

2025© Universidad Politécnica Salesiana.

CUENCA – ECUADOR – SUDAMÉRICA

BRYAM FERNANDO CABRERA SARMIENTO

WILLIAM ARIOLFO LITUMA MOSCOSO

Análisis, desarrollo y migración de los módulos de proveedores y clientes del sistema contable de la empresa Accescont para agilizar los procesos relacionados al manejo de la facturación para el departamento financiero de la empresa

DEDICATORIA

A mi madre Enriqueta, que con su amor, apoyo y conocimiento me ha guiado en cada paso de mi vida. Gracias a su ejemplo y dedicación, he podido combinar su experiencia profesional con la mía, aprendiendo juntos y profundizando en nuevas áreas para mejorar lo que hacemos. Este proyecto es el resultado de ese trabajo compartido, de todo lo que me has enseñado y del impulso que siempre me has dado para seguir adelante. Mamá, este logro es tanto tuyo como mío.

Bryam Cabrera

A mis padres Rosa y Ariolfo, por su apoyo y dedicación me han guiado hasta este punto, brindándome su ejemplo y amor en cada paso, son quienes han inculcado en mi la disciplina y pasión por lo que hago. Al amor de mi vida, Gissela que me ha demostrado que cuando nos proponemos algo, los límites no existen, siempre has sido esa luz que me guía y alienta, es un sueño más que se ve convertido en realidad de tu mano.

William Lituma

AGRADECIMIENTO

A mi hermano Esteban, por tus enseñanzas y apoyo incondicional; a mi papá Jorge, por tus palabras de aliento que me impulsaron en cada momento; a mi compañero de tesis William, por tu dedicación y compromiso que hicieron posible este proyecto. Agradezco de corazón a nuestra tutora, Ing. Pilar, quien siempre estuvo dispuesta a apoyarnos y a compartir sus valiosas enseñanzas y a ti, mi amor Diana, por estar siempre a mi lado, brindándome fuerza y apoyo trabajando junto a mí desde el inicio hasta el final. Sin ustedes, este logro no habría sido posible.

Bryam Cabrera

A mi hermano Adrián que supo inculcarme su disciplina, amor por su profesión y con sus lecciones de vida, ha hecho que no me rinda en ningún momento, A mi compañero de tesis Bryam, por tu dedicación y apoyo en este proyecto. A nuestra tutora quien guio los pasos del avance de esta tesis; nuevamente a ti amor, Gissela quien me impulsó con palabras de aliento de principio a fin en este camino. Este logro en mi vida es también gracias a ustedes.

William Lituma

TABLA DE CONTENIDO

Índice de Figuras.....	8
Índice de Tablas.....	9
Resumen.....	11
Abstract.....	12
1 Introducción.....	13
2 Determinación del Problema.....	14
3 Marco teórico referencial.....	16
3.1 Metodologías Ágiles y Adaptabilidad:.....	18
3.1.1 Creación del Product Backlog.....	19
3.1.2 Sesiones de Levantamiento de Requerimientos.....	19
3.1.3 Historias Técnicas.....	20
3.1.4 Sprint Planning.....	21
3.1.5 Ejecución del Sprint.....	21
3.1.6 Revisión del Sprint.....	22
3.1.7 Incremento del Producto.....	22
3.1.8 Retrospectiva del Sprint.....	22
3.1.9 Análisis de requerimientos.....	23
3.1.10 Desarrollo de sistemas usando metodologías ágiles.....	24
3.2 Marco Legal (SRI y Contabilidad en Ecuador):.....	25
3.2.1 Normativa Tributaria del SRI.....	25
3.2.2 Obligaciones Contables de las Empresas.....	26
3.2.3 Declaraciones y Pagos de Impuestos.....	26
3.2.4 Facturación Electrónica y Documentación Tributaria.....	27
3.3 Migración de Tecnologías:.....	27
3.3.1 Arquitectura monolítica en sistemas contables tradicionales.....	28
3.3.2 Evolución de los sistemas contables.....	29
3.3.3 Arquitecturas relacionadas con los sistemas contables actuales.....	29
3.3.4 Arquitectura de Microservicios en Sistemas Contables.....	31
3.3.5 Arquitectura del sistema contable.....	32

3.4	Evaluación y Pruebas:	35
3.4.1	Pruebas de Usabilidad con Usuarios Finales	35
3.4.2	Validación Legal y Fiscal de los Módulos de Facturación y Retención	35
4	Desarrollo del proyecto	36
4.1.1	Implementación del marco Scrum	36
4.1.2	Estructura del Sistema Contable	43
4.1.3	Validación del Sistema.....	59
4.1.4	Pruebas de Seguridad	60
4.1.5	Buenas Prácticas de Desarrollo:	62
5	Resultados y discusión.....	62
5.1	Sesiones de Levantamiento de información y socialización.....	62
5.2	Diseño de arquitectura.	63
5.2.1	Características de la arquitectura de microservicios.	63
5.3	Desarrollo de módulos de clientes y proveedores	64
5.3.1	Funcionalidad de los módulos de clientes y proveedores.	64
5.3.2	Módulo de Clientes.....	65
5.3.3	Módulo de Proveedores	70
5.3.4	Sistema antiguo frente al sistema actual	73
5.3.5	Mejoras en el rendimiento:	75
5.3.6	Migración de datos exitosa:	76
5.4	Evaluación de los módulos.....	77
5.4.1	Funciones de seguridad.....	77
5.5	Satisfacción de los usuarios:	78
6	Conclusiones.....	81
7	Glosario.....	82
8	Referencias	85
9	Anexos	87
9.1	Formulario de levantamiento de requisitos	87
9.2	Historias de Usuario	89
9.3	Análisis UX/UI.....	92
9.4	Desarrollo modular	94
9.5	Login.....	94
9.6	Módulo Clientes	95
9.7	Facturación electrónica	95
9.8	Módulo de Proveedores	95

9.9	Formulario de Validación	97
9.10	Satisfacción del Personal de Accescont durante las Pruebas del Sistema...	99
9.11	Repositorio	100

ÍNDICE DE FIGURAS

Figura 1 Sistema Monolítico Fuente: Autores	29
Figura 2 Sistema de microservicios Fuente: Autores	32
Figura 3 Microservicios del sistema contable. Fuente: Autores	47
Figura 4 Estructura de un microservicio. Fuente: Autores	48
Figura 5 ETL sistema contable. Fuente: Autores	55
Figura 6 docker compose. Fuente: Autores.....	58
Figura 7 Interfaz de usuario anterior vs actual. Fuente: Autores.....	60
Figura 8 Socialización y levantamiento de información. Fuente: Autores.....	63
Figura 9 Gestión de clientes. Fuente: Accescont.....	66
Figura 10 Gestión de productos o servicios. Fuente: Accescont.....	66
Figura 11 Generación de una Factura a un cliente. Fuente: Accescont	67
Figura 12 Verificación en el SRI de una Factura emitida. Fuente: Accescont	67
Figura 13 Lista de Facturas autorizadas. Fuente: Accescont.....	68
Figura 14 Comprobante de Facturación. Fuente: Accescont	69
Figura 15 Gestión de Proveedores. Fuente: Accescont.....	70
Figura 16 Registro de Factura de Compra. Fuente: Accescont	71
Figura 17 Emisión de retención desde el módulo proveedores. Fuente: Accescont	71
Figura 18 Comprobación de retención desde el SRI. Fuente: Accescont.....	72
Figura 19 Comprobante de retención generado. Fuente: Accescont	73
Figura 20 Módulo de clientes sistema anterior. Fuente: Accescont	74
Figura 21 Módulo de clientes búsqueda. Fuente: Accescont.....	74
Figura 22 Módulo de clientes lista de clientes. Fuente: Accescont	75
Figura 23 Módulo de clientes búsqueda. Fuente: Accescont.....	75
Figura 24 Códigos de retenciones sistema antiguo. Fuente: Autores.....	76
Figura 25 Códigos de retención nuevo sistema. Fuente: Autores.....	77
Figura 26 Interfaz de usuario final. Fuente: Autores.....	79

ÍNDICE DE TABLAS

Tabla 1 Historias Técnicas.....	39
Tabla 2 Pruebas de seguridad.....	61

ANÁLISIS, DESARROLLO Y
MIGRACIÓN DE LOS MÓDULOS
DE PROVEEDORES Y CLIENTES
DEL SISTEMA CONTABLE DE LA
EMPRESA ACCESCONT PARA
AGILIZAR LOS PROCESOS
RELACIONADOS AL MANEJO DE
LA FACTURACIÓN PARA EL
DEPARTAMENTO FINANCIERO
DE LA EMPRESA.

AUTOR(ES):

BRYAM FERNANDO CABRERA SARMIENTO

WILLIAM ARIOLFO LITUMA MOSCOSO

RESUMEN

Este proyecto se centra en el análisis, desarrollo y migración de los módulos de proveedores y clientes del sistema contable de la empresa Accescont para agilizar los procesos relacionados al manejo de la facturación en el departamento financiero de la empresa, para así, cumplir con la normativa tributaria ecuatoriana, mejorando la eficiencia de los procesos de facturación y de retenciones en relación con el antiguo sistema de la empresa.

Para cumplir con este objetivo se adoptó una arquitectura de microservicios debido a sus ventajas de flexibilidad, escalabilidad y rendimiento. Estos microservicios son independientes, cada uno con su propio contenedor, que se orquesta desde el docker compose, utilizando las tecnologías de MongoDB, Express.js, Node.js y Vue.js, las mismas, a largo plazo, permitirán incorporar nuevas funcionalidades en el sistema sin afectar al sistema central, ni a los servicios creados en este proyecto.

La disminución del tiempo para crear y enviar las facturas desde los microservicios actuales, comparado con el sistema monolítico anterior, gracias a la integración con el SRI, fue uno de los hitos más significativos de este proyecto, usando Node.js y Express, junto con MongoDB para el Backend y Vue para el Front.

Finalmente, para validar el funcionamiento de la aplicación se desarrollaron pruebas de usabilidad y seguridad, las primeras demostraron la mejora significativa en la experiencia del usuario y la interacción con el sistema. Mientras que, las pruebas de seguridad demostraron que la implementación de los microservicios es segura, ya que cuentan con JWT y refresco de tokens, esto permite contar con módulos eficientes, seguros y con la posibilidad de expandir nuevas funcionalidades al sistema.

Palabras clave:

Microservicio, Node.js, docker, SRI, facturación, usuario, base de datos.

ABSTRACT

This project focuses on the analysis, development and migration of the suppliers and clients modules of Accescont's accounting system to streamline the processes related to the management of invoicing in the company's financial department, in order to comply with Ecuadorian tax regulations, improving the efficiency of the invoicing and withholding processes in relation to the company's old system.

To meet this goal, a microservices architecture was adopted due to its flexibility, scalability and performance advantages. These microservices are independent, each one with its own container, which is orchestrated from the docker compose, using MongoDB, Express.js, Node.js and Vue.js technologies, which, in the long term, will allow incorporating new functionalities in the system without affecting the central system or the services created in this project.

The decrease in the time to create and send invoices from the current microservices, compared to the previous monolithic system, thanks to the integration with the SRI, was one of the most significant milestones of this project, using Node.js and Express, along with MongoDB for the Backend and Vue for the Front.

Finally, to validate the operation of the application, usability and security tests were developed, the former demonstrated a significant improvement in the user experience and interaction with the system. The security tests showed that the implementation of the microservices is secure, since they have JWT and token refreshment, which allows having efficient and secure modules with the possibility of expanding new functionalities to the system.

Keywords:

Microservice, Node.js, docker, SRI, billing, user, database.

1 INTRODUCCIÓN

En el ámbito de la gestión financiera empresarial, los sistemas contables resultan imprescindibles para el registro, análisis y reporte de la información financiera de forma precisa y organizada. Estos sistemas permiten optimizar la eficiencia operativa mediante la automatización de procesos clave. En particular, los módulos de clientes y proveedores juegan un papel crucial, ya que garantizan un flujo de trabajo ágil y preciso en las transacciones comerciales.

Este proyecto tiene como propósito analizar, desarrollar y migrar los módulos de clientes y proveedores del sistema contable de Accescont, para agilizar los procesos relacionados al manejo de la facturación para el departamento financiero de la empresa. Para ello se han definido los siguientes objetivos específicos:

1. Analizar los requerimientos para los módulos del nuevo sistema contable con la participación de los usuarios y partes interesadas.
2. Diseñar la arquitectura del sistema contable según los requerimientos establecidos.
3. Desarrollar los módulos de clientes y proveedores para el sistema contable que cumpla con los requisitos identificados, haciéndolos adaptables y escalables para un posterior crecimiento del sistema contable.
4. Evaluar el desarrollo de los módulos para garantizar su calidad y funcionalidad realizando pruebas de validación y usabilidad, que aseguren un adecuado funcionamiento acorde a los objetivos y requisitos planteados.

Para cumplir con estos objetivos, se propone el uso de metodologías ágiles de desarrollo de software, asegurando que el sistema se mantenga adaptable y escalable frente a los cambios en las normas tributarias y a las nuevas necesidades que pudiesen surgir con el paso del tiempo. La estructura modular del sistema

facilita tanto la implementación como su mantenimiento promoviendo el uso eficiente de recursos.

Con este fin, se determina que los módulos deben estar divididos en varios servicios backend (clientes, proveedores, productos y servicios, facturación y retenciones) y un frontend desarrollado en Vue, todos orquestados mediante Docker. Esta configuración modular facilita el despliegue y mantenimiento del sistema, otorgándole mayor flexibilidad y escalabilidad.

Este proyecto brinda a los usuarios del sistema la capacidad de gestionar de manera más eficiente los registros de clientes y proveedores, además de configurar parámetros personalizados para la facturación y retención. El sistema se encarga de la autorización de documentos electrónicos, ante el Servicio de Rentas Internas (SRI), mediante la estructuración de los documentos en formato XML con firma electrónica, lo que facilitará su posterior envío para la aprobación y cumplimiento regulatorio establecido para Ecuador.

La estructura del proyecto, basada en microservicios independientes pero interconectados, asegura una mayor robustez y flexibilidad ante los desafíos técnicos que puedan presentarse durante el desarrollo y despliegue. Esta metodología, respaldada por el uso de Docker, garantiza que cada componente del sistema pueda desarrollarse, probarse y desplegarse de manera eficiente, lo que contribuye significativamente al éxito integral del proyecto.

2 DETERMINACIÓN DEL PROBLEMA

La empresa Accescont, con más de 35 años de trayectoria en el mercado contable y financiero, enfrenta diversos retos debido a la obsolescencia de su sistema contable actual. Este sistema, construido sobre una arquitectura monolítica, ha demostrado ser insuficiente para atender las crecientes y cambiantes demandas tanto del negocio como de las normativas contables vigentes en el país. A continuación, se destacan los principales problemas identificados:

1. No escalable, el sistema monolítico de Accescont ha alcanzado su límite de degradación del código, lo que impide su escalabilidad para gestionar un mayor volumen de transacciones y usuarios. Esto ha resultado en tiempos de respuesta prolongados y una reducción en la eficiencia operativa. En un entorno empresarial dinámico, la capacidad de escalar rápidamente es fundamental para mantener la competitividad.
2. Actualización o modificación del sistema monolítico, actualmente implica un esfuerzo considerable en términos de tiempo y recursos, por lo que cualquier cambio en el sistema puede afectar a múltiples componentes, lo que aumenta el riesgo de errores y fallos. Esta situación dificulta la implementación de nuevas funcionalidades y la corrección de errores de manera eficiente.
3. El sistema actual presenta dificultades para integrarse con nuevas tecnologías y herramientas que podrían mejorar la eficiencia y efectividad de los procesos contables. La falta de compatibilidad y flexibilidad tecnológica limita la capacidad de Accescont para innovar y adaptarse a nuevas tendencias y normativa en el sector contable.
4. Emisión de documentos electrónicos, en el contexto de las regulaciones contables y fiscales en Ecuador, es fundamental que el sistema contable cumpla con todas las normativas y garantice la seguridad de la información financiera. El sistema actual de Accescont enfrenta desafíos para cumplir con los requisitos de emisión de documentos electrónicos exigidos por el Servicio de Rentas Internas (SRI) y otras regulaciones locales, lo que puede resultar en sanciones y pérdida de confianza por parte de los clientes.
5. El sistema monolítico de Accescont tiene limitaciones en la gestión y análisis de datos, lo que dificulta obtener información precisa y en tiempo real, esto afecta la capacidad de la empresa para realizar auditorías internas y responder a requerimientos de auditorías externas de manera oportuna.
6. La interfaz y experiencia de usuario del sistema actual es deficiente, lo que afecta la productividad de los empleados, porque su uso es lento en comparación con un sistema dinámico y actual, por ello, un sistema

moderno y fácil de usar es crucial para mejorar la eficiencia operativa y la satisfacción del cliente.

En resumen, los problemas identificados en el sistema contable actual de Accescont son: escalabilidad limitada, mantenimiento y actualización costosos, integración con nuevas tecnologías, cumplimiento normativo y seguridad, gestión de datos y análisis deficientes, y una experiencia de usuario deficiente.

Para abordar estos problemas, se propone la migración del sistema contable de Accescont a una arquitectura de microservicios. Esta nueva arquitectura permitirá mejorar la escalabilidad mediante la división del sistema en servicios independientes que pueden ser escalados de manera autónoma. Además, reducirá los costos, riesgos de mantenimiento y actualización al permitir que a cada microservicio se lo trate de manera independiente.

La nueva arquitectura facilitará la integración con tecnologías emergentes al adoptar estándares y herramientas modernas. Además, se garantizará el cumplimiento normativo y la seguridad mediante la implementación de soluciones específicas para la gestión de documentos electrónicos y la protección de datos. Así como también se optimizará la gestión y análisis de datos a través de servicios especializados en el procesamiento y análisis de información financiera. Por último, mejorará la experiencia del usuario mediante el desarrollo de interfaces modernas, eficientes y fáciles de usar.

3 MARCO TEÓRICO REFERENCIAL

Este documento presenta los fundamentos conceptuales y metodológicos indispensables para comprender y desarrollar un sistema contable basado en microservicios para la empresa Accescont. Esta sección está destinada a proporcionar una visión integral de los principios y tecnologías que orientarán la migración y el desarrollo del nuevo sistema; asegurando que se cumplan tanto los

requerimientos funcionales como no funcionales, además de las normativas legales y regulatorias específicas de Ecuador.

En primer lugar, la migración de sistemas contables tradicionales hacia una arquitectura de microservicios, surge como una respuesta eficaz a las limitaciones inherentes de los sistemas monolíticos. A lo largo de los años, los sistemas monolíticos, caracterizados por integrar todas las funcionalidades en una única aplicación, han demostrado ser ineficaces para adaptarse a los cambios tecnológicos rápidos y las demandas crecientes de escalabilidad y flexibilidad. A diferencia, la arquitectura de microservicios permite descomponer una aplicación en pequeños servicios autónomos, cada uno de los cuales ejecuta un proceso independiente y se comunica a través de interfaces ligeras, como HTTP/REST. Esto facilita el desarrollo, despliegue y escalabilidades independientes de los servicios, promoviendo la agilidad y eficiencia en el mantenimiento del software (Newman, 2015; Richardson, 2018).

Uno de los aspectos clave en el desarrollo de este sistema contable es la correcta implementación de los módulos de clientes y proveedores en función de las normativas ecuatorianas, particularmente las emitidas por el Servicio de Rentas Internas (SRI). En Ecuador, la normativa legal impone requisitos específicos para la emisión de facturas electrónicas y la gestión de retenciones, lo que exige una integración adecuada con el sistema del SRI. Este proceso, además de garantizar la conformidad legal, permite la automatización de procedimientos contables esenciales para empresas como Accescont. Para esto, el uso del formato XML se convierte en un pilar fundamental, dado que es el estándar requerido por el SRI para la transmisión de documentos electrónicos fiscales (SRI, 2024c, 2024a).

Además de las exigencias legales, el diseño del sistema debe considerar aspectos técnicos como la implementación de tecnologías modernas que soporten la arquitectura de microservicios. En este proyecto, se ha optado por utilizar tecnologías como MongoDB, Express, Vue.js y Docker. MongoDB, como base de datos NoSQL, ofrece ventajas significativas en términos de flexibilidad y escalabilidad al manejar grandes volúmenes de datos (Chodorow, 2013), lo cual es

crucial en un sistema contable que involucra numerosas transacciones y registros. Express.js, como framework para la creación de APIs (Brown, 2019), permite una integración fluida y rápida de los servicios, mientras que Vue.js facilita el desarrollo del frontend, garantizando una experiencia de usuario eficiente. Docker, por su parte, es esencial para la contenerización de los servicios (Merkel, 2014), lo que permite la ejecución y despliegue de los módulos en diferentes entornos sin dependencia del hardware subyacente. El uso de metodologías ágiles en el desarrollo del sistema también es fundamental. La naturaleza iterativa de estas metodologías, como Scrum, permite una mayor adaptabilidad y una respuesta más rápida a los cambios en los requisitos de los usuarios. Este enfoque no solo mejora la calidad del software desarrollado, sino que también garantiza una implementación que se alinea con las necesidades reales de la empresa, facilitando la escalabilidad y el crecimiento del sistema a largo plazo (Schwaber & Sutherland, 2017).

Finalmente, el marco teórico subraya la relevancia de realizar pruebas exhaustivas de validación y usabilidad, con el fin de asegurar que los módulos desarrollados cumplan con los estándares de calidad y funcionalidad esperados. Esto resulta especialmente crítico en el contexto contable, ya que los errores en el procesamiento de transacciones o en la generación de facturas pueden tener graves repercusiones tanto financieras como legales.

3.1 METODOLOGÍAS ÁGILES Y ADAPTABILIDAD:

El uso de metodologías ágiles, como Scrum, es una estrategia eficaz para mejorar la flexibilidad, adaptabilidad y eficiencia en el desarrollo de software, especialmente en proyectos complejos como la implementación o mejora de sistemas contables. En este caso, se utilizará Scrum, que permite un enfoque iterativo e incremental, facilitando la incorporación continua de cambios y la retroalimentación de los usuarios en cada sprint. Este enfoque resulta particularmente útil en el desarrollo de sistemas contables, donde los requerimientos pueden cambiar constantemente debido a modificaciones normativas, fiscales o tecnológicas (Thesing et al., 2021).

Scrum fomenta una colaboración constante entre los equipos de desarrollo y los usuarios finales, lo que asegura que las funcionalidades clave del sistema, como la gestión de cuentas y la generación de reportes financieros, se ajusten a las necesidades reales del negocio. Según (Schwaber & Sutherland, 2017), esta interacción continua permite que el equipo de desarrollo entregue valor de forma constante, adaptándose a los cambios que surgen en el camino.

En el contexto específico del proyecto de Accescont, la naturaleza dinámica de los requerimientos debido a las normativas fiscales y la necesidad de ajustes continuos en el sistema contable hace que Scrum sea un marco ideal. Su capacidad para manejar cambios frecuentes y su enfoque centrado en la entrega incremental de valor son esenciales para garantizar que el sistema contable evolucione según las necesidades del negocio y las exigencias regulatorias, promoviendo así una implementación exitosa y escalable.

A continuación, se detallan las principales etapas y prácticas de Scrum aplicadas a este proyecto.

3.1.1 CREACIÓN DEL PRODUCT BACKLOG

El Product Backlog es la lista priorizada de los requerimientos y funcionalidades que el sistema debe cumplir, organizada en historias de usuario y tareas técnicas. La creación del Product Backlog es una de las primeras etapas en el proceso de desarrollo ágil y comienza con una serie de actividades de levantamiento de requerimientos.

3.1.2 SESIONES DE LEVANTAMIENTO DE REQUERIMIENTOS

El levantamiento de requerimientos es un proceso crucial en Scrum, ya que define las necesidades del sistema a través de la colaboración con las partes interesadas, incluyendo el equipo de desarrollo, los usuarios finales y los miembros del departamento financiero de Accescont. Este proceso involucra las siguientes actividades:

- **a) Entrevistas y Reuniones:** Las entrevistas con los usuarios clave y las partes interesadas permiten identificar y entender las necesidades específicas de los módulos de clientes y proveedores. Estas reuniones deben enfocarse en comprender los flujos de trabajo actuales, los desafíos del sistema contable y los requisitos funcionales y no funcionales para el nuevo sistema. Según Schwaber & Sutherland, 2017., estas reuniones son fundamentales para garantizar que todos los miembros del equipo tengan una comprensión clara de los objetivos del proyecto.
- **b) Revisión de Informes de Auditoría y Documentación Existente:** Revisar los informes de auditoría previos y la documentación existente del sistema contable actual permite identificar fallas, ineficiencias y oportunidades de mejora. Además, ayuda a asegurar que el sistema cumpla con las normativas fiscales y contables vigentes, especialmente las relacionadas con el SRI.
- **c) Identificación de Flujos de Trabajo y Procesos Clave:** Es esencial mapear los flujos de trabajo relacionados con la gestión de proveedores y clientes, que incluyen la creación de registros, la gestión de facturación, las retenciones y la conciliación de pagos. Identificar estos procesos permite asegurar que el sistema sea intuitivo, eficiente y adaptado a las necesidades del negocio.
- **d) Documentación de Requerimientos:** Una vez recogidos todos los datos relevantes, los requerimientos se documentan en un formato que facilite la comprensión y el análisis por parte del equipo de desarrollo. Esta documentación sirve de base para las historias de usuario que conformarán el Product Backlog, como indica Schwaber, 2004.

3.1.3 HISTORIAS TÉCNICAS

Las historias técnicas son tareas específicas que abordan aspectos técnicos del desarrollo del sistema, como la integración de bases de datos, la configuración de servicios backend y el desarrollo de interfaces frontend. De estas historias técnicas se derivan los ítems del Product Backlog, que forman la base de los sprints.

3.1.4 SPRINT PLANNING

El Sprint Planning es la reunión que se realiza al inicio de cada sprint, en la que el equipo de desarrollo define qué ítems del Product Backlog se abordarán durante el sprint. Según Sutherland (2015), el objetivo de esta reunión es establecer una meta clara para el sprint y desglosar las historias de usuario en tareas técnicas que puedan completarse dentro del ciclo de desarrollo. Para el proyecto de Accescont, el Sprint Planning debe tener en cuenta las prioridades del departamento financiero, como la automatización de la facturación y la integración con las normativas tributarias.

Durante esta etapa, el Product Owner (responsable del Product Backlog) debe priorizar los ítems según su valor para el negocio, mientras que el equipo de desarrollo estima el esfuerzo requerido para implementar cada ítem. La colaboración entre el equipo de desarrollo y el Product Owner es crucial para asegurar que las tareas sean realistas y alineadas con los objetivos del proyecto.

3.1.5 EJECUCIÓN DEL SPRINT

La ejecución del sprint es el período durante el cual el equipo de desarrollo trabaja en los ítems del Product Backlog definidos en el Sprint Planning. Este proceso está centrado en la entrega continua de valor mediante la implementación de funcionalidades o mejoras del sistema. Según Molina Montero et al., (2018), durante el sprint, el equipo se enfoca en la colaboración constante, las reuniones diarias de seguimiento (Daily Scrum) y la adaptación a los cambios de los requerimientos. Este enfoque permite ajustar el curso del proyecto de manera rápida y eficiente ante cualquier desafío o cambio que surja.

Para el proyecto de Accescont, la ejecución del sprint implicará la codificación, integración y prueba de los módulos de clientes y proveedores, de forma iterativa, asegurando que cada parte del sistema sea funcional antes de pasar a la siguiente fase.

3.1.6 REVISIÓN DEL SPRINT

Al final de cada sprint, el equipo realiza una reunión de revisión (Sprint Review) para presentar los avances logrados a las partes interesadas. Durante esta reunión, se demuestran las nuevas funcionalidades implementadas, se recibe retroalimentación y se revisa si los objetivos del sprint fueron alcanzados. Según Hernández-Salazar & Beltrán (2020), esta reunión es clave para garantizar que el proyecto siga en la dirección correcta y para ajustar los requisitos según las necesidades del negocio.

En el caso de Accescont, la revisión del sprint se centraría en evaluar si los módulos de facturación, retención y registro de clientes y proveedores cumplen con los requerimientos del departamento financiero y las normativas fiscales.

3.1.7 INCREMENTO DEL PRODUCTO

El Incremento del Producto es el resultado del sprint, es decir, una versión funcional y mejorada del sistema que incluye todas las historias de usuario completadas. Este incremento debe ser evaluado no solo por su funcionalidad, sino también por su calidad, estabilidad y capacidad para cumplir con los requisitos legales y fiscales. Según Rodríguez González (2008), cada incremento debe estar listo para ser entregado o desplegado al usuario final, garantizando que el producto final cumpla con los estándares de calidad y la adaptabilidad necesaria.

Al final de cada ciclo de sprints, el sistema contable de Accescont debe estar más alineado con los objetivos de agilizar la gestión de proveedores y clientes, permitiendo una facturación eficiente y conforme a las normativas tributarias vigentes en Ecuador.

3.1.8 RETROSPECTIVA DEL SPRINT

Una vez concluido el sprint, el equipo lleva a cabo la Retrospectiva del Sprint, una reunión en la que se reflexiona sobre lo sucedido durante el ciclo de desarrollo. El objetivo principal de la retrospectiva es identificar qué funcionó bien, qué áreas se pueden mejorar y cómo optimizar los procesos para los siguientes sprints. Según

Sutherland (2015), esta reflexión continua permite que el equipo ajuste sus estrategias de trabajo, mejore la colaboración y se enfoque en entregar un valor aún mayor en los siguientes sprints.

Durante esta reunión, se abordan aspectos como la eficacia de las reuniones, las herramientas utilizadas, la calidad del código y las interacciones con los usuarios. Las lecciones aprendidas en cada sprint se implementan rápidamente en el siguiente ciclo, lo que incrementa la eficiencia y la efectividad del equipo a medida que el proyecto avanza.

3.1.9 ANÁLISIS DE REQUERIMIENTOS.

La etapa de análisis de requerimientos en el desarrollo ágil de sistemas es uno de los puntos más importantes para garantizar que el producto final cumpla adecuadamente con las necesidades funcionales y técnicas del sistema, además el sistema no solo debe alinearse con los objetivos de los usuarios finales del mismo sino también con los objetivos de la empresa, como está definido en el marco Scrum es una etapa necesaria para definir el product backlog inicial, además en cada una de las reuniones del sprint se debe definir si se debe agregar algún punto a los requisitos del sistema.

En este contexto. (Serna, 2021) señala la importancia de la correcta recolección de información, el modelado y la validación de requerimientos. Estas etapas son necesarias para comprender las necesidades de la empresa y traducirlas a objetivos técnicos precisos. También, como señala (Wong Durand, 2017), la participación activa de los usuarios desde el inicio del proyecto permite identificar de manera más efectiva las necesidades específicas del proyecto esto contribuye a detectar los requisitos críticos en etapas tempranas del desarrollo por ende mejora la calidad de los resultados finales.

A medida que, los sistemas contables evolucionan, Juca Banchón & Moreno Narváez, (2024) resaltan la relevancia de adaptarse de manera constante a los cambios legislativos y tecnológicos. Por esta razón, la participación activa de los usuarios y otras partes interesadas debe mantenerse a lo largo de todo el proyecto,

permitiendo así, que los requisitos del sistema se ajusten frente a cualquier variación en el entorno empresarial durante el desarrollo del sistema y posteriormente en los mantenimientos necesarios.

3.1.10 DESARROLLO DE SISTEMAS USANDO METODOLOGÍAS ÁGILES.

El desarrollo de software moderno, especialmente para sistemas empresariales críticos como los contables, exige enfoques metodológicos que permitan crear soluciones adaptables y escalables de manera rápida e incremental, con el fin de satisfacer las cambiantes demandas del entorno empresarial. En este contexto, las metodologías ágiles han sido diseñadas precisamente para abordar estos retos.

Según Martin (2014), los manifiestos ágiles establecen principios clave que priorizan las interacciones humanas sobre los procesos, fomentan una colaboración cercana con el cliente y buscan la simplicidad en las soluciones. Estos principios se concretan en prácticas como: el desarrollo iterativo e incremental, la entrega continua, la integración temprana y frecuente del código, y la retroalimentación constante de los usuarios. Estas prácticas, en conjunto, flexibilizan el proceso de desarrollo, permitiendo una adaptación ágil ante nuevos requerimientos.

En el marco de Scrum, se destacan roles fundamentales como: el Equipo de Desarrollo, compuesto por los programadores responsables de la construcción del producto contable; el Scrum Master, cuya principal función es facilitar el proceso y eliminar cualquier obstáculo que pueda surgir; y el Product Owner, quien actúa como representante clave del cliente, canalizando sus necesidades y priorizando los requisitos del producto.

Como se muestra en los puntos anteriores en la metodología Scrum se encuentran diferentes etapas importantes el Sprint Planning, donde se define el trabajo a realizar durante el sprint; el Daily Scrum, que permite un seguimiento continuo del progreso; y la Revisión del Sprint, en la cual se presenta el incremento funcional desarrollado al finalizar el sprint. Estos elementos constituyen pilares clave para la ejecución efectiva de Scrum.(Schwaber & Sutherland, 2017)

3.2 MARCO LEGAL (SRI Y CONTABILIDAD EN ECUADOR):

El marco legal en Ecuador desempeña un papel crucial en la gestión contable y fiscal de las empresas. La normativa vigente regula las obligaciones tributarias que deben cumplir los contribuyentes, asegurando un control adecuado de la recaudación de impuestos y el cumplimiento de las leyes. El Servicio de Rentas Internas (SRI) es el principal organismo encargado de regular y supervisar el sistema tributario, velando para que las empresas operen dentro del marco legal y contribuyan al desarrollo económico del país (SRI, 2024c).

3.2.1 NORMATIVA TRIBUTARIA DEL SRI

La Ley de Régimen Tributario Interno es el pilar del sistema impositivo en Ecuador, regulando impuestos clave como el Impuesto a la Renta y el Impuesto al Valor Agregado (IVA). Esta ley define las responsabilidades fiscales de los contribuyentes, incluyendo los plazos y procedimientos necesarios para cumplir con sus obligaciones tributarias, garantizando así el adecuado funcionamiento del sistema fiscal del país. (SRI, 2024b).

Código Tributario: El Código Tributario es la normativa que regula los derechos y obligaciones de los contribuyentes en relación con el Servicio de Rentas Internas (SRI). Este código abarca aspectos clave como la recaudación, fiscalización y las sanciones asociadas al incumplimiento de las obligaciones fiscales. Las empresas deben cumplir estrictamente con sus responsabilidades tributarias para evitar sanciones y garantizar el correcto funcionamiento de su contabilidad. (SRI, 2024a).

Resoluciones del SRI: El Servicio de Rentas Internas (SRI) emite resoluciones periódicas que inciden en el cumplimiento tributario. Estas resoluciones tienen como objetivo aclarar procedimientos o introducir nuevas normativas para la declaración y el pago de impuestos. Es esencial que las empresas se mantengan actualizadas con las últimas resoluciones, ya que estas afectan directamente la

contabilidad y los procesos fiscales, asegurando así el cumplimiento adecuado de las obligaciones tributarias (SRI, 2024a).

3.2.2 OBLIGACIONES CONTABLES DE LAS EMPRESAS

Ley de Compañías: En Ecuador, las empresas deben cumplir con las disposiciones de la Ley de Compañías, la cual exige llevar una contabilidad organizada. Los libros contables y estados financieros deben estar de acuerdo con las regulaciones establecidas, garantizando la transparencia y control de las operaciones económicas (SRI, 2024a).

Normas Internacionales de Información Financiera (NIIF): En Ecuador, las NIIF se han adoptado para regular la forma en que las empresas deben presentar sus estados financieros. Esta normativa establece estándares internacionales que garantizan una presentación clara y coherente de la información financiera. Las empresas que superan un umbral específico están obligadas a implementar dichas normas.

3.2.3 DECLARACIONES Y PAGOS DE IMPUESTOS

Impuesto a la Renta: Las empresas en Ecuador están obligadas a calcular y declarar anualmente el Impuesto a la Renta. Este proceso involucra la presentación de ingresos y deducciones permitidas para determinar la base imponible. Es esencial que las empresas cumplan con los plazos establecidos por el SRI para evitar sanciones (SRI, 2024b).

Impuesto al Valor Agregado (IVA): El IVA es un impuesto indirecto que grava el consumo. Las empresas deben declarar y pagar el IVA de manera mensual o bimestral, según su categoría de contribuyente. La correcta gestión de este impuesto es fundamental para evitar discrepancias con el SRI. (SRI, 2024b).

Retenciones en la Fuente: Las retenciones en la fuente son un mecanismo donde las empresas actúan como agentes de retención para ciertos pagos, tales como salarios, pagos a proveedores y otros servicios. Estas retenciones se aplican de acuerdo con los porcentajes establecidos por el SRI. (SRI, 2024b).

3.2.4 FACTURACIÓN ELECTRÓNICA Y DOCUMENTACIÓN TRIBUTARIA

Facturación Electrónica en Ecuador: El SRI ha implementado la facturación electrónica como un mecanismo obligatorio para muchas empresas. Esta modalidad garantiza la seguridad y eficiencia en el proceso de facturación, facilitando la supervisión por parte del SRI. Las empresas deben integrar sus sistemas de facturación para emitir estos documentos electrónicos de manera conforme con las especificaciones del SRI (SRI, 2024c).

Documentos Tributarios Electrónicos (DTE): Los DTEs incluyen diversos documentos fiscales, como facturas, notas de crédito y débito, que deben ser generados y firmados electrónicamente. La firma electrónica asegura la autenticidad e integridad de estos documentos, los cuales son esenciales para cumplir con los requerimientos tributarios del país (SRI, 2024c).

3.3 MIGRACIÓN DE TECNOLOGÍAS:

La actualización de tecnologías es fundamental para garantizar que los sistemas contables no solo sean eficientes, sino también escalables y adaptables ante las necesidades del negocio y de la normativa del país. En este contexto, se ha optado por una arquitectura de microservicios que incluye MongoDB, Express, Vue.js y Docker, cada una seleccionada por sus características específicas que promueven la flexibilidad y el despliegue continuo.

MongoDB, como una base de datos NoSQL, se destaca por su capacidad para manejar grandes volúmenes de datos no estructurados, lo que la hace ideal para gestionar el flujo masivo de transacciones y registros en un entorno contable. Además, su modelo flexible permite adaptarse rápidamente a cambios en los requisitos del sistema, sin las rigideces de las bases de datos tradicionales (Chodorow, 2013).

Por otro lado, Express.js se utiliza ampliamente como framework para crear APIs eficientes y robustas. Su simplicidad y rendimiento permiten una integración fluida

entre los diferentes microservicios, mejorando la capacidad de respuesta y reduciendo los tiempos de desarrollo (Brown, 2019). En el frontend, Vue.js ha sido elegido por su capacidad de crear interfaces de usuario interactivas y de fácil mantenimiento. Esta herramienta no solo facilita el desarrollo ágil de aplicaciones web modernas, sino que también mejora la experiencia del usuario final al ofrecer un rendimiento rápido y eficiente.

Finalmente, Docker se ha convertido en una herramienta clave para la contenedorización de los microservicios. Al aislar cada servicio en contenedores independientes, Docker facilita el despliegue continuo en múltiples entornos sin las complicaciones de compatibilidad con el hardware o las dependencias específicas de cada servicio. Esta capacidad para empaquetar y ejecutar aplicaciones de manera consistente es esencial para garantizar la estabilidad y la eficiencia en sistemas complejos como los contables (Merkel, 2014).

3.3.1 ARQUITECTURA MONOLÍTICA EN SISTEMAS CONTABLES TRADICIONALES

Los sistemas contables tradicionales se han desarrollado bajo una arquitectura monolítica (Figura 1). En este modelo, todos los componentes del software, como las bases de datos, las interfaces de usuario y la lógica de negocio, están integrados en una única aplicación. Aunque es un enfoque relativamente fácil de implementar, presenta limitaciones en términos de escalabilidad, flexibilidad y mantenimiento. Los cambios o actualizaciones requieren modificaciones en todo el sistema, lo que aumenta la probabilidad de errores y disminuye la capacidad de adaptar el sistema ante nuevos requisitos. (Sommerville, 2016)

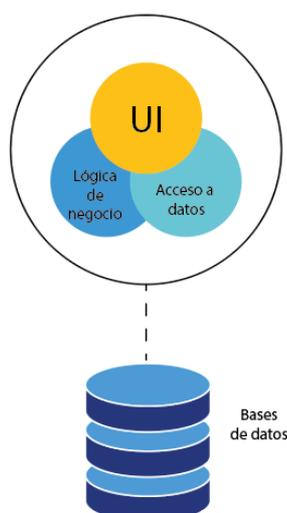


Figura 1 Sistema Monolítico Fuente: Autores

3.3.2 EVOLUCIÓN DE LOS SISTEMAS CONTABLES.

Históricamente, los sistemas contables han pasado por varias etapas, desde el uso de registros manuales hasta la adopción de tecnologías informáticas. Con la creciente complejidad de las operaciones contables de las empresas y la necesidad de manejar grandes volúmenes de datos, los sistemas contables han evolucionado hacia soluciones más eficientes, con el uso de bases de datos, y recientemente, hacia arquitecturas distribuidas como los microservicios (Gelinas et al., 2018).

3.3.3 ARQUITECTURAS RELACIONADAS CON LOS SISTEMAS CONTABLES ACTUALES.

En cuanto a los sistemas contables actuales la elección de una u otra arquitectura esta enlazada a las funcionalidades y debe ser elegida basándose en el tamaño de la empresa, la complejidad de sus procesos contables, la necesidad de escalabilidad y el nivel de integración con otros sistemas. Las arquitecturas más recomendadas para un sistema contable eficiente deben garantizar seguridad, eficiencia, escalabilidad y facilidad de mantenimiento.

Para ello una lista de las arquitecturas que pueden ser usadas en un sistema contable puede ser la siguiente.

Arquitectura en Capas: Esta es una arquitectura clásica donde el sistema se divide en capas separadas, como la capa de presentación (UI), la capa de lógica de negocio y la capa de datos. Cada capa tiene una función definida, lo que facilita el mantenimiento y la escalabilidad, por ende, es práctica para aplicaciones que manejan grandes volúmenes de datos como es el caso de un sistema contable, pero su principal desventaja es que puede volverse muy rígido si las capas se vuelven muy dependientes una de la otra. (Richards, 2015; Sommerville, 2016)

Arquitectura Orientada a Servicios (SOA): Aunque es similar a la arquitectura de microservicios su escala es mucho mayor, SOA organiza las funcionalidades del sistema en servicios que se comunican a través de protocolos estándar como SOAP o REST. En un sistema contable, cada servicio podría encargarse de una función de negocio, como el manejo de inventarios, el cálculo de impuestos o la facturación, además esta arquitectura es útil para empresas que necesitan integrar el sistema contable con otros servicios empresariales como un ERP o un CRM y buscan la escalabilidad y reusabilidad de sus servicios. Su principal desventaja es que la implementación y mantenimiento puede ser compleja y requiere de una infraestructura robusta. (Sommerville, 2016)

Arquitectura de Eventos (Event-Driven): Es un sistema impulsado por eventos, cada acción genera un evento, como la creación de una factura, la recepción de un pago o la generación de un reporte, estos eventos son procesados de manera asíncrona. Los diferentes módulos del sistema escuchan estos eventos y responden adecuadamente mediante la lógica implementada para cada evento. Esta arquitectura es ideal para sistemas en tiempo real donde se pueden procesar grandes volúmenes de datos, la desventaja es que son sistemas complejos de diseñar e implementar. (Richards, 2015)

Arquitectura de Microkernel (Plugin-Based): En esta arquitectura, el sistema contable tiene un núcleo básico que maneja las operaciones fundamentales, registro de transacciones, manejo de usuarios, etc. Permite la extensión del sistema mediante plugins o módulos adicionales que gestionan funciones específicas, como informes avanzados, ideal para sistemas contables que requieren personalización

frecuente o que operan en varias jurisdicciones con diferentes normativas fiscales.(Fowler, 2019; Richards, 2015)

Arquitectura Basada en la Nube (Cloud-Native): En esta arquitectura, el sistema contable está diseñado para ejecutarse en una infraestructura en la nube como AWS, Azure o Google Cloud, aprovechando los servicios gestionados que ofrece la nube, como bases de datos, almacenamiento, seguridad y escalabilidad automática (Theodoropoulos et al., 2023).

Arquitectura Serverless: En un sistema contable serverless, las funciones específicas se ejecutan en la nube bajo demanda, eliminando la necesidad de gestionar servidores dedicados. Esto es ideal para tareas como la generación de reportes, el envío de facturas o la validación de transacciones (Li et al., 2023).

3.3.4 ARQUITECTURA DE MICROSERVICIOS EN SISTEMAS CONTABLES.

Esta arquitectura divide el sistema en pequeños servicios independientes como se muestra en Figura 2, cada uno responsable de una funcionalidad específica, como la facturación, la gestión de proveedores y clientes, o el manejo de productos. Cada microservicio es autónomo y puede ser desplegado de manera independiente, lo que facilita la escalabilidad del sistema, permitiendo que cada microservicio pueda escalarse de forma separada del resto. Además, simplifica la integración con otros sistemas, como en el caso de un sistema contable que debe conectarse con el SRI o con servicios bancarios. No obstante, su principal desventaja radica en la necesidad de utilizar herramientas de orquestación para gestionar la comunicación eficiente entre los microservicios. (Amaral et al., 2016)

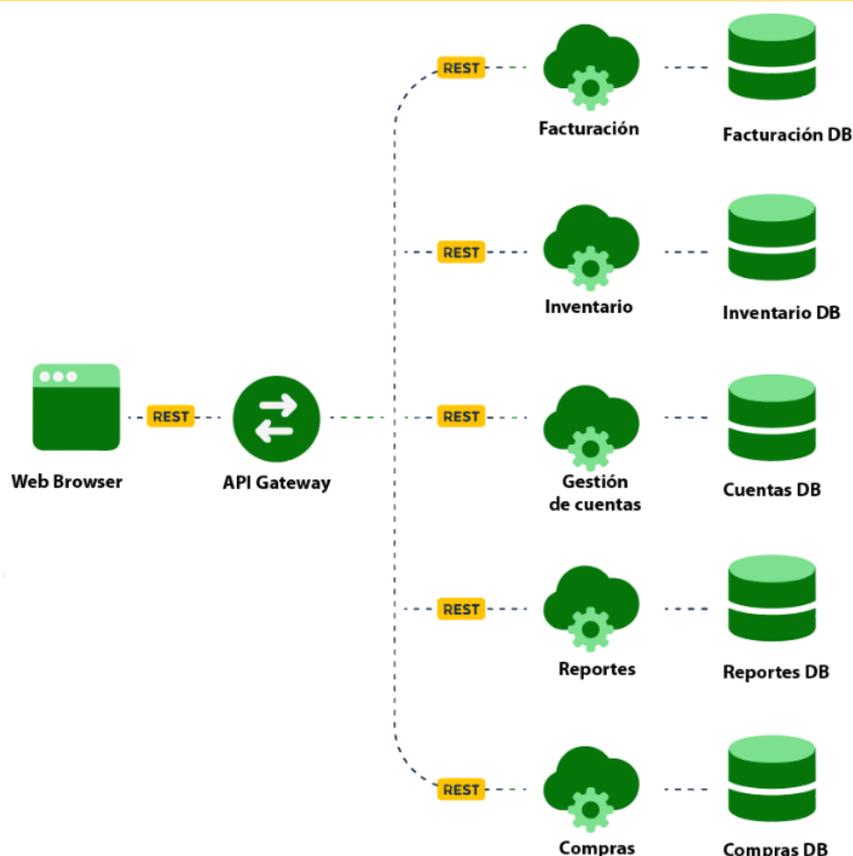


Figura 2 Sistema de microservicios Fuente: Autores

3.3.5 ARQUITECTURA DEL SISTEMA CONTABLE.

La arquitectura del sistema contable desempeña un rol fundamental en el diseño, la implementación y el mantenimiento del sistema. En el contexto de un marco de trabajo ágil como Scrum, la arquitectura debe ser flexible, escalable y modular, permitiendo una rápida adaptación a las iteraciones y cambios continuos en las necesidades del negocio y las normativas fiscales. La relación entre la arquitectura de un sistema contable y Scrum está basada en la capacidad de la arquitectura para soportar las características clave de Scrum, tales como entregas incrementales, colaboración interdisciplinaria, retroalimentación continua y adaptabilidad al cambio.

a) Flexibilidad y Adaptabilidad en el Contexto Ágil

Una de las principales características de un entorno ágil es su capacidad para adaptarse rápidamente a los cambios, ya sean de requerimientos de negocio,

prioridades del cliente o nuevas regulaciones. La arquitectura del sistema contable debe facilitar esta adaptabilidad al permitir que los equipos de desarrollo puedan iterar rápidamente sobre el sistema sin comprometer su estabilidad o funcionalidad. En este sentido, un enfoque arquitectónico basado en microservicios resulta altamente beneficioso, ya que permite descomponer el sistema en módulos independientes y autónomos que pueden ser desarrollados, desplegados y escalados de forma independiente. Esto facilita el ajuste rápido a los cambios requeridos por el negocio o por nuevas normativas fiscales, sin la necesidad de realizar modificaciones a gran escala en el sistema completo (Amaral et al., 2016).

En Scrum, cada iteración o sprint debe producir un incremento funcional del sistema. La arquitectura debe soportar esta estructura modular, permitiendo que diferentes funcionalidades (como el cálculo de impuestos o la generación de reportes) sean desarrolladas, validadas y entregadas de manera continua. Además, los equipos Scrum pueden trabajar en paralelo sobre distintos módulos del sistema, lo cual requiere que la arquitectura permita la integración continua y la desacoplabilidad de los componentes del sistema (Sommerville, 2016).

b) Escalabilidad en el Sistema Contable

Los sistemas contables deben ser capaces de manejar grandes volúmenes de datos y operaciones de forma eficiente. La escalabilidad es, por tanto, una característica esencial que la arquitectura del sistema debe garantizar. A medida que el sistema crece y se integra con otros servicios (como plataformas fiscales o servicios de terceros), la arquitectura debe ser capaz de soportar aumentos en la carga de trabajo sin sacrificar el rendimiento.

Desde la perspectiva de Scrum, la escalabilidad también está relacionada con la entrega incremental de valor. Las arquitecturas orientadas a microservicios y nativas en la nube son altamente escalables, ya que permiten que los componentes del sistema se ajusten dinámicamente según la demanda. Por ejemplo, la carga de trabajo asociada con la generación de informes fiscales puede aumentarse durante períodos fiscales, mientras que el procesamiento de transacciones se puede escalar

de forma independiente. Esta capacidad de escalar componentes individuales del sistema es consistente con los principios ágiles de entrega de valor continuo y adaptabilidad (Fowler, 2019; Theodoropoulos et al., 2023).

c) Integración con Otras Aplicaciones y Servicios

Los sistemas contables modernos deben interactuar con diversas aplicaciones externas, como sistemas de gestión empresarial (ERP), plataformas de pago y servicios fiscales, como el Servicio de Rentas Internas (SRI) en Ecuador. Esto requiere una arquitectura que facilite la interoperabilidad y la integración continua.

La arquitectura de microservicios permite que cada componente del sistema contable se conecte con otros servicios de manera eficiente, utilizando protocolos estándar como REST o SOAP para la comunicación entre servicios. La adopción de arquitecturas basadas en la nube también facilita la integración con servicios externos, utilizando plataformas de integración como API gateways o middleware. En el contexto de Scrum, esta integración debe ser continua y permitir la retroalimentación rápida de los usuarios o sistemas externos para realizar ajustes y mejoras en cada iteración.

d) Mantenimiento y Evolución del Sistema

La mantenibilidad y la capacidad de evolución de un sistema contable son esenciales para garantizar que el sistema siga siendo relevante y útil a medida que cambian las normativas fiscales y las necesidades del negocio. Scrum enfatiza la importancia de las iteraciones cortas y la retroalimentación constante para mejorar el sistema. La arquitectura debe ser lo suficientemente modular y desacoplada como para permitir que las actualizaciones o mejoras se realicen sin interrumpir el funcionamiento de otros componentes del sistema.

En este sentido, las arquitecturas en capas o orientadas a servicios (SOA) permiten una separación clara entre las diferentes capas del sistema, como la presentación, la lógica de negocio y los datos. Esta separación facilita el mantenimiento y las actualizaciones sin afectar al sistema en su conjunto. Además, la adopción de

patrones como event-driven architecture (EDA) permite una actualización o adición de funcionalidades de manera más flexible, al basar el flujo de datos en eventos y no en procesos secuenciales rígidos (Richards, 2015).

3.4 EVALUACIÓN Y PRUEBAS:

La fase de evaluación y pruebas es crucial en el desarrollo de cualquier sistema contable, especialmente durante la migración de módulos como facturación y retención. El éxito de un sistema contable moderno depende, en gran medida, de validar que todos los componentes cumplan con los requisitos funcionales, legales y fiscales establecidos, además de proporcionar una experiencia de usuario eficiente y libre de errores. En esta sección, se analizarán tanto las pruebas de usabilidad con los usuarios finales, como la validación legal y fiscal de los módulos de facturación y retención.

3.4.1 PRUEBAS DE USABILIDAD CON USUARIOS FINALES

Las pruebas de usabilidad tienen como objetivo principal validar que los módulos desarrollados o migrados no solo sean funcionales, sino que también ofrezcan una experiencia de usuario satisfactoria. Estas pruebas se realizan involucrando a los usuarios finales que utilizarán el sistema en un entorno real. La usabilidad es un aspecto crucial en los sistemas contables, ya que su naturaleza implica que los usuarios interactúen con el sistema de manera continua y repetitiva, lo que aumenta la importancia de la eficiencia y la facilidad de uso. (Sohaib & Khan, 2010)

3.4.2 VALIDACIÓN LEGAL Y FISCAL DE LOS MÓDULOS DE FACTURACIÓN Y RETENCIÓN

La validación legal y fiscal es un aspecto fundamental en la implementación de los módulos de facturación y retención en un sistema contable. La correcta emisión de facturas y la aplicación de retenciones no solo deben ajustarse a los requerimientos funcionales del sistema, sino también cumplir con las normativas legales y fiscales del país donde se operen, como por ejemplo en el caso de las retenciones que tienen diferentes rangos, de acuerdo con los artículos o servicios adquiridos.

En el caso de Ecuador, los módulos de facturación y retención deben adherirse a los estándares establecidos por el Servicio de Rentas Internas (SRI). Esto implica la correcta emisión de facturas electrónicas y la aplicación precisa de las retenciones conforme a las normativas fiscales vigentes. Estos módulos deben contar con mecanismos de validación fiscal que aseguren que los cálculos de impuestos y retenciones se realicen con exactitud y en concordancia con la legislación tributaria aplicable (SRI, 2024a).

Conformidad con las Normas Tributarias: Asegurar que los módulos respeten las regulaciones fiscales, aplicando correctamente los porcentajes de retención e impuestos (SRI, 2024a).

Interoperabilidad con el SRI: Verificar que los documentos electrónicos generados, como facturas y comprobantes de retención, cumplan con los formatos exigidos por el SRI, permitiendo su envío y validación electrónica (SRI, 2024a).

Auditoría y Trazabilidad: Los sistemas deben incluir mecanismos de auditoría que aseguren la trazabilidad de las transacciones fiscales, facilitando el cumplimiento ante eventuales inspecciones o revisiones legales (SRI, 2024a).

4 DESARROLLO DEL PROYECTO

El desarrollo del proyecto tuvo como objetivo general analizar, desarrollar y migrar los módulos de clientes y proveedores del sistema contable de la empresa Accescont en la ciudad de Cuenca.

4.1.1 IMPLEMENTACIÓN DEL MARCO SCRUM

En esta sección se presenta el proceso realizado para el levantamiento de requerimientos fundamentado en lo descrito en el capítulo anterior, y al marco Scrum. Así como a las historias de usuario e historias técnicas obtenidas a partir del Formulario de levantamiento de requisitos que se encuentra en los anexos.

4.1.1.1 Sesiones de Levantamiento de Requerimientos

Estas sesiones fueron diseñadas para recopilar información clave y garantizar que los módulos desarrollados cumplan con las expectativas y necesidades reales de Accescont. Las actividades realizadas fueron:

a) Entrevistas y Reuniones

- **Objetivo:** Entender las necesidades específicas del personal contable y administrativo en relación con los módulos de clientes y proveedores.
- **Participantes:** Personal contable, administrativo, y los desarrolladores encargados del proyecto.
- **Actividades realizadas:**
 1. Entrevistas individuales y grupales para identificar problemas actuales, expectativas y requisitos específicos.
 2. Reuniones de seguimiento para aclarar dudas y confirmar los requisitos identificados.
 3. Documentación de las conclusiones y acuerdos alcanzados durante las sesiones.

b) Revisión de Informes de Auditoría y Documentación Existente

- **Objetivo:** Obtener una visión clara de las áreas que necesitan mejora y asegurar la conformidad con las normativas internas y externas.
- **Actividades realizadas:**
 1. Análisis de informes de auditoría previos para identificar puntos débiles y áreas de mejora en los procesos actuales.
 2. Revisión de manuales y procedimientos internos de Accescont para asegurar que los módulos cumplan con los estándares establecidos.
 3. Documentación de observaciones relevantes que ayuden en el diseño de las soluciones.

c) Identificación de Flujos de Trabajo y Procesos Clave

Mapear los procesos actuales para encontrar oportunidades de automatización y mejora en los módulos.

- **Actividades realizadas:**

1. Realización de talleres para visualizar y discutir los flujos de trabajo actuales, de los cuales se obtuvo las historias de usuario para cada proceso relacionado con los módulos de clientes y proveedores.
2. Identificación de procesos manuales que podrían ser automatizados para aumentar la eficiencia.
3. Creación de diagramas de flujo que detallan los pasos clave en el manejo de clientes y proveedores.
4. Identificación de puntos críticos que deben ser abordados para optimizar el flujo de trabajo.

d) Documentación de Requerimientos

Se elabora un documento detallado de requerimientos que incluye:

- **Requerimientos funcionales:** Gestión de clientes y proveedores, emisión de documentos electrónicos, integración con el SRI, firma y verificación electrónica de documentos.
- **Requerimientos no funcionales:** Escalabilidad, alta disponibilidad, seguridad en la transmisión y almacenamiento de datos.
- **Requerimientos legales y normativos:** Cumplimiento de regulaciones nacionales, generación de reportes financieros y tributarios, auditorías internas y externas periódicas.

4.1.1.2 Historias técnicas

A partir de las historias de usuario obtenidas de las reuniones con el personal de la empresa se desarrollan las historias técnicas con el objetivo de crear el producto Backlog de los módulos del sistema.

Para el caso de los módulos de clientes y proveedores las historias técnicas levantadas se muestran en la Tabla 1:

Tabla 1 Historias Técnicas

Usuario	Actividades como desarrollador	Actividades como desarrollador Frontend	Actividades como desarrollar Backend
Contador	Diseñar y configurar la estructura de la base de datos para almacenar información detallada de los proveedores, asegurando la eficiencia en la búsqueda y clasificación por tipo de proveedor.	Crear una interfaz de usuario intuitiva y funcional para permitir la edición de datos de los proveedores, garantizando la precisión y consistencia de la información.	Implementar lógica y validaciones para la eliminación segura de proveedores inactivos, evitando la eliminación accidental y manteniendo la integridad de la base de datos.
Administrador	Crear un módulo para cargar y almacenar documentos adjuntos a las facturas, asegurando la integración adecuada con el sistema de gestión de facturas y la seguridad de los archivos.	Desarrollar una interfaz intuitiva para la visualización de facturas ingresadas y próximas a caducar, garantizado una vista rápida y manejo de los documentos	Configurar un sistema de notificaciones automatizadas para alertar sobre facturas próximas a su vencimiento, utilizando herramientas de programación asíncrona para garantizar el envío oportuno de notificaciones.
Administrador	Integrar el sistema con la API del Servicio de Rentas Internas (SRI) para validar		Implementar un sistema de registro y auditoría de acciones realizadas en el proceso de emisión

	automáticamente las retenciones emitidas y cumplir con las regulaciones fiscales.		de retenciones, garantizando la trazabilidad y el cumplimiento normativo
Contador	Diseñar y configurar la base de datos para almacenar información detallada de los clientes, asegurando eficiencia en búsqueda y actualización.	Crear una interfaz intuitiva para registrar, editar y eliminar clientes, garantizando una experiencia de usuario fluida.	Implementar validaciones automáticas del formato de los documentos de identificación de los clientes y otros datos fiscales.
Administrador	Configurar un proceso automatizado para enviar documentos tributarios al SRI, asegurando la precisión y la puntualidad en el envío de información.	Desarrollar una interfaz detallada de los movimientos tributarios, para poder visualizar todo el flujo de los cambios de las facturas emitidas por Accescont.	Configurar un historial detallado de pagos realizados a cada proveedor, utilizando técnicas de almacenamiento de datos eficientes para mantener un registro preciso y accesible de las transacciones.
Administrador	Crear y configurar validaciones automáticas que aseguren el cumplimiento de las normativas fiscales al emitir retenciones, utilizando reglas y verificaciones que garanticen la exactitud de las retenciones.		Desarrollar un módulo que genere el documento de retención en un formato válido para ser enviado a autorización del Servicio de Rentas Internas (SRI), asegurando la precisión y el cumplimiento de los requisitos establecidos por la autoridad tributaria
Contador	Configurar la generación de documentos incluyendo los impuestos tributarios para garantizar la	Diseñar una interfaz que permita agregar todos los detalles pertenecientes a una factura para enviarlas y posterior	Configurar la integración con la API del SRI, para el envío y validación de las facturas electrónicas

autorización del a la autorización
documento cuando poder visualizarlas.
sea enviado al SRI.

De estas historias técnicas se obtiene el product backlog.

- Diseño y configuración de la base de datos para proveedores.
- Desarrollo de una interfaz de usuario para la edición de datos de proveedores.
- Implementación de lógica y validaciones para la eliminación segura de proveedores inactivos.
- Creación de un módulo para el almacenamiento de documentos adjuntos a las facturas.
- Integración con la API del Servicio de Rentas Internas (SRI) para la validación de retenciones.
- Creación y configuración de validaciones automáticas para garantizar el cumplimiento normativo en las retenciones.
- Configuración para la generación de reportes contables específicos.
- Integración de registros de activos fijos en los asientos contables.
- Diseño y configuración de la base de datos para clientes.
- Desarrollo de una interfaz para registrar, editar y eliminar clientes.
- Implementación de validaciones automáticas para RUC y cédula.
- Configuración de bloqueos para evitar duplicados de clientes.
- Diseño de la base de datos para facturas (detalles de cliente, productos, impuestos, etc.).
- Creación de una interfaz para registrar facturas.
- Integración con la API del SRI para el envío de facturas.
- Desarrollo de reportes de facturas emitidas.

4.1.1.3 *Sprint Planning*

Luego de obtener el product backlog se genera la planificación del desarrollo por sprints, cada uno con entregables diferentes.

Sprint 1:

- Diseño y configuración de la base de datos de proveedores utilizando MongoDB.
- Desarrollo del microservicio para la gestión de proveedores (CRUD), que permita crear, leer, actualizar y eliminar proveedores.
- Dockerización del microservicio de proveedores para asegurar su portabilidad y facilitar el despliegue en diferentes entornos.

Sprint 2:

- Creación del microservicio para almacenar documentos adjuntos a las facturas, permitiendo gestionar la documentación relacionada.
- Integración de Docker para contenerizar el servicio de almacenamiento de documentos, asegurando su fácil manejo y escalabilidad.
- Integración con la API del SRI para validar retenciones de acuerdo con la normativa fiscal vigente.

Sprint 3:

- Implementación del microservicio de emisión de retenciones, gestionando automáticamente las retenciones de impuestos.
- Dockerización del microservicio de retenciones para garantizar su despliegue independiente.
- Configuración de procesos automatizados para generar y enviar reportes fiscales al SRI, facilitando el cumplimiento normativo.

Sprint 4:

- Diseño y configuración de la base de datos para clientes con MongoDB, asegurando que los datos sean accesibles y gestionables eficientemente.
- Desarrollo del microservicio de clientes para registrar, editar y validar los datos de los clientes.
- Implementación de la validación automática del RUC y cédula mediante microservicios, con su correspondiente dockerización para facilitar la gestión y despliegue.

Sprint 5:

- Diseño de la base de datos para facturas, estructurando correctamente los detalles del cliente, productos, impuestos y otros elementos necesarios.
- Desarrollo del microservicio para el registro y envío de facturas, permitiendo que las facturas se procesen y envíen de manera eficiente.
- Dockerización e integración con la API del SRI para la gestión de facturación electrónica, asegurando el cumplimiento fiscal.

Sprint 6:

- Desarrollo de un microservicio para generar reportes de facturas emitidas, facilitando el análisis y seguimiento de las transacciones.
- Optimización del módulo de facturas, incluyendo la gestión de descuentos e impuestos individualizados por producto, mejorando la precisión de las facturas.
- Dockerización del microservicio de facturación y ajustes finales para integrar de forma eficiente con los otros servicios del sistema.

4.1.2 ESTRUCTURA DEL SISTEMA CONTABLE

El sistema contable desarrollado para Accescont sigue una estructura con módulos que permite la separación clara de los procesos de clientes y proveedores. Este diseño modular es fundamental para garantizar la escalabilidad y adaptabilidad del

sistema, permitiendo que cada módulo funcione de manera independiente y pueda ser modificado o actualizado sin afectar los demás componentes. La segmentación del sistema en módulos se alinea con la implementación de una arquitectura de microservicios, que permite desarrollar, desplegar y mantener los servicios de manera autónoma y eficiente.

4.1.2.1 *Desarrollo Modular de Clientes y Proveedores*

Clientes: El módulo de clientes está diseñado para gestionar toda la información relacionada con los usuarios que generan facturas y consumen los servicios de la empresa. Este módulo permite registrar, modificar y consultar datos de los clientes de manera dinámica. La integración con MongoDB facilita la gestión de grandes volúmenes de datos y ofrece flexibilidad para adaptarse a futuros requerimientos de expansión o cambios en el modelo de datos.

Proveedores: El módulo de proveedores está encargado de gestionar las relaciones de la empresa con sus suministradores de bienes y servicios. Al igual que el módulo de clientes, MongoDB se utiliza para almacenar información de los proveedores, asegurando la integridad y disponibilidad de los datos en tiempo real.

4.1.2.2 *Integración de XML para la Facturación Electrónica*

La facturación electrónica en Ecuador es obligatoria para las empresas según la normativa del SRI. Para cumplir con este requisito, el sistema utiliza archivos en formato **XML** para generar y transmitir los documentos electrónicos (facturas, notas de crédito y débito). Este proceso es automatizado por el sistema y sigue los lineamientos establecidos por el SRI, que exige la firma digital de cada documento electrónico.

La implementación de la firma electrónica en el formato XAdES se realiza a través del uso de bibliotecas en Java y Python, donde los archivos XML generados son firmados utilizando un certificado digital autorizado. Posteriormente, los archivos firmados son enviados al SRI mediante un protocolo **SOAP**. Este enfoque asegura la autenticidad, integridad y no repudio de las transacciones, cumpliendo con los requisitos fiscales del país.

4.1.2.3 Tecnologías Utilizadas

MongoDB: MongoDB es una base de datos NoSQL utilizada para el almacenamiento de los datos de clientes y proveedores. Su capacidad para manejar documentos flexibles en formato JSON es clave para la implementación de un sistema modular, donde los datos de cada módulo pueden crecer y adaptarse sin restricciones.

Express.js: es el framework que facilita la creación de las APIs para los módulos del sistema contable. Proporciona una capa de abstracción eficiente para la comunicación entre los módulos y otros servicios, permitiendo que las operaciones de clientes y proveedores se realicen de manera eficiente.

Vue.js: se encarga del desarrollo de la interfaz de usuario. Este framework facilita la creación de componentes reutilizables y dinámicos, lo que mejora la experiencia del usuario al interactuar con el sistema contable. Además, Vue.js permite una integración fluida con otras tecnologías como BootstrapVue para asegurar una interfaz visualmente atractiva y funcional.

Docker: es utilizado para la contenerización de los microservicios, permitiendo un despliegue ágil y uniforme en cualquier entorno. Cada módulo se ejecuta en su propio contenedor, lo que garantiza que los servicios puedan ser escalados o modificados de manera independiente sin interferir con el funcionamiento de otros componentes del sistema.

BootstrapVue: es una extensión de Bootstrap para Vue.js, es utilizado para la construcción de la interfaz visual del sistema contable. A través de su uso, se asegura que la interfaz sea responsiva y fácil de usar, adaptándose tanto a usuarios finales como a administradores del sistema.

Tecnologías adicionales:

- **Axios:** Utilizado para realizar solicitudes HTTP hacia los servicios backend, facilitando la comunicación entre el frontend y el backend.
- **Bootstrap y BootstrapVue:** Bootstrap proporciona la base para estilos consistentes, mientras que BootstrapVue permite usar estos estilos de manera

nativa dentro de componentes Vue.js, asegurando una interfaz atractiva y funcional.

- **Moment.js:** Herramienta para manejar fechas y tiempos, permitiendo formatos y cálculos de tiempo de manera sencilla en la aplicación.
- **SweetAlert2:** Biblioteca para mostrar alertas y notificaciones personalizadas, mejorando la experiencia del usuario al proporcionar mensajes visuales atractivos y claros.
- **JWT Decode:** Utilizado para decodificar tokens JWT, que es útil para manejar autenticación y control de acceso en la aplicación.
- **v-select2-component:** Componente de Vue.js para mejorar la experiencia del usuario en la selección de opciones en formularios, proporcionando una interfaz de selección más intuitiva.
- **Vue Router:** Librería que permite la navegación entre diferentes vistas de la aplicación, gestionando el enrutamiento en el frontend.
- **Vuex:** Gestor de estado centralizado para Vue.js, utilizado para manejar el estado compartido entre los componentes de la aplicación.
- **XLSX:** Permite la generación y manipulación de archivos Excel, útil para exportar e importar datos en la aplicación.
- **Zxcvbn:** Biblioteca para medir la fortaleza de contraseñas, mejorando la seguridad al alertar al usuario cuando la contraseña es débil.

4.1.2.4 *Diseño de la Arquitectura del Sistema*

Diseño de la Arquitectura Basada en Microservicios

Se diseña la arquitectura del sistema para cumplir con los requerimientos establecidos, el desarrollo de los módulos de clientes y proveedores. Esta arquitectura (Figura 3) se basó en microservicios y consideró los siguientes aspectos:

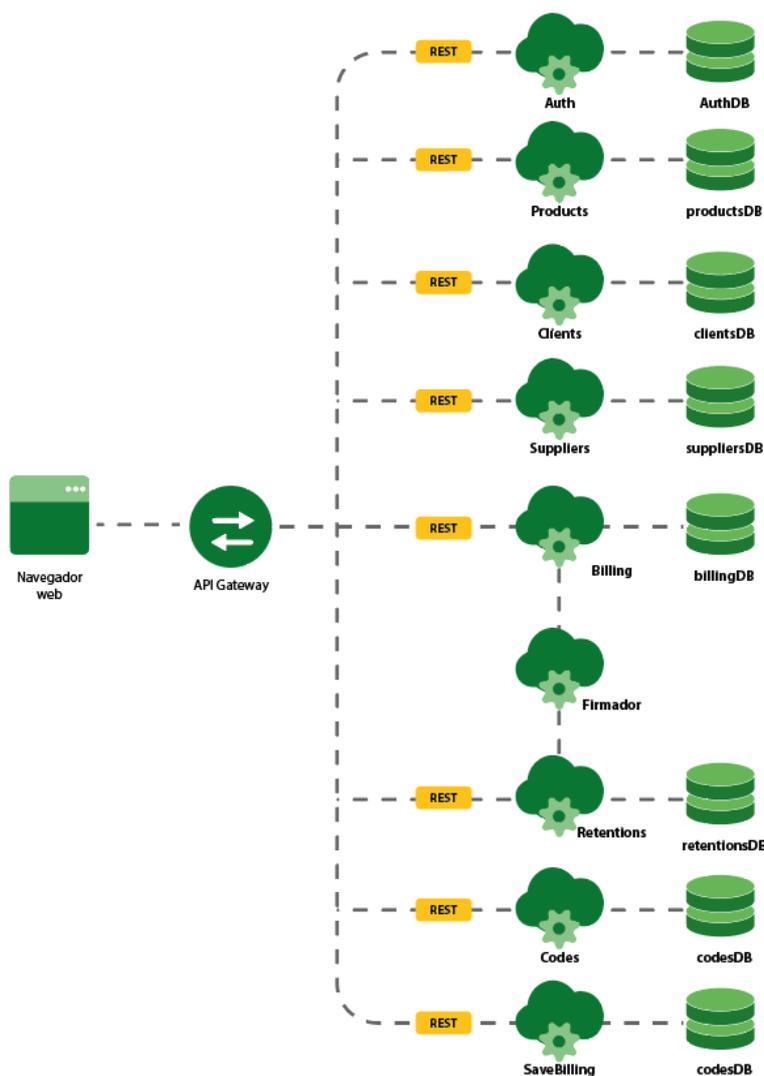


Figura 3 Microservicios del sistema contable. Fuente: Autores

- **Descomposición del Sistema** en microservicios independientes:
 - Servicio de Autenticación y Autorización
 - Servicio de Gestión de Clientes
 - Servicio de Gestión de Productos
 - Servicio de Emisión de Facturas
 - Servicio de Emisión de Retenciones
 - Servicio de Gestión de Proveedores
 - Servicio para códigos de impuestos
 - Servicio para guardar Facturas recibidas.

- **Diseño de la Base de Datos:**

Al poseer una arquitectura de microservicios como en se muestra en la Figura 3, cada microservicio posee una base de datos propia e independiente con su estructura, excepto el microservicio firmador, que es únicamente un microservicio usado para firmar los documentos electrónicos de facturas y retenciones, por ende los dos microservicios mencionados se comunican con el firmador para que antes de ser enviado al SRI el documento sea firmado.

- **Estructura de los Microservicios.**

Cada microservicio es similar en cuanto a estructura cómo se puede observar en la Figura 4, todo está contenido dentro de la carpeta src, y se divide el microservicio en las configuraciones, controlador, modelo, rutas, utilidades, aplicación principal, variables de entorno de cada microservicio y como ya se había indicado anteriormente contiene un archivo Dockerfile para la orquestación de los contenedores.

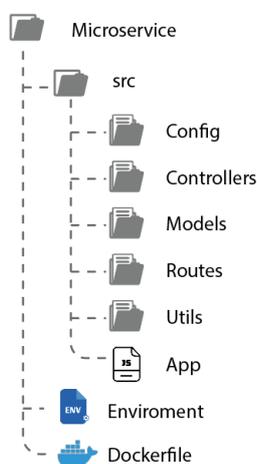


Figura 4 Estructura de un microservicio. Fuente: Autores

4.1.2.5 Dockerización

La Dockerización implica empaquetar una aplicación junto con todas las bibliotecas, dependencias, configuraciones y archivos necesarios para que esta funcione de manera autónoma en cualquier sistema operativo. El resultado de este empaquetamiento es un contenedor Docker, que es ligero y contiene todo lo

necesario para ejecutar la aplicación, eliminando problemas de configuración de entorno entre distintos sistemas.

4.1.2.6 *Ventajas de la dockerización*

- **Portabilidad:** Los contenedores Docker pueden ejecutarse en cualquier máquina que tenga Docker instalado, sin importar el sistema operativo. Esto permite mover las aplicaciones de un entorno a otro (por ejemplo, de desarrollo a producción) sin preocuparse por problemas de compatibilidad, lo que facilita la migración y el despliegue.
- **Aislamiento:** Cada contenedor funciona en un entorno completamente aislado, sin interferir con otros contenedores ni con el sistema operativo principal. Esto significa que se pueden ejecutar múltiples contenedores en una misma máquina sin preocuparse por conflictos de dependencias o configuraciones.
- **Escalabilidad:** Docker simplifica la escalabilidad horizontal, permitiendo crear rápidamente múltiples instancias de una aplicación. Esto es especialmente útil en entornos de producción o cuando se necesita manejar un alto volumen de tráfico de manera eficiente.
- **Rapidez en el despliegue:** La creación de contenedores es rápida y ligera en comparación con las máquinas virtuales, lo que agiliza significativamente los ciclos de desarrollo y despliegue, permitiendo iteraciones más veloces y eficientes.

4.1.2.7 *Arquitectura de microservicios dockerizada*

El sistema está diseñado bajo una arquitectura de microservicios, donde cada componente funcional está desacoplado y opera de manera independiente. Esto facilita el despliegue, mantenimiento y escalabilidad de cada servicio. Los microservicios clave son los siguientes:

1. Auth (Autenticación)

Es un componente crítico en una arquitectura de microservicios, responsable de gestionar la autenticación y autorización de los usuarios. Este servicio asegura que las solicitudes a otros microservicios estén protegidas y que los usuarios accedan solo a los recursos que les están permitidos.

El microservicio Auth cumple con las siguientes funciones clave:

Registro de usuarios: Permite que nuevos usuarios se registren en el sistema proporcionando datos como: nombre, correo electrónico y contraseña. Normalmente, las contraseñas son almacenadas de manera segura mediante técnicas de hashing.

Inicio de sesión (Login): Los usuarios pueden iniciar sesión proporcionando sus credenciales (normalmente, email y contraseña). Si las credenciales son correctas, el microservicio genera un token JWT (JSON Web Token) que el cliente utilizará en todas las solicitudes subsiguientes.

Autorización: Valida si un usuario tiene los permisos necesarios para acceder a recursos específicos. Cada vez que un usuario realiza una solicitud a otro microservicio, debe incluir el token JWT en la cabecera de la solicitud para que se pueda verificar su autenticidad.

Refresco de tokens: Algunos sistemas implementan la funcionalidad de "refresh token", que permite a los usuarios obtener nuevos tokens de autenticación sin tener que volver a iniciar sesión, extendiendo la sesión activa del usuario.

Gestión de sesiones y cierre de sesión: Aunque los tokens JWT no requieren cerrar sesión explícitamente, algunos sistemas permiten a los usuarios invalidar sus tokens de forma manual (cierre de sesión), o bien establecer una caducidad en los tokens que exige a los usuarios reautenticarse pasado cierto tiempo.

2. Billing (Facturación)

El microservicio de facturación es responsable de generar las facturas electrónicas. Las funcionalidades principales incluyen la creación del documento XML que cumple con los estándares del SRI, la asignación de detalles del cliente y los productos involucrados, y el cálculo de impuestos y totales. Este microservicio también coordina el envío de la factura para ser firmada por el microservicio firmador antes de su transmisión al SRI.

3. Retentions (Retenciones)

El microservicio de retenciones es responsable de generar las retenciones electrónicas. Las funcionalidades principales incluyen la creación del documento XML que cumple con los estándares del SRI en respuesta a alguna factura emitida por un proveedor, la asignación de detalles del proveedor y los tipos de retención aplicados, y el cálculo de impuestos y totales. Este microservicio también coordina el envío de la factura para ser firmada por el microservicio firmador antes de su transmisión al SRI.

4. Clients (Clientes)

El microservicio de clientes administra toda la información relacionada con los clientes de la empresa. Provee operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre los clientes, que luego son consumidas por el servicio de facturación al momento de generar una nueva factura. Cada cliente tiene información relevante como razón social, número de identificación, dirección y si está obligado a llevar contabilidad.

5. Suppliers (Proveedores)

Similar al microservicio de clientes, el microservicio de proveedores gestiona la información de los proveedores de la empresa. Esta información es crucial para generar facturas de compra o registrar proveedores en las transacciones de la

empresa. Se implementa con una estructura CRUD que interactúa con otros servicios.

6. Products (Productos)

Este microservicio gestiona la información de los productos y servicios que la empresa ofrece. El servicio almacena detalles como el nombre del producto, código, precio y stock. La facturación requiere acceder a esta información para incluir los productos o servicios vendidos en cada transacción.

7. Firmador (Firma Digital)

Este microservicio es esencial para el cumplimiento legal en Ecuador, ya que todas las facturas deben estar firmadas digitalmente antes de ser enviadas al SRI. El microservicio firmador utiliza un archivo JAR de Java que toma los documentos XML generados por el servicio de facturación, aplica la firma digital y retorna el archivo firmado listo para ser enviado.

8. API Gateway

La API Gateway actúa como la puerta de entrada única para todas las solicitudes del sistema. Este componente orquesta la comunicación entre los microservicios, permitiendo que el cliente interactúe con el sistema de forma eficiente y segura. Además, maneja la autenticación y autorización de las solicitudes utilizando un sistema de tokens.

9. MongoDB (Base de Datos)

MongoDB es la base de datos utilizada por todos los microservicios para almacenar la información necesaria. Su estructura flexible y escalable permite gestionar grandes volúmenes de datos, manteniendo la eficiencia en las operaciones CRUD que requieren los servicios de clients, suppliers, products y billing. Cada documento almacenado en MongoDB representa entidades como clientes, proveedores, productos o facturas.

4.1.2.8 Integración con el SRI

El sistema se integra directamente con el SRI para enviar las facturas electrónicas.

Este proceso sigue el siguiente flujo:

1. El servicio de facturación genera el documento XML.
2. El documento XML es firmado por el servicio de firmador utilizando la firma digital de la empresa.
3. Una vez firmado, el archivo se envía al SRI utilizando la función SOAP del microservicio de facturación.
4. El SRI responde con la aceptación o rechazo del documento, lo cual se registra en el sistema.

Este enfoque modular y basado en microservicios facilita el mantenimiento y la evolución del sistema, permitiendo integrar nuevas funcionalidades o actualizar componentes sin afectar el funcionamiento del sistema en su conjunto.

4.1.2.9 Despliegue y Escalabilidad

El sistema está diseñado para ser desplegado en contenedores Docker, lo que facilita su implementación en entornos tanto locales como en la nube. Cada microservicio está encapsulado en su propio contenedor, lo que permite escalarlos de manera independiente según la demanda.

El uso de Docker y Docker Compose simplifica la configuración del entorno y asegura que todos los microservicios puedan interactuar correctamente, garantizando un despliegue continuo y automatizado. La base de datos MongoDB también está contenida en Docker, lo que permite la gestión centralizada de los datos sin necesidad de configuraciones complejas.

4.1.2.10 Migración de Datos

La migración de datos es, sin duda, un proceso fundamental en la implementación de un sistema contable, ya que implica trasladar información valiosa desde el sistema anterior hacia la nueva plataforma. En un entorno basado en microservicios, este proceso debe realizarse de manera sumamente eficiente,

garantizando en todo momento la integridad, la precisión, y la coherencia de los datos migrados. Para asegurar el éxito de la migración, es necesario seguir varias etapas clave:

1. Extracción de Datos desde la Base SQL

El primer paso en este proceso es la extracción de datos de la base de datos relacional (SQL) y convertirlas a un formato admisible por la base de datos MongoDB que es un formato JSON.

2. Transformación de los Datos

La transformación es la etapa más compleja en la migración de SQL a NoSQL, ya que implica adaptar los datos a un nuevo modelo. Debido a que, para MongoDB, se debe definir las referencias entre las colecciones.

También se debe tener en cuenta el tipo de variable para los datos en la tabla SQL, con el objetivo de evitar inconsistencias, como valores vacíos o nulos que podrían generar problemas en MongoDB

3. Carga de los Datos en MongoDB

Una vez que los datos han sido transformados, deben cargarse en la base de datos NoSQL. Este proceso, conocido como ETL (Extract, Transform, Load), tiene a MongoDB como destino final para la carga, y para este fin se utiliza Postman Runner (Figura 5), que automatiza la carga de datos a través de solicitudes HTTP.

Los datos extraídos se convierten a un formato JSON, para este fin es necesario que la estructura del archivo JSON este bien estructurado y adecuado para Mongo DB.

Al hacer automática la carga de los datos, se facilita la migración. También en la etapa de pruebas del sistema ayuda a comprobar la carga masiva de datos directamente a una base en un entorno local antes del despliegue de la aplicación.

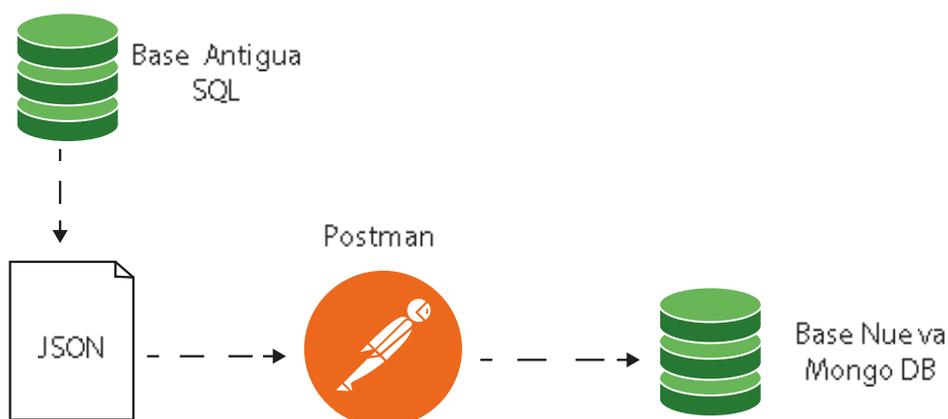


Figura 5 ETL sistema contable. Fuente: Autores

Verificación y Validación

Tras la migración, es indispensable realizar una verificación de los datos para asegurarse de que todos los datos han sido cargados correctamente y que no se ha perdido información. Esto puede incluir la ejecución de consultas en MongoDB para comparar los datos con los del sistema SQL original.

4.1.2.11 Desarrollo del Frontend

Para el desarrollo del Frontend se cubrió un desarrollo intuitivo y modular que permita una experiencia de usuario ágil y eficiente. El framework principal fue **Vue.js** que permite gestionar las diferentes funcionalidades del sistema como: la administración de clientes, proveedores y la generación de facturas. Los aspectos más importantes en el desarrollo técnico del frontend se muestran a continuación:

4.1.2.11.1 Organización del Código

El frontend se organizó mediante una arquitectura modular, dividiendo la aplicación en componentes claramente definidos. Cada módulo gestiona una parte específica del sistema como: clientes, proveedores o facturación; Esto permite el trabajo en secciones independientes del código, sin generar dependencias innecesarias. Permitiendo así la facilidad en el mantenimiento y escalabilidad del sistema.

- **Gestión de Clientes y Proveedores:** Se desarrolla formularios interactivos que permiten crear, editar y eliminar registros de clientes y proveedores. Cada uno

de estos vinculado con las operaciones de backend mediante solicitudes asíncronas con Axios, se pudo actualizar los datos en tiempo real sin necesidad de recargar la página, Además, para optimizar la experiencia del usuario al seleccionar datos en los formularios con muchas opciones, se integró **v-select2-component**.

- **Módulo de Facturación:** Se desarrollo un módulo que facilita el proceso de facturación, desde la selección de productos hasta la aplicación de descuentos y el cálculo de impuestos, donde cada producto tiene la opción de definir su propio descuento y tasa de impuestos, lo que brinda mayor flexibilidad al usuario. Además, la facturación se conecta directamente con el backend para generar los documentos XML requeridos por el SRI

4.1.2.11.2 Integración con Backend y Autenticación

La comunicación entre el frontend y el backend se gestiona utilizando **Axios**, una librería que facilita las solicitudes HTTP. Cada componente del frontend tiene sus propios métodos para interactuar con los microservicios del backend, lo que hace que las operaciones como la creación o modificación de clientes y proveedores sean rápidas y eficientes. Además, se ha implementado un sistema de autenticación basado en **JWT** que asegura que solo los usuarios autorizados puedan acceder a las funciones protegidas del sistema. El token JWT se obtiene al iniciar sesión y se almacena en el estado global utilizando **Vuex**, lo que permite que el token esté disponible para todas las solicitudes que requieran autenticación.

4.1.2.11.3 Validación de Formularios y Manejo de Datos

Una de las áreas clave en el desarrollo del frontend fue la implementación de validaciones dinámicas en los formularios. Los formularios, tanto para la creación como para la edición de datos, están equipados con validaciones en tiempo real que alertan al usuario sobre cualquier error antes de que se envíen los datos al servidor. Se diseñaron reglas de validación específicas para campos críticos, como la verificación de números de identificación tributaria (RUC, cédula) y correos electrónicos.

Además, para el manejo de fechas y plazos de vencimiento de facturas, se utilizó **Moment.js**, una herramienta que facilita la manipulación y formateo de fechas. Esto resultó especialmente útil en la facturación, donde es crucial que las fechas de emisión y vencimiento se manejen de forma precisa.

4.1.2.11.4 Alertas y Notificaciones para el Usuario

Con el fin de mejorar la interacción con el usuario, se integraron alertas visuales personalizadas utilizando **SweetAlert2**. Estas alertas se activan en momentos clave del flujo de trabajo, como la confirmación de eliminación de un cliente o la creación exitosa de una factura. Además, se manejaron diferentes tipos de notificaciones para advertir sobre errores del sistema, por ejemplo, cuando hay fallos en la conexión con el Servicio de Rentas Internas (SRI) o en caso de problemas al firmar electrónicamente una factura.

4.1.2.11.5 Optimización del Desempeño y Carga de Datos

Para asegurar que el frontend ofrezca un rendimiento óptimo, se implementaron varias técnicas de optimización. Una de las más importantes fue el uso de **Lazy Loading** para cargar componentes y vistas únicamente cuando sean necesarios. Esto reduce el tiempo de carga inicial de la aplicación, permitiendo que los usuarios interactúen con las secciones críticas del sistema más rápidamente.

Otra optimización clave fue la minificación del código de JavaScript y CSS durante el proceso de compilación, lo que reduce el tamaño de los archivos enviados al navegador, mejorando así los tiempos de carga. Asimismo, se implementó el almacenamiento en caché de algunas solicitudes, como las listas de clientes y productos, lo que disminuye la carga en el backend y mejora la experiencia del usuario en términos de velocidad.

4.1.2.11.6 Pruebas y Validación del Frontend

Se realizaron pruebas exhaustivas para asegurar la estabilidad y funcionalidad del frontend. Esto incluyó pruebas unitarias de los componentes más críticos, garantizando que cada uno funcionara de manera independiente y correcta.

Además, se llevaron a cabo **pruebas de usabilidad** con usuarios finales para identificar posibles áreas de mejora en la experiencia de usuario, ajustando la interfaz para que sea lo más intuitiva posible, especialmente en el manejo de los módulos de facturación y gestión de clientes.

4.1.2.11.7 Despliegue y Contenerización

El despliegue del frontend se realizó utilizando **Docker**, lo que permite que la aplicación funcione de manera consistente en diferentes entornos sin importar las configuraciones del servidor. La contenerización del frontend facilita la integración continua y el despliegue automático, asegurando que las actualizaciones puedan realizarse sin interrupciones para los usuarios. Además, se configuró un servidor **Nginx** dentro del contenedor para servir la aplicación de manera eficiente, optimizando la respuesta y manejo de solicitudes.

4.1.2.12 Despliegue y Mantenimiento

El despliegue y mantenimiento de un sistema contable basado en microservicios implica la implementación de cada servicio en un entorno de producción, que para este caso en particular se usó Docker para contenerizar los servicios y para la correcta orquestación de los mismos se usa Docker Compose (Figura 6) que permite gestionar múltiples contenedores y así automatizar su despliegue tanto en desarrollo, pruebas y producción.

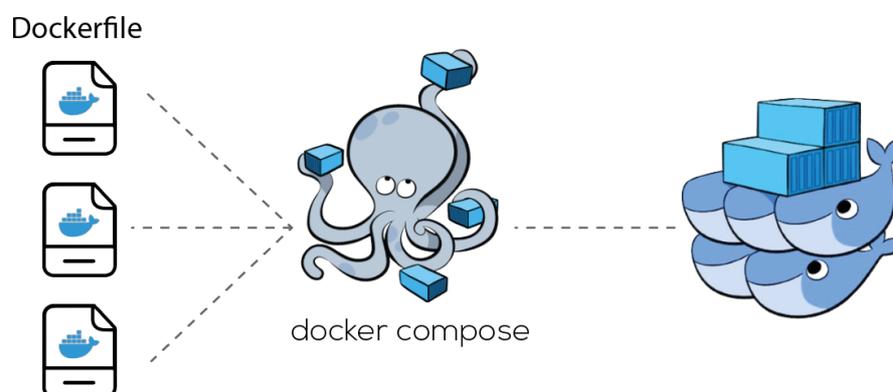


Figura 6 docker compose. Fuente: Autores

En cuanto al mantenimiento que se debe realizar dentro del sistema se incluyen tareas de actualización de dependencias de cada servicio, resolución de errores y

adaptación a nuevas normativas tributarias que pudiesen surgir. El sistema al ser elaborado con la arquitectura de microservicios, hace más sencillo el mantenimiento ya que se puede actualizar cada servicio por separado evitando el riesgo de afectar al sistema completo.

4.1.3 VALIDACIÓN DEL SISTEMA

La etapa final de cada sprint se debe validar las funcionalidades del sistema que serían los entregables para cada sprint, las mismas deben cumplir con las expectativas de los usuarios y con la normativa tributaria existente, incluida la revisión de las funcionalidades de conexión con sistemas externos en este caso el envío y la autorización del SRI. Por esta razón, tenemos que considerar el cálculo de los impuestos bajo la normativa tributaria actual del Ecuador, siguiendo los estándares de generación y envío de documentos electrónicos.

Pruebas de Usabilidad:

Estas pruebas buscan comprobar que los usuarios finales puedan interactuar con el sistema, y comparar tiempos de respuesta en relación con el sistema anterior. Para esto se tiene en cuenta varios aspectos como por ejemplo, la rapidez de uso del sistema y la precisión en la emisión de facturas y retenciones, (Figura 7) así como el grado de satisfacción de cada usuario, a través del seguimiento en los tiempos que tardan los usuarios en completar las tareas de emisión de documentos electrónicos para medir las mejoras en la experiencia de uso.



Figura 7 Interfaz de usuario anterior vs actual. Fuente: Autores

4.1.4 PRUEBAS DE SEGURIDAD

La seguridad es un pilar fundamental en la validación de este sistema, pues garantiza el manejo adecuado y la protección de información sensible, como los datos de clientes, proveedores y transacciones. Anteriormente, la falta de control en el acceso a la información exponía los datos a cualquier persona con conocimiento de la IP del servidor, aumentando el riesgo de accesos no autorizados. La Tabla 2 muestra las pruebas de seguridad aborda exhaustivamente los controles de autenticación, autorización, encriptación y manejo de errores para mitigar estos riesgos y asegurar la confidencialidad, integridad y disponibilidad de los datos.

Tabla 2 Pruebas de seguridad

Tipo de Prueba	Objetivo	Endpoints	Criterios de Validación
Autenticación (Login y Token)	Verificar que solo usuarios autenticados accedan a los endpoints protegidos.	POST /auth/login	Retorna un token JWT válido con credenciales correctas; 401 Unauthorized con credenciales incorrectas.
Autorización	Asegurar que solo usuarios con permisos accedan a recursos específicos.	GET /clients/:id, DELETE /clients/:id	Sin token: 403 Forbidden; token inválido: 401 Unauthorized; token válido: 200 OK con acceso al recurso.
Inyecciones SQL y XSS	Prevenir ataques de inyección en campos sensibles.	POST /clients	Código malicioso no causa 500 Internal Server Error; el sistema rechaza entradas peligrosas con 400 Bad Request.
Seguridad de Transmisión (HTTPS)	Confirmar que todas las conexiones de datos sensibles usen HTTPS.	Todos los endpoints	Conexión no segura es rechazada; el sistema debe redirigir o bloquear HTTP no seguro.
Integridad del Token JWT	Validar que los tokens no puedan ser manipulados.	Todos los endpoints protegidos	Token alterado es rechazado con 401 Unauthorized.
Limitación de Intentos de Login	Proteger contra ataques de fuerza bruta limitando intentos de login fallidos.	POST /auth/login	Tras varios intentos fallidos, el sistema responde con 429 Too Many Requests o bloquea temporalmente la IP.
Manejo de Errores	Asegurar que los mensajes de error no expongan información sensible del sistema.	Todos los endpoints	Mensajes de error son genéricos y no exponen detalles técnicos como rutas o estructura de base de datos.

4.1.5 BUENAS PRÁCTICAS DE DESARROLLO:

El desarrollo del sistema sigue principios de código limpio y principios SOLID, asegurando que el código sea mantenible, escalable con el tiempo y fácil de comprender. Estas buenas prácticas contribuyen a la calidad del software y facilitan el mantenimiento futuro, disminuyendo también la degradación de código por modificaciones.

5 RESULTADOS Y DISCUSIÓN

En esta sección, se detallan los resultados cuantitativos y cualitativos obtenidos tras la implementación del sistema contable en Accescont. Estos resultados reflejan directamente en la migración de datos, rendimiento, funcionalidad y la satisfacción de los usuarios.

5.1 SESIONES DE LEVANTAMIENTO DE INFORMACIÓN Y SOCIALIZACIÓN

La siguiente imagen muestra las **sesiones de levantamiento de información y socialización** (Figura 8) las mismas que fueron exitosas, ya que permitieron captar de manera precisa las necesidades del proyecto. Gracias a la participación activa de todos los involucrados, se lograron ajustar los objetivos y alinear el sistema con las expectativas de los usuarios. Esto facilitó la comprensión del nuevo sistema y generó un sentido de compromiso entre los participantes, sentando una base sólida para la implementación del proyecto.



Figura 8 Socialización y levantamiento de información. Fuente: Autores

5.2 DISEÑO DE ARQUITECTURA.

El diseño de la arquitectura del sistema se basó en la arquitectura de microservicios, por ende permitió la separación de las tareas del sistema y la escalabilidad del mismo. La separación del sistema en módulos independientes, de clientes y proveedores, eliminó las fallencias del sistema anterior, en términos de flexibilidad, despliegue y mantenimiento.

5.2.1 CARACTERÍSTICAS DE LA ARQUITECTURA DE MICROSERVICIOS.

Modularidad.

Cada microservicio se desarrolló para gestionar funcionalidades específicas, como la generación de facturas electrónicas o validación de retenciones y la firma de documentos electrónicos. Todas estas mejoras permiten realizar mantenimientos en el sistema o actualizaciones sin afectar a otros módulos del sistema.

Escalabilidad.

El uso de Docker para la contenerización de microservicios garantiza su despliegue y escalamiento de forma independiente según avancen las necesidades del sistema.

Interoperabilidad

Con el uso de APIs RESTful para la comunicación entre los microservicios y la integración con sistemas externos, en este caso el sistema del SRI, se asegura el correcto flujo de datos y la conformidad con la normativa contable de Ecuador.

Base de datos distribuida.

Cada microservicio posee su propia base de datos que satisface sus necesidades operativas, por lo que se evitan los cuellos de botella al acceder a la información y se asegura la seguridad de los datos de los clientes y proveedores.

5.3 DESARROLLO DE MÓDULOS DE CLIENTES Y PROVEEDORES

En base la arquitectura definida del proyecto se definieron las funciones que deben tener los módulos de clientes y proveedores, por lo que según los requisitos identificados, los resultados que se obtuvieron son los siguientes:

5.3.1 FUNCIONALIDAD DE LOS MÓDULOS DE CLIENTES Y PROVEEDORES.

Los módulos de clientes y proveedores implementados en el sistema contable cumplen con las funcionalidades requeridas por los usuarios, así como con la normativa y se alinean con los objetivos del negocio de Accescont,.

Los resultados obtenidos a partir de las evaluaciones a los usuarios del sistema mediante el Formulario de Validación que se encuentra en los anexos, se detallan a continuación:

Gestión eficiente: Los usuarios interactúan con todo el CRUD de clientes de manera ágil e intuitiva, por ende, se cumple con la satisfacción de los usuarios en el manejo de la aplicación.

Validaciones automáticas: Se han integrado validaciones automáticas para los números de identificaciones ingresadas al sistema, también elimina la posibilidad de ingreso de registros duplicados, reduciendo los errores en la introducción de datos y mejorando la precisión de las bases de datos al momento de realizar una consulta.

Interfaz de usuario intuitiva: El manejo del sistema se refleja una mejora significativa en tiempos de respuesta, en relación con el sistema anterior.

Compatibilidad con SRI: Los módulos de clientes y proveedores están completamente integrados con las funcionalidades externas del SRI para los servicios de facturación electrónica y retenciones respectivamente, asegurando la conformidad legal, y garantizando la autorización de los documentos emitidos.

Escalabilidad: La arquitectura de microservicios facilita la expansión y mejora continua de estos módulos sin comprometer el sistema global, permitiendo añadir más funcionalidades y servicios conforme las necesidades de la empresa y según el plan de negocios de la misma.

A futuro se tiene planes de expansión del sistema en los módulos de asientos contables y registros de cuentas.

5.3.2 MÓDULO DE CLIENTES

El **módulo de clientes** del sistema permite una gestión integral y eficiente de las relaciones con los clientes. En primer lugar, ofrece la funcionalidad de **creación y gestión de clientes**, donde los usuarios pueden registrar nuevos clientes y actualizar sus datos según sea necesario, asegurando que la información siempre esté actualizada, lo mencionado se muestra en la Figura 9

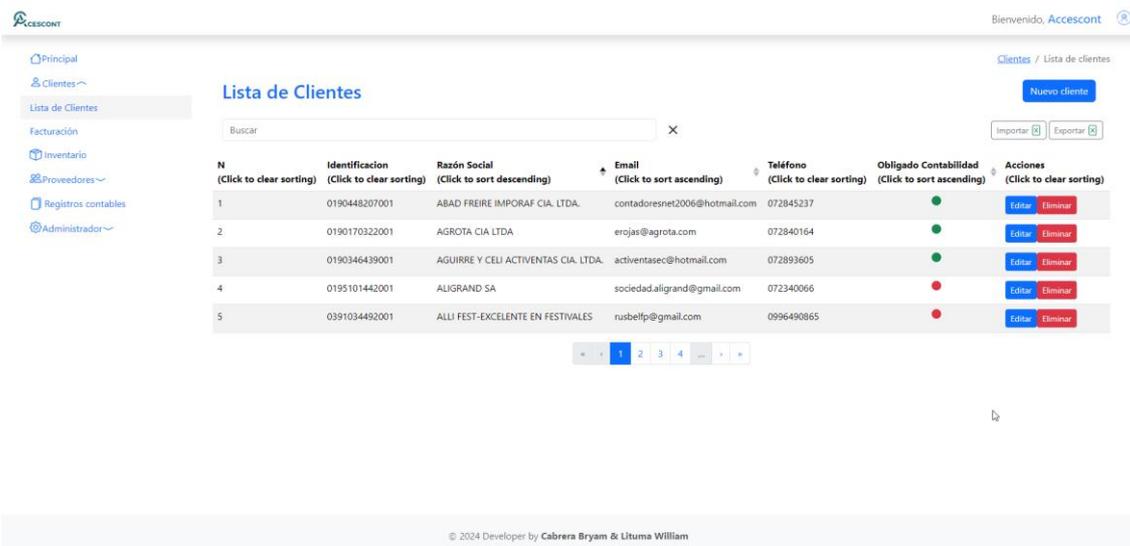


Figura 9 Gestión de clientes. Fuente: Accescont

De igual manera, el módulo facilita la creación y gestión de productos o servicios (Figura 10), permitiendo a los usuarios agregar, modificar o eliminar los productos o servicios que se ofrecen a los clientes, con detalles específicos como se muestra a continuación

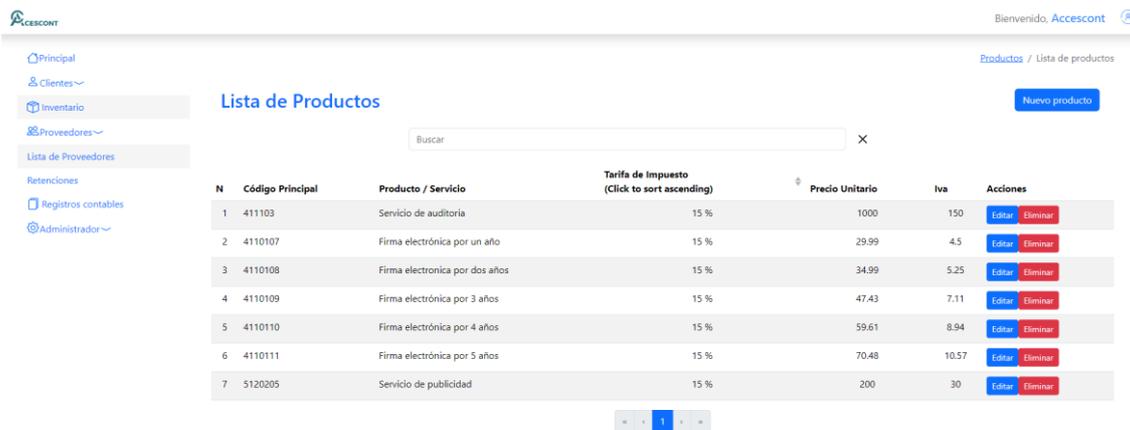


Figura 10 Gestión de productos o servicios. Fuente: Accescont

Finalmente, una de sus funciones principales es la **emisión de facturas a clientes**, donde los usuarios pueden generar facturas de manera sencilla, asociando productos o servicios, aplicando descuentos y calculando impuestos automáticamente. Todo esto garantiza un proceso de facturación ágil ajustado a las necesidades de la empresa y del cliente, como se muestra en la Figura 11.

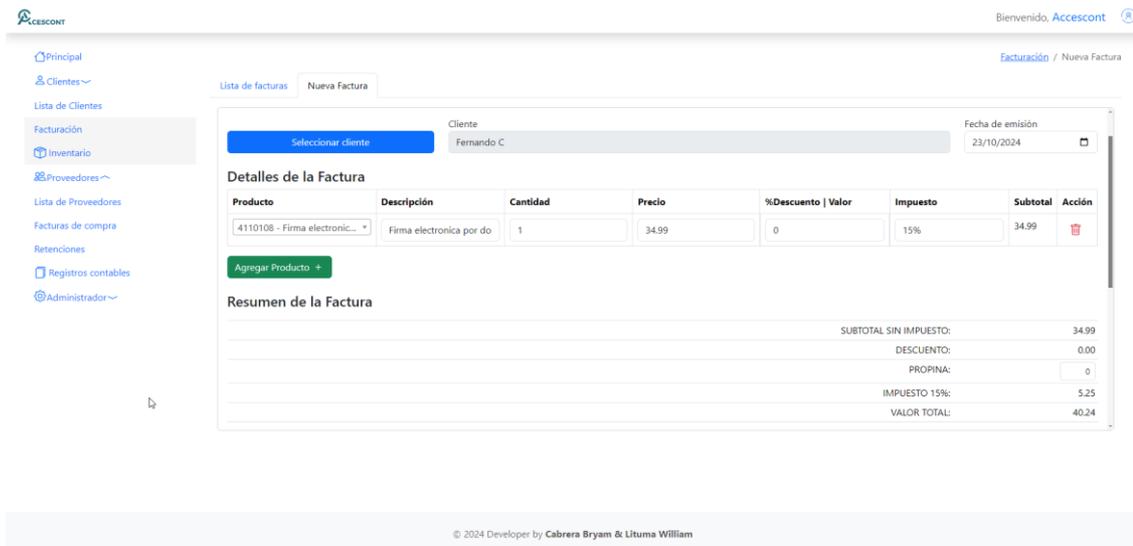


Figura 11 Generación de una Factura a un cliente. Fuente: Accescont

La Figura 12 muestra el proceso de **consulta del comprobante electrónico en el portal del SRI**. Una vez enviado el comprobante desde el sistema, éste es autorizado por el SRI, y el usuario puede visualizar todos los detalles de la factura, incluyendo la clave de acceso, la fecha y hora de autorización, y el valor total, lo que garantiza la validación fiscal.

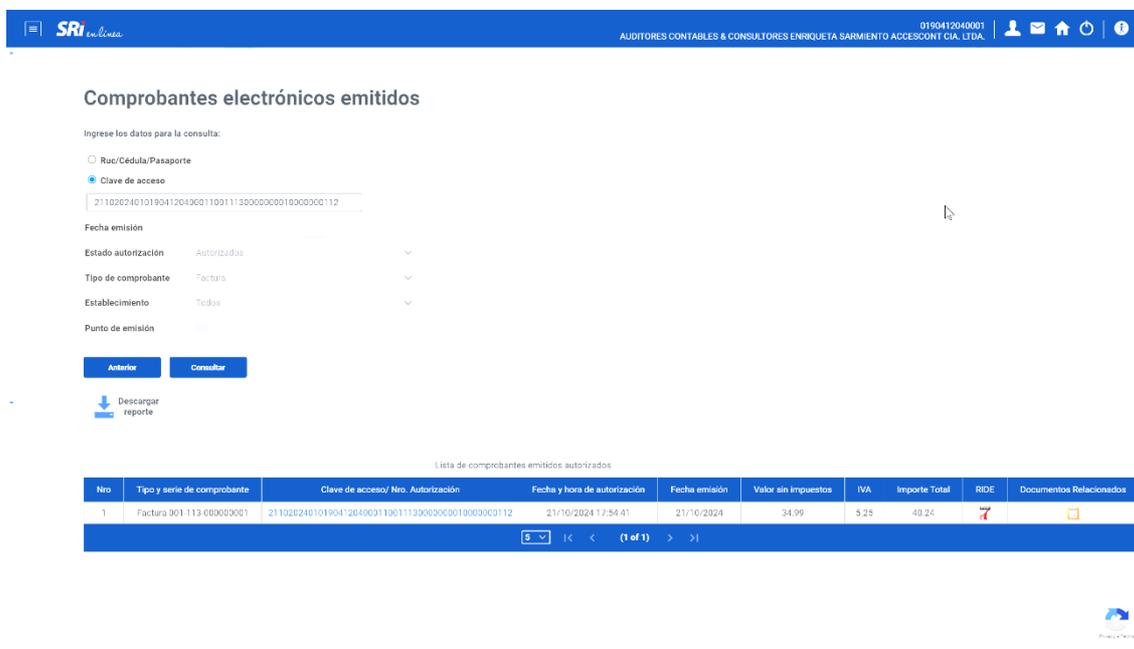
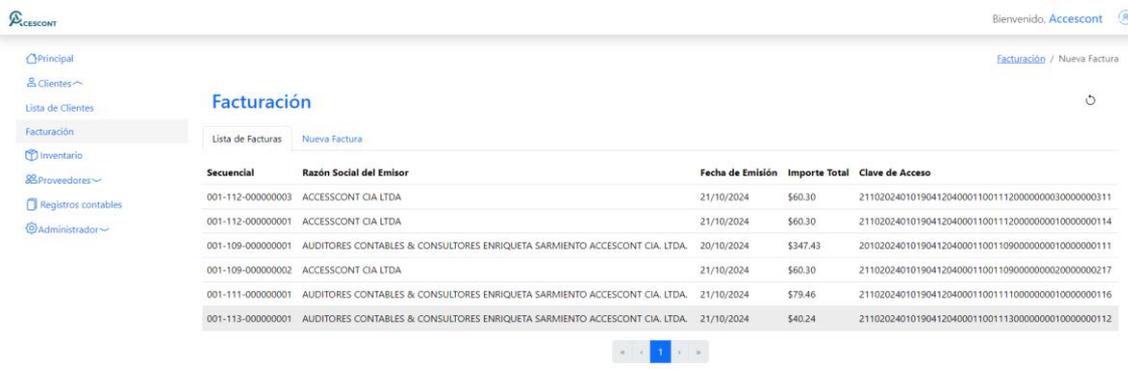


Figura 12 Verificación en el SRI de una Factura emitida. Fuente: Accescont

La Figura 13 muestra la **lista de facturas emitidas** desde el módulo de facturación del sistema, donde se puede observar cada comprobante con su respectivo número

de autorización y fecha de emisión, proporcionando un control claro y ordenado de las transacciones emitidas a los clientes.



Secuencial	Razón Social del Emisor	Fecha de Emisión	Importe Total	Clave de Acceso
001-112-000000003	ACCESSCONT CIA LTDA	21/10/2024	\$60.30	21102024010190412040001100111200000000300000000311
001-112-000000001	ACCESSCONT CIA LTDA	21/10/2024	\$60.30	21102024010190412040001100111200000000100000000114
001-109-000000001	AUDITORES CONTABLES & CONSULTORES ENRIQUETA SARMIENTO ACCESSCONT CIA. LTDA.	20/10/2024	\$347.43	20102024010190412040001100110900000000100000000111
001-109-000000002	ACCESSCONT CIA LTDA	21/10/2024	\$60.30	21102024010190412040001100110900000000200000000217
001-111-000000001	AUDITORES CONTABLES & CONSULTORES ENRIQUETA SARMIENTO ACCESSCONT CIA. LTDA.	21/10/2024	\$79.46	21102024010190412040001100111000000000100000000116
001-113-000000001	AUDITORES CONTABLES & CONSULTORES ENRIQUETA SARMIENTO ACCESSCONT CIA. LTDA.	21/10/2024	\$40.24	211020240101904120400011001113000000000100000000112

Figura 13 Lista de Facturas autorizadas. Fuente: Accescont

La Figura 14 **Error! No se encuentra el origen de la referencia.** muestra una factura electrónica emitida y autorizada por el SRI, con todos los detalles del emisor, cliente, productos y valores, incluyendo la clave de acceso para su validación. Este documento refleja una transacción clara y completa, cumpliendo con las normativas fiscales.



R.U.C.: 0190412040001

FACTURA

No. 001-113-000000001

NÚMERO DE AUTORIZACIÓN

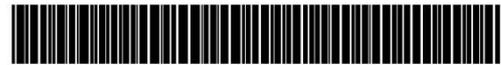
2110202401019041204000110011130000000010000000112

FECHA Y HORA DE AUTORIZACIÓN: 21/10/2024 17:54:41

AMBIENTE: PRUEBAS

EMISIÓN: NORMAL

CLAVE DE ACCESO



2110202401019041204000110011130000000010000000112

AUDITORES CONTABLES & CONSULTORES ENRIQUETA SARMIENTO ACCESCONT CIA. LTDA.
 ACCESCONT CIA. LTDA.
 Dirección Matriz: TOMÁS ORDOÑEZ 14-31 Y PIO BRAVO
 Dirección Sucursal: DE LA MISTELA Y RAFAEL CARPIO ABAD
 OBLIGADO A LLEVAR CONTABILIDAD SI

Razón Social / Nombres y Apellidos: Fernando C
 Identificación: 0105246318
 Fecha: 21/10/2024 Placa / Matrícula: Guía
 Dirección:

Cod. Principal	Cod. Auxiliar	Cantidad	Descripción	Detalle Adicional	Precio Unitario	Subsidio	Precio sin Subsidio	Descuento	Precio Total
4110108		1.00	Firma electronica por dos años		34.99	0.00	0.00	0.00	34.99

Forma de pago	Valor
20 - OTROS CON UTILIZACION DEL SISTEMA FINANCIERO	40.24

SUBTOTAL 15%	34.99
SUBTOTAL 0%	0.00
SUBTOTAL NO OBJETO DE IVA	0.00
SUBTOTAL EXENTO DE IVA	0.00
SUBTOTAL SIN IMPUESTOS	34.99
TOTAL DESCUENTO	0.00
ICE	0.00
IVA 15%	5.25
TOTAL DEVOLUCION IVA	0.00
IRBPNR	0.00
PROPINA	0.00
VALOR TOTAL	40.24
VALOR TOTAL SIN SUBSIDIO	0.00
AHORRO POR SUBSIDIO: (Incluye IVA cuando corresponda)	0.00

Figura 14 Comprobante de Facturación. Fuente: Accescont

5.3.3 MÓDULO DE PROVEEDORES

El **módulo de proveedores** del sistema ofrece una forma sencilla y eficiente de gestionar las relaciones con los proveedores. Entre sus principales funciones está la **creación y administración de proveedores**, lo que permite registrar nuevos proveedores de manera rápida y mantener actualizada la información de contacto o datos importantes cuando sea necesario como se muestra en la Figura 15.

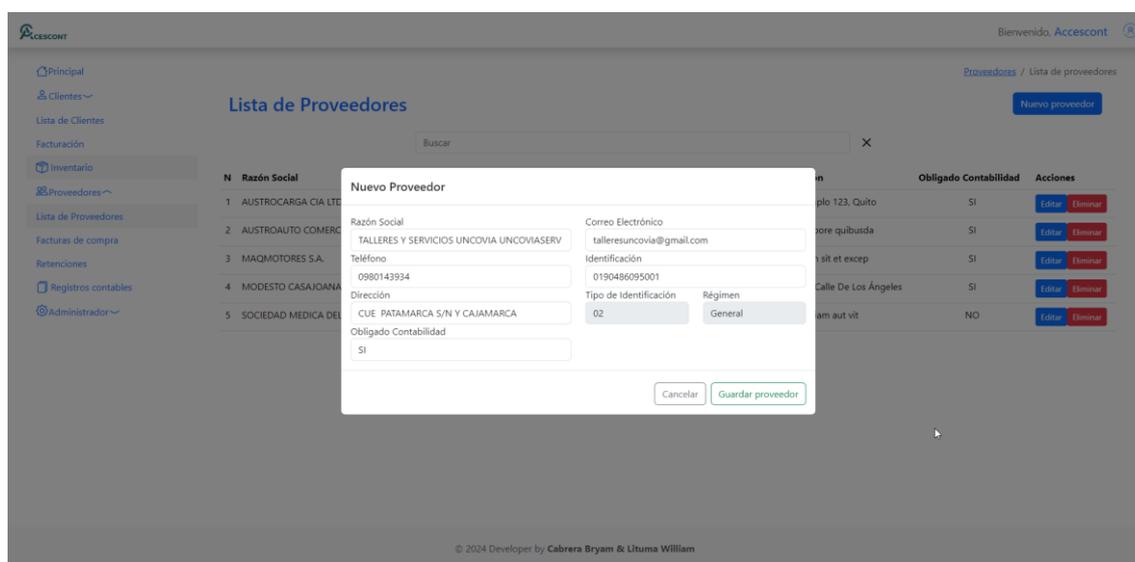


Figura 15 Gestión de Proveedores. Fuente: Accescont

También facilita el **registro de facturas de compra**, permitiendo a los usuarios ingresar detalles como la fecha de emisión, el número del comprobante, la razón social del emisor y el monto total de la compra. Esto asegura que todas las transacciones con proveedores queden registradas de manera clara y ordenada. Como se muestra a continuación en la Figura 16.

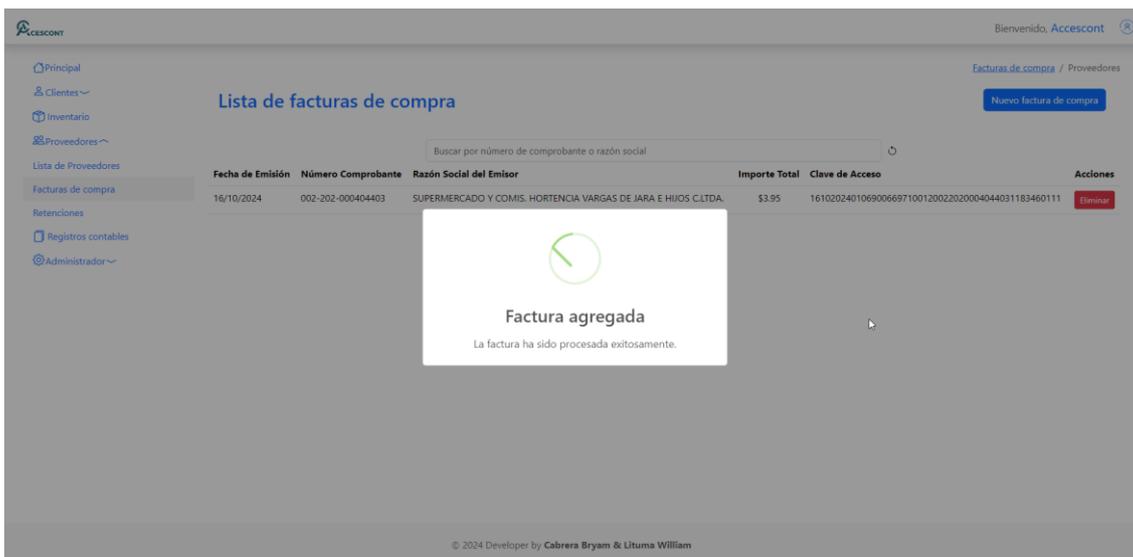


Figura 16 Registro de Factura de Compra. Fuente: Accescont

Por último, el módulo de proveedores permite emitir retenciones de manera sencilla, calculando automáticamente los impuestos a partir de las facturas ingresadas. Los usuarios pueden generar el comprobante de retención y enviarlo al SRI para su autorización, cumpliendo con las normativas fiscales. En la Figura 17, se muestra cómo el usuario ingresa los detalles de la retención, la fecha, impuestos y porcentaje, antes de enviarla para su aprobación.

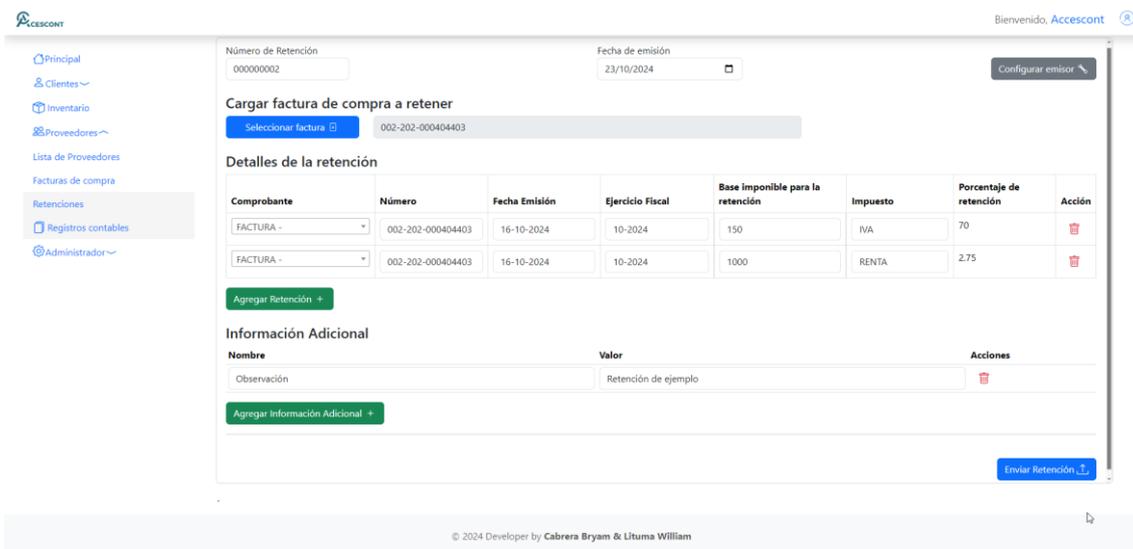


Figura 17 Emisión de retención desde el módulo proveedores. Fuente: Accescont

La Figura 18, muestra la interfaz del **SRI en línea**, donde se realiza la consulta del comprobante de retención que fue enviado previamente. Aquí, se puede visualizar



R.U.C.: 0190412040001

COMPROBANTE DE RETENCIÓN

No. 001-114-000000002

NÚMERO DE AUTORIZACIÓN

161020240701904120400011001114000000020000000210

FECHA Y HORA DE AUTORIZACIÓN 22/10/2024 16:50:14

AMBIENTE: PRUEBAS

EMISIÓN: NORMAL

CLAVE DE ACCESO



ACCESSCONT CIA LTDA

Dirección Matriz: REMIGIO TAMARIZ 004 Y EL BATAN

Dirección Sucursal: REMIGIO TAMARIZ 004 Y EL BATAN

OBLIGADO A LLEVAR CONTABILIDAD SI

Razón Social / Nombres y Apellidos: Proveedor Ejemplo S.A.

Identificación 1726722646001

Fecha 16/10/2024

Comprobante	Número	Fecha Emisión	Ejercicio Fiscal	Base Imponible para la Retención	Impuesto	Porcentaje Retención	Valor Retenido
FACTURA	002202000404403	12/10/2024	10/2024	150	IVA	70.0	105
FACTURA	002202000404403	12/10/2024	10/2024	1000	Impuesto a la Renta	2.75	27.5

Información Adicional

Observación:	Retención de ejemplo
--------------	----------------------

Figura 19 Comprobante de retención generado. Fuente: Accescont

5.3.4 SISTEMA ANTIGUO FRENTE AL SISTEMA ACTUAL

Como se muestra en la Figura 20 y Figura 21, el sistema antiguo presenta una interfaz gráfica basada en un entorno de terminal, donde los datos de los clientes se visualizan en un formato de texto plano. Los usuarios interactúan con este sistema mediante comandos numéricos asignados a diferentes funciones como

cancelar, modificar, eliminar, consultar, imprimir o manejar documentos. Los comandos y el filtro son poco accesibles, ya que la búsqueda de clientes se realiza de manera manual, ingresando ciertos parámetros en campos específicos, lo que resulta poco intuitivo y limitante para los usuarios menos experimentados.

Empresa: ACCESCONT CIA LTDA		2024/Oct/23 INVP001.CBL		
Usuario: BRYAM CABRERA		BD: CLIENTES DEL SISTEMA		
[0] Cancelar [1] Modificar [2] Eliminar [3] Consultar [4] Imprimir [5] Documentos				
? NOMBRE	CODIGO	CEDULA/RUC	TELEFONO	SALDO
ABAD FREIRE IMPORAF CIA. LTDA.	ABAFRE01	0190448207001	072845237	0.00
AGROTA CIA LTDA	AGRCIA01	0190170322001	072840164	0.00
AGUIRRE Y CELI ACTIVENTAS CIA. LTDA.	AGUY 01	0190346439001	072893605	1,465.71
ALIGRAND SA	ALISA 01	0195101442001	072340066	0.00
ALIMENTICIAS SAN JOSE	ALISANO1			0.00
ALLI FEST-EXCELENTE EN FESTIVALES	ALLFES01	0391034492001	0996490865	0.00
ALMACEN JUAN MONTERO CIA. LTDA.	ALMJUA01	0190055507001	072863755	600.00-
ALMACENES FABIAN PINTADO GFP CIA. LTDA.	ALMFAB01	0190446840001	074120178	0.00
AMIMPLUS CIA. LTDA.	AMICIA01	0190388999001	074213062	558.25
ANA DALILA PEÑALOZA CHAMIK	DALPEÑ01	1400894091	0959554141	0.00
ANCHUNDIA PEÑA BYRON DAVID	ANCPEN01	0103641692	0993514533	0.00
ANDRADE TORRES MONICA ELIZABETH	ANDTOR01	0103140489001	0958917423	0.00
ANGEL SANCHEZ	ANGSAN01	1103573703	0981520625	0.00
ARIEL RAJID ABAD GOMEZ	ARIRAJ01	0926335811001	0991612358	0.00
AROMASAS S.A.S	AROS.A01	0195096260001	0987177588	0.00
ASOCIACION DAMAS SALESIANAS DE CUENCA	ASODAMO1	0190312798001	072841107	0.00
ASOCIACION DE PRODUCTORES CAMARONEROS ASO&CAM	ASODE 01	0791819393001	072148678	0.00
AUSTROAUTO COMERCIAL CIA LTDA	AUSCOM01	0190168026001	074104040	0.00
AUSTROCARGA CIA LTDA	AUSCIA01	0190169669001	074086723	0.00
AUSTROIMAGENES S.A.	AUSS.A02	0190170594001	072889464	0.00
AUSTROPRONTO S.A.	AUSS.A01	0190352420001	074092148	0.00
AUTOPASE S.A.	AUTS.A01	0190067009001	0939464988	59.20-
AVILA BRITO VICTOR JAIME	AVIBRIO1	0100875160001	072460119	270.00
AVOREPHARMA S.A.	AVOS.A01	0190412245001	074193601	0.00

F1-Ayuda F2-Agregar F3-Insertar F4-Buscar F5-Imprimir F6-Mas funciones

Advertencia: Version Review 5.0

Figura 20 Módulo de clientes sistema anterior. Fuente: Accescont

Empresa: ACCESCONT CIA LTDA		2024/Oct/23 INVP001.CBL		
Usuario: BRYAM CABRERA		BD: CLIENTES DEL SISTEMA		
[0] Cancelar [1] Modificar [2] Eliminar [3] Consultar [4] Imprimir [5] Documentos				
? NOMBRE	CODIGO	CEDULA/RUC	TELEFONO	SALDO
LOJA FABIOLA	LOJFAB01	0102557741	0990547602	0.00
LOJA HERNANDEZ BRIAN ALEXIS	LOJHER01	0106604507001	0987405480	0.00
LOJA SALINAS WILSON OSWALDO	LOJSAL01	1715306583001	0985677260	0.00
LOJA-HERNANDEZ CONSTRUCCIONES LHCONSTRUC CIA.L	LOJCON01	0190503844001	0984038283	805.00
LOJANO PUNIN JESSENIA ELIZABETH	LOJPUN01	0106946411	0985950904	0.00
LOPEZ QUEVEDO MARIA AUGUSTA	LOPQUE01	0300834975001	072240393	0.00
LUCILA MONRROY BRANDA	LUCMON01	0925318578	0962079506	0.00
LUNA QUITO JESSICA GABRIELA	LUNQUI01	0107347395001	0992289770	0.00
LUZ MERCEDES GUAZHAMBO QUIZHPI	LUZMER01	0105976898001	0986252324	0.00
MACAS TAPIA LAURA ISABEL	MACTAPO1	0105838056001	0987448208	0.00
MADERPALLET CIA. LTDA.	MADCIA01	0190349527001	074121440	22.11
MANZANO MOSQUERA GLADYS ALEXANDRA	MANMOS01	0102314267001	0999576910	34.99
MAQMOTORES S.A.	MAQS.A01	0993211141001	072844210	0.00
MARIA CABRERA PEREZ	MARCBRO1	0930188933001	0939657252	0.00
MARTINEZ USHINA HUGO EDUARDO	MARUSH01	1803675139001	0999886453	0.00
MARVICNET CIA. LTDA.	MARCTA01	0190451194001	074034627-	1,780.62
MEDINA JARA MANUEL JACINTO	MEDJAR01	0102202884001	0994200324	24.99
MEDINA MACHUCA JOSE ALEJANDRO	MEDMAC01	0100221597001	072850140	0.00
MEJIA INIGUEZ SILVIA KATIUSCA	MEJINI01	0102643335001	0728111470	0.00
MEJIA REGALADO THANIA	MEJREG01	0102103926001	0994931450	0.00
MERCHAN JORGE LEONARDO	MERJOR01			0.00
MERCHAN VELEZ JORGE LEONARDO	JORMER01	1102589874001	0993320668	0.00
MERCOINTELG S.A.	MERS.A01	0190328694001	4134720	0.00
MINMETECECUADOR CIA. LTDA.	MINCIA01	0190392694001	072901195	0.00

Nombre: LOJA Codigo: RUC:

Advertencia: Version Review 5.0

Figura 21 Módulo de clientes búsqueda. Fuente: Accescont

El nuevo sistema de gestión de clientes es más intuitivo y fácil de usar, con una interfaz moderna que presenta una vista en tablas y permite buscar en cualquier campo, para realizar acciones como editar o eliminar clientes de forma sencilla. También facilita la importación y exportación de datos, y usa íconos claros para

gestionar la obligatoriedad contable. Además, la paginación mejora la navegación, haciendo que la gestión de clientes sea más ágil y eficiente.

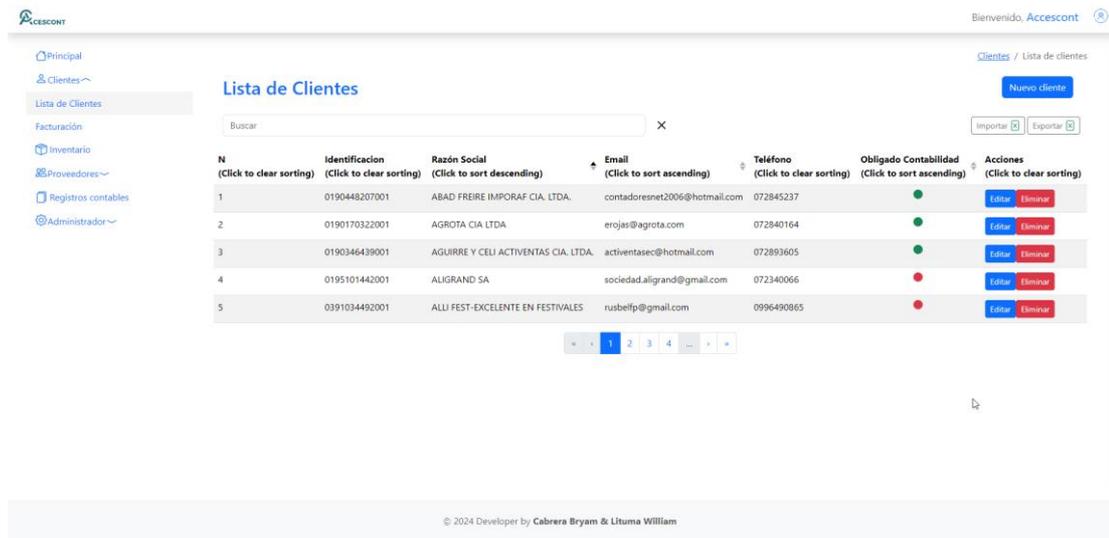


Figura 22 Módulo de clientes lista de clientes. Fuente: Accescont

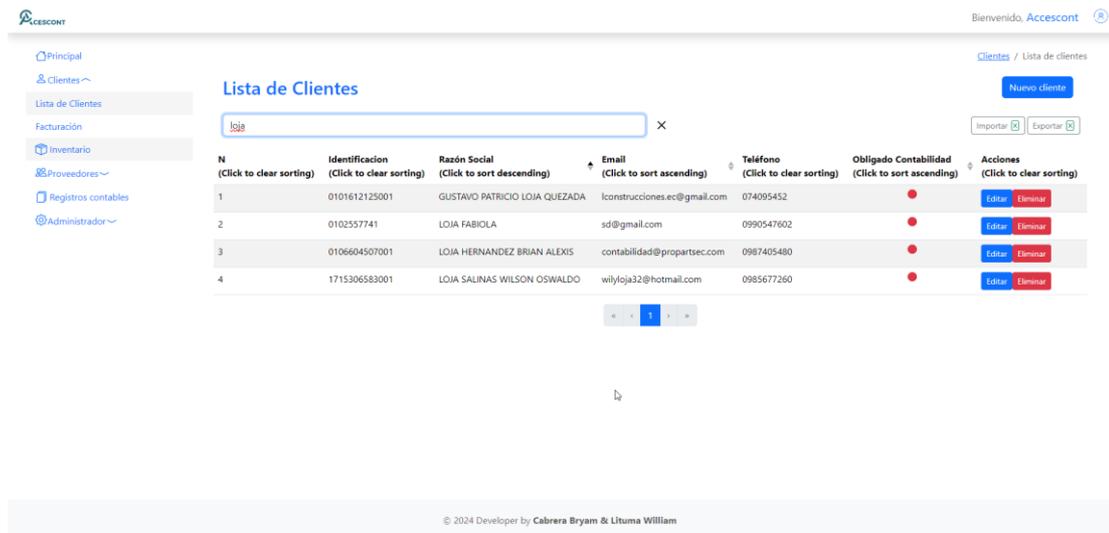


Figura 23 Módulo de clientes búsqueda. Fuente: Accescont

5.3.5 MEJORAS EN EL RENDIMIENTO:

Posterior a la implementación del sistema basado en microservicios, se observó una reducción significativa en los tiempos de emisión de documentos electrónicos en comparación con el sistema monolítico anterior, sobre todo con la implementación de la nueva interfaz que es mucho más amigable. El nuevo sistema demostró ser más eficiente y escalable, permitiendo manejar mayores volúmenes de

transacciones sin afectar su rendimiento. Además, la modularidad del sistema facilita su mantenimiento, lo que significa que se tiene mayor agilidad en la implementación de actualizaciones o mejoras.

5.3.6 MIGRACIÓN DE DATOS EXITOSA:

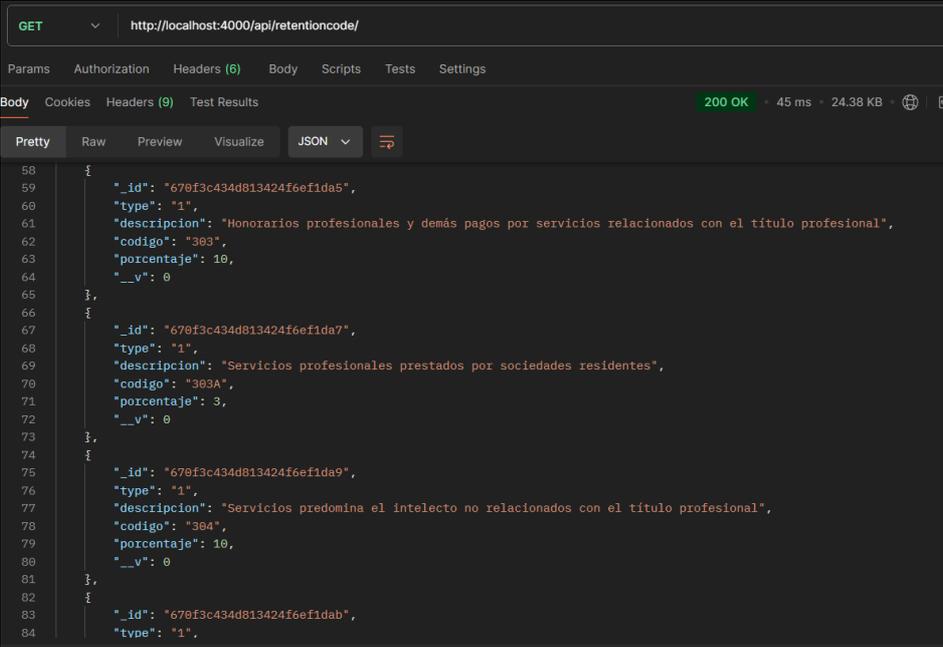
La migración de los datos de clientes y proveedores desde el sistema antiguo (Figura 24) al nuevo sistema basado en microservicios (Figura 25) se realizó con éxito. No se reportaron pérdidas de datos al momento de migrar la información de clientes y proveedores, por otro lado, como el esquema de datos de la factura es diferente, se implementó el guardado de facturas directamente desde el SRI al sistema actual, de esta manera se migraron los datos desde un ente externo, conservando la funcionalidad del sistema anterior.

Empresa: ACCESCONT CIA LTDA		2024/Oct/23 SRIP008.CBL
Usuario: BRYAM CABRERA		BD: CONCEPTOS DE RETENCION EN LA FUENT
[0] Cancelar [1] Modificar [2] Eliminar [3] Consultar [4] Imprimir [5] Detalle		
? CODI.	SE NOMBRE	PORC
146	01 OTRAS RETENCIONES APLICABLES A OTROS PORCENTAJES	0.00
303	01 HONORARIOS, COMISIONES Y DIETAS A PERSONAS NATURALES	10.00
303A	01 SERVICIOS PROFESIONALES PRESTADOS POR SOCIEDADES RESIDENTES	3.00
304	01 SERVICIOS PREDOMINA INTELLECTO, PAGOS A NOTARIOS, HONORARIOS	10.00
305	01 HONORARIOS A EXTRANJEROS POR SERVICIOS OCASIONALES	25.00
306	01 POR COMPRAS LOCALES DE MATERIA PRIMA	1.00
307	01 SERVICIOS PREDOMINA LA MANO DE OBRA	2.00
308	01 SERVICIOS ENTRE SOCIEDADES	2.00
309	01 POR SERVICIOS PUBLICIDAD Y COMUNICACION	2.75
310	01 TRANSPORTE PRIVADO DE PASAJEROS O SERVICIO PUBLICO O PRIVADO	1.00
311	01 POR LUBRICANTES	2.00
312	01 TRANSFERENCIA DE BIENES MUEBLES DE NATURALEZA CORPORAL	1.75
3120	01 TRANSFERENCIA DE BIENES MUEBLES DE NATURALEZA CORPORAL	1.75
313	01 POR CONCEPTO DE SERVICIO DE TRANSPORTE PRIVADO DE PASAJEROS	1.00
314	01 POR REGALIAS, DERECHOS DE AUTOR, MARCAS, PATENTES Y SIMILARE	8.00
315	01 POR REMUNERACIONES A DEPORTISTAS, ENTRENADORES, CUERPO TECNI	8.00
316	01 POR PAGOS REALIZADOS A NOTARIOS Y REGISTRADORES DE LA PROPIE	8.00
317	01 POR COMISIONES PAGADAS A SOCIEDADES	1.00
318	01 POR PROMOCION Y PUBLICIDAD	2.00
319	01 POR ARRENDAMIENTO MERCANTIL LOCAL	2.00
320	01 POR ARRENDAMIENTO DE BIENES INMUEBLES	10.00
321	01 POR ARRENDAMIENTO DE BIENES INMUEBLES A SOCIEDADES	8.00
322	01 POR SEGUROS Y REASEGUROS (10 % DEL VALOR DE LAS PRIMAS FACTU	1.00
323	01 POR RENDIMIENTOS FINANCIEROS (NO APLICA PARA IFI'S)	2.00

F1-Ayuda F2-Agregar F3-Insertar F4-Buscar F5-Imprimir F6-Mas funciones

Advertencia: Revision Review 5.0

Figura 24 Códigos de retenciones sistema antiguo. Fuente: Autores



```
GET http://localhost:4000/api/retencioncode/
Params Authorization Headers (6) Body Scripts Tests Settings
Body Cookies Headers (9) Test Results 200 OK · 45 ms · 24.38 KB
Pretty Raw Preview Visualize JSON
58 {
59   "_id": "670f3c434d813424f6ef1da5",
60   "type": "1",
61   "descripcion": "Honorarios profesionales y demás pagos por servicios relacionados con el título profesional",
62   "codigo": "303",
63   "porcentaje": 10,
64   "__v": 0
65 },
66 {
67   "_id": "670f3c434d813424f6ef1da7",
68   "type": "1",
69   "descripcion": "Servicios profesionales prestados por sociedades residentes",
70   "codigo": "303A",
71   "porcentaje": 3,
72   "__v": 0
73 },
74 {
75   "_id": "670f3c434d813424f6ef1da9",
76   "type": "1",
77   "descripcion": "Servicios predomina el intelecto no relacionados con el título profesional",
78   "codigo": "304",
79   "porcentaje": 10,
80   "__v": 0
81 },
82 {
83   "_id": "670f3c434d813424f6ef1dab",
84   "type": "1",
```

Figura 25 Códigos de retención nuevo sistema. Fuente: Autores

5.4 EVALUACIÓN DE LOS MÓDULOS

Para la evaluación del desarrollo de los módulos se detectaron dos puntos cruciales que definieron la aprobación del sistema. Las pruebas de usabilidad realizadas por los usuarios del sistema, y la seguridad aportada a cada uno de los módulos son factores que hicieron que el sistema sea confiable y funcione acorde a la normativa contable del país, por lo que se describen los siguientes resultados:

5.4.1 FUNCIONES DE SEGURIDAD.

El sistema contable desarrollado para Accescont incorpora varias medidas de seguridad clave para proteger la información y garantizar un funcionamiento seguro. Estas medidas se diseñaron pensando en la protección de datos sensibles y en el cumplimiento de las normativas fiscales y empresariales. Las principales funciones de seguridad incluyen:

Autenticación y Autorización: El acceso a los distintos módulos del sistema está controlado mediante un sistema de autenticación basado en tokens. Esto asegura que solo los usuarios autenticados pueden acceder a las funciones autorizadas

Gestión Segura de las Configuraciones: Las credenciales sensibles, como las contraseñas y las claves de los certificados de firma electrónica, se gestionan a

través de variables de entorno, evitando que queden expuestas en el código fuente. Esta práctica reduce significativamente el riesgo de acceso no autorizado a la información crítica.

Encriptación de Datos: Para garantizar la seguridad en las comunicaciones entre los diferentes microservicios, se utilizan protocolos de encriptación SSL/TLS. Esto protege los datos mientras se intercambian entre los módulos del sistema, particularmente aquellos relacionados con las facturas y las retenciones, asegurando que la información esté siempre protegida.

Uso de Contenedores Seguros: Cada microservicio está desplegado dentro de un contenedor Docker, lo que proporciona un entorno aislado y controlado para cada servicio. Esto no solo mejora la seguridad al aislar los procesos, sino que también facilita la identificación y resolución de vulnerabilidades específicas en cada servicio.

Monitoreo Continuo de Vulnerabilidades: El proyecto está integrado con herramientas de seguridad que permiten monitorear continuamente las dependencias y detectar posibles vulnerabilidades. Esta supervisión constante permite al equipo tomar medidas rápidas para corregir cualquier problema de seguridad, garantizando que el sistema esté siempre protegido.

5.4.2 SATISFACCIÓN DE LOS USUARIOS:

Las pruebas de usabilidad arrojaron resultados positivos, mostrando una mejora notable en la experiencia del usuario final. Los usuarios destacaron una mayor facilidad en la navegación por el sistema, así como un incremento en la productividad del personal que utiliza el sistema para las gestiones diarias. Estas mejoras reflejan un diseño más intuitivo y eficiente, que responde mejor a las necesidades del negocio y de sus usuarios (Figura 26).

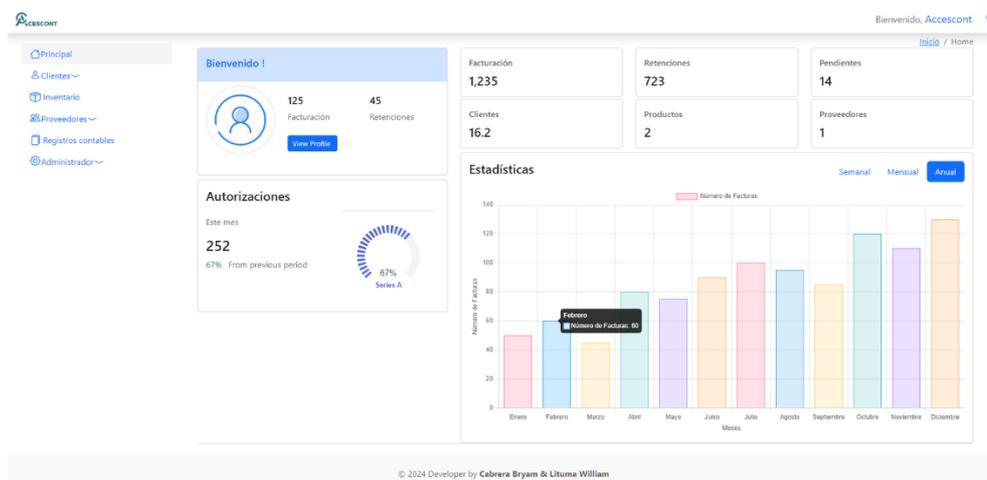


Figura 26 Interfaz de usuario final. Fuente: Autores

El sistema contable basado en microservicios no solo ha alcanzado los objetivos de escalabilidad y modularidad descritos inicialmente, sino que también ha facilitado una mejor comprensión de la estructura global del sistema, resolviendo eficazmente el deterioro y la deuda técnica del código del sistema anterior. La arquitectura modular ha permitido una integración ágil de nuevos servicios, adaptándose rápidamente a las necesidades del negocio y asegurando la flexibilidad para futuras expansiones. Además, se ha logrado con éxito el cumplimiento normativo del Servicio de Rentas Internas (SRI), con la correcta autorización de documentos gracias a la implementación precisa de los estándares exigidos, como la generación de facturas y retenciones en formato XML firmado digitalmente.

No obstante, un desafío considerable fue la creación del microservicio firmador, debido a las limitaciones tecnológicas del sistema del SRI, que aún opera con un modelo tecnológico complejo. Esto ha complicado la adopción de nuevas tecnologías más eficientes. A pesar de estas dificultades, se logró implementar una solución que respeta el esquema de documentos requerido por el SRI, garantizando la interoperabilidad del sistema, aunque esto señala la posibilidad de mejorar la integración a largo plazo.

Mirando hacia el futuro, se planea la implementación de módulos adicionales, como los de asientos contables y gestión de cuentas, los cuales contribuirán significativamente a la optimización de la gestión contable de la empresa. Estos

desarrollos permitirán automatizar procesos clave, disminuyendo la carga manual y aumentando la eficiencia del departamento contable, alineándose con las proyecciones de crecimiento y mejora continua de la empresa.

6 CONCLUSIONES

En la fase inicial del proyecto, Scrum permitió una estrecha colaboración entre los equipos de desarrollo y los usuarios finales, quienes proporcionaron los requisitos esenciales para los módulos de clientes y proveedores del sistema contable. Mediante la retroalimentación continua en las reuniones de Sprint Reviews, los usuarios pudieron validar los avances y ajustar los requisitos a medida que surgían nuevas necesidades o cambios en las normativas fiscales. Este proceso aseguró que los módulos desarrollados respondieran a las necesidades reales del negocio y que la integración con el sistema contable fuera fluida y efectiva.

El sistema desarrollado basado en una arquitectura de microservicios para la empresa Accescont ha alcanzado los objetivos planteados, logrando mejorar la escalabilidad, flexibilidad y eficiencia del sistema, gracias a dicha arquitectura y la modularidad actual del sistema es posible integrar nuevos servicios con facilidad, reduciendo la degradación del código y la obsolescencia del sistema anterior. Por otro lado, la implementación de las normas del SRI desde la creación del código de facturación y retenciones, asegura el cumplimiento de las normativas tributarias exigidas por dicho ente regulatorio, lo que facilita la autorización de documentos electrónicos sin problemas.

En el desarrollo del nuevo sistema se presentaron algunos retos como la antigüedad del formato de recepción de documentos por parte del SRI y junto con el formato rígido que tiene que seguir para la autorización de documentos hace que algunas tareas se dificulten, no obstante, al seguir el esquema de los documentos que el SRI propone se puede llegar al objetivo sin complicaciones.

La migración exitosa de datos desde el antiguo sistema monolítico, fue complejo debido a la diferencia de esquemas de bases de datos usados en el antiguo sistema y en el nuevo, precisamente porque las relaciones de una base SQL son complicadas de representar en un sistema con bases no NoSQL, sin embargo al usar solo lo necesario de las base de datos antigua y al tratamiento de datos posterior se logró

migrar los datos correctamente hasta la nueva base de datos, esto junto con la utilización de metodologías ágiles, permitió un desarrollo adaptado a las necesidades del usuario final, facilitando entregas incrementales en el marco SCRUM.

Las contribuciones que tuvo este proyecto al negocio fueron significativas ya que han mejorado el rendimiento de sus colaboradores, los tiempos muertos en el negocio se detectaron en diferentes áreas como en el envío de facturas con un proceso lento y poco dinámico, por lo que al usar el sistema actual la reducción de tiempos de espera y fluides en el llenado de facturas y acceso a la información hacen que se reduzca el trabajo que se tiene que realizar, aumentando la satisfacción de los usuarios. También la seguridad en los datos de los clientes es muy importante, por lo que las capas de seguridad agregadas al sistema, como la restricción de acceso a estos datos, y la limitación de usuarios que pudiesen tener acceso a esa información es importante según la ley de protección de datos personales que rige en el país.

En el desarrollo de este proyecto se encontraron ciertas limitaciones, como la disponibilidad del SRI para la autorización de documentos, por lo que se deben establecer medidas de guardado temporal por indisponibilidad del sistema del SRI, además se requirió de capacitación a los usuarios y un tiempo de adaptación al nuevo sistema lo que retrasó su uso como sistema principal.

Para el futuro los objetivos del negocio contemplan implementar los módulos de manejo de asientos contables y de gestión de cuentas, por lo tanto, con el esquema actual se reduce significativamente el trabajo a futuro, por lo que, con estos módulos la eficiencia operativa del departamento financiero aumentaría y contribuiría los objetivos del negocio.

7 GLOSARIO

API (Application Programming Interface): Interfaz de programación que permite la comunicación entre diferentes aplicaciones, facilitando la integración de servicios.

Asientos Contables: Registros en los libros de contabilidad que reflejan las transacciones financieras de una empresa.

Base de Datos Relacional (SQL): Tipo de base de datos que organiza los datos en tablas relacionadas y utiliza lenguaje SQL (Structured Query Language) para su gestión.

Base de Datos NoSQL: Tipo de base de datos que permite almacenar y recuperar datos de manera más flexible que una base de datos relacional, como MongoDB.

CRUD (Create, Read, Update, Delete): Operaciones básicas de manipulación de datos en un sistema informático.

Docker: Herramienta que permite "contenedizar" aplicaciones, es decir, empaquetarlas junto con sus dependencias para que funcionen de manera idéntica en cualquier entorno.

ETL (Extract, Transform, Load): Proceso de extracción, transformación y carga de datos en un sistema. Se utiliza comúnmente en la migración de datos entre sistemas.

Facturación Electrónica: Proceso de emisión de facturas en formato digital, cumpliendo con las normativas establecidas por el SRI (Servicio de Rentas Internas).

Firmador Digital: Microservicio encargado de aplicar firmas electrónicas a documentos para garantizar su autenticidad e integridad.

MongoDB: Base de datos NoSQL orientada a documentos, útil para manejar grandes volúmenes de datos en formato flexible como JSON.

Normativa del SRI (Servicio de Rentas Internas): Conjunto de regulaciones fiscales de Ecuador que rigen la emisión de facturas electrónicas y la declaración de retenciones e impuestos.

Retenciones: Deducciones de impuestos que las empresas deben realizar a ciertos pagos, como salarios o servicios, y reportar al SRI.

Scrum: Metodología ágil de desarrollo de software que divide los proyectos en ciclos cortos llamados "sprints", con revisiones periódicas y entregas incrementales.

SOAP (Simple Object Access Protocol): Protocolo estándar utilizado para el intercambio de información estructurada en formato XML entre aplicaciones.

Sprint: Ciclo de trabajo en Scrum que tiene una duración fija (generalmente entre 1 y 4 semanas) y al final del cual se entrega un incremento funcional del producto.

SQL (Structured Query Language): Lenguaje de consulta estructurado utilizado para gestionar y manipular bases de datos relacionales.

Token JWT (JSON Web Token): Estándar abierto que se utiliza para transmitir de forma segura información entre dos partes, en formato JSON, y se usa comúnmente para la autenticación en sistemas web.

XML (Extensible Markup Language): Lenguaje de marcado utilizado para almacenar y transportar datos estructurados de manera que puedan ser fácilmente procesados por máquinas, comúnmente empleado en la facturación electrónica.

8 REFERENCIAS

- Amaral, M., Polo, J., Carrera, D., Mohamed, I., Unuvar, M., & Steinder, M. (2016). Performance evaluation of microservices architectures using containers. *Proceedings - 2015 IEEE 14th International Symposium on Network Computing and Applications, NCA 2015*, 27-34. <https://doi.org/10.1109/NCA.2015.49>
- Brown, E. (2019). *Web Development with Node and Express*.
- Chodorow, K. (2013). *MongoDB-The-Definitive-Guide-2nd-Edition*.
- Fowler, M. (2019). *Patterns of Enterprise Application Architecture*.
- Gelinas, U. J. ., Dull, R. B. ., Wheeler, P. R. ., & Hill, M. Callahan. (2018). *Accounting information systems*. 716.
- Hernández-Salazar, E., & Beltrán, C. A. (2020). *Un enfoque práctico de metodología ágil para la ingeniería de software A practical agile methodology approach to software engineering*.
- Juca Banchón, Y. C., & Moreno Narváez, V. P. (2024). La evolución de la contabilidad en la era digital: La profesión contable en el Ecuador. *Pacha. Revista de Estudios Contemporáneos del Sur Global*, 5(16), e240287. <https://doi.org/10.46652/pacha.v5i16.287>
- Li, Y., Lin, Y., Wang, Y., Ye, K., & Xu, C. (2023). Serverless Computing: State-of-the-Art, Challenges and Opportunities. *IEEE Transactions on Services Computing*, 16(2), 1522-1539. <https://doi.org/10.1109/TSC.2022.3166553>
- Martin, R. C. . (2014). *Agile software development, principles, patterns, and practices*. Pearson.
- Merkel, D. (2014). *Docker: lightweight Linux containers for consistent development and deployment*. https://www.researchgate.net/publication/261960832_Docker_lightweight_Linux_containers_for_consistent_development_and_deployment
- Molina Montero, B., Vite Cevallos, H., & Dávila Cuesta, J. (2018). *Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software*.
- Newman, S. (2015). *Building Microservices: Designing Fine-Grained Systems - Sam Newman - Google Libros*. [https://books.google.com.ec/books?hl=es&lr=&id=ZvM5EAAAQBAJ&oi=fnd&pg=PT8&dq=Newman,+S.+\(2015\).+Building+Microservices:+Designing+Fine-Grained+Systems.+O%27Reilly+Media.&ots=ui6naz5GUI&sig=lr4iFc0nKWxP6pZO3OzTV-OWUDM#v=onepage&q&f=false](https://books.google.com.ec/books?hl=es&lr=&id=ZvM5EAAAQBAJ&oi=fnd&pg=PT8&dq=Newman,+S.+(2015).+Building+Microservices:+Designing+Fine-Grained+Systems.+O%27Reilly+Media.&ots=ui6naz5GUI&sig=lr4iFc0nKWxP6pZO3OzTV-OWUDM#v=onepage&q&f=false)
- Richards, M. (2015). *Additional Resources 4 Easy Ways to Learn More and Stay Current Programming Newsletter*.
- Richardson, C. (2018). *Microservices Patterns: With examples in Java - Chris Richardson - Google Libros*. [https://books.google.com.ec/books?hl=es&lr=&id=QTgzEAAAQBAJ&oi=fnd&pg=P T21&dq=Richardson,+C.+\(2018\).+Microservices+Patterns:+With+Examples+in+Java.+Manning+Publications.&ots=95ca2BNHAc&sig=F4RFqMPpy7I7XvGUgOuOMI_P IFs#v=onepage&q=Richardson%2C%20\(2018\).%20Microservices%20Pattern s%3A%20With%20Examples%20in%20Java.%20Manning%20Publications.&f=false](https://books.google.com.ec/books?hl=es&lr=&id=QTgzEAAAQBAJ&oi=fnd&pg=P T21&dq=Richardson,+C.+(2018).+Microservices+Patterns:+With+Examples+in+Java.+Manning+Publications.&ots=95ca2BNHAc&sig=F4RFqMPpy7I7XvGUgOuOMI_P IFs#v=onepage&q=Richardson%2C%20(2018).%20Microservices%20Pattern s%3A%20With%20Examples%20in%20Java.%20Manning%20Publications.&f=false)

- Rodríguez González, P. (2008). *FACULTAD DE INFORMÁTICA ESTUDIO DE LA APLICACIÓN DE METODOLOGÍAS ÁGILES PARA LA EVOLUCIÓN DE PRODUCTOS SOFTWARE*.
- Schwaber, K. (2004). *Agile Project Management with Scrum*. <http://www.bisenter.com>
- Schwaber, K., & Sutherland, J. (2017). *The Scrum Guide™ The Definitive Guide to Scrum: The Rules of the Game*.
- Serna, E. (2021). *Dialnet-MetodosFormalesIngenieriaDeRequisitosYPruebasDelSo-877256*.
- Sohaib, O., & Khan, K. (2010). Integrating usability engineering and agile software development: A literature review. *2010 International Conference on Computer Design and Applications, ICCDA 2010, 2*.
<https://doi.org/10.1109/ICCDA.2010.5540916>
- Sommerville, Ian. (2016). *Software engineering* (10.^a ed.). Pearson.
- SRI. (2024a). *Guía para contribuyentes*.
- SRI. (2024b). *Normativa para agentes de retención y contribuyentes especiales - intersri - Servicio de Rentas Internas*. <https://www.sri.gob.ec/normativa-para-agentes-de-retencion-y-contribuyentes-especiales>
- SRI. (2024c). *REGLAMENTO DE COMPROBANTES DE VENTA RETENCI_N Y DOCUMENTOS COMPLEMENTARIOS*.
- Sutherland, J. (2015). *El arte de hacer el doble de trabajo en la mitad de tiempo*.
<https://assets.super.so/e745dfd7-232e-47c1-a806-753f45b941ba/files/e5c9770b-4e7d-4df6-880f-a9a15ace7b51.pdf>
- Theodoropoulos, T., Rosa, L., Benzaid, C., Gray, P., Marin, E., Makris, A., Cordeiro, L., Diego, F., Sorokin, P., Girolamo, M. Di, Barone, P., Taleb, T., & Tserpes, K. (2023). Security in Cloud-Native Services: A Survey. *Journal of Cybersecurity and Privacy 2023, Vol. 3, Pages 758-793, 3(4)*, 758-793. <https://doi.org/10.3390/JCP3040034>
- Thesing, T., Feldmann, C., & Burchardt, M. (2021). Agile versus Waterfall Project Management: Decision Model for Selecting the Appropriate Approach to a Project. *Procedia Computer Science, 181*, 746-756.
<https://doi.org/10.1016/J.PROCS.2021.01.227>
- Wong Durand, S. (2017). *Análisis y requerimientos de software*. 128.
https://repositorio.continental.edu.pe/bitstream/20.500.12394/4281/1/DO_FIN_103_MAI_UC0939_2018.pdf

9 ANEXOS

9.1 FORMULARIO DE LEVANTAMIENTO DE REQUISITOS

Formulario para Levantamiento de Casos de Uso y Requisitos

Nombre del Proyecto: Análisis, desarrollo y migración de los módulos de proveedores y clientes del sistema contable de la empresa Accescont para agilizar los procesos relacionados al manejo de la facturación para el departamento financiero de la empresa.

Sprint No.:

Fecha:

Product Owner: Ing. Enriqueta Sarmiento.

Equipo de Desarrollo: Ing. Bryam Cabrera, Ing. William Lituma

1. Historia de Usuario

- **Título:**

- **Descripción:**

Como _____, quiero

, para _____

- **Criterios de Aceptación:**

-

-

-

2. Requerimientos Funcionales

- **Descripción del Requerimiento:**

- **Prioridad:**

Alta

Media

Baja

- **Criterios de Éxito:**

-

-
-

3. Requerimientos No Funcionales

- _____
- _____
- _____
- _____

4. Criterios de Finalización

- Revisión del Product Owner
- Pruebas unitarias completadas
- Pruebas de integración completadas
- Funcionalidad desplegada correctamente

9.2 HISTORIAS DE USUARIO

Gestión de Clientes y Proveedores:

Como contador, quiero filtrar y ver clientes y proveedores según su tipo, para encontrar rápidamente la información y agilizar el proceso de búsqueda.

Como contador, necesito modificar la información de los clientes y proveedores para mantener actualizados los datos de contacto y dirección, garantizando la precisión de la base de datos.

Como administrador, quiero poder eliminar clientes y proveedores que ya no se utilizan o están inactivos, manteniendo la base de datos limpia y ordenada para evitar confusiones.

Como contador, quiero registrar nuevos clientes y proveedores, incluyendo su información fiscal, para asegurar que estén listos en el sistema para emitirles facturas.

Como contador, deseo poder filtrar clientes por RUC o razón social, facilitando la búsqueda rápida de clientes específicos.

Como contador, quiero actualizar la información de clientes existentes para garantizar que los datos fiscales y de contacto estén siempre actualizados.

Como administrador, quiero eliminar clientes inactivos, para asegurar que la base de datos esté siempre precisa y organizada.

Validación de la Información:

Como contador, necesito que el sistema valide automáticamente el formato del RUC y otros datos fiscales, para evitar errores durante la entrada de información.

Como administrador, quiero que el sistema impida la duplicación de clientes por RUC o cédula, garantizando la integridad de la base de datos.

Gestión de Facturas:

Como contador, quiero registrar facturas emitidas, incluyendo detalles del cliente, productos o servicios, impuestos aplicables y descuentos, para mantener un registro completo de todas las transacciones.

Como contador, deseo ver un historial de todas las facturas emitidas a cada cliente, para tener un control claro de las transacciones previas.

Como contador, quiero enviar las facturas electrónicas al SRI, asegurando el cumplimiento de las normativas fiscales.

Como contador, necesito adjuntar archivos adicionales, como contratos o guías de remisión, a las facturas para tener toda la documentación relevante en un solo lugar.

Gestión de Retenciones:

Como contador, quiero adjuntar archivos relevantes a las facturas registradas, para tener un respaldo adicional y facilitar la revisión de documentos relacionados.

Como contador, deseo recibir alertas de vencimiento sobre facturas pendientes para evitar retrasos en los pagos y cumplir con los plazos establecidos.

Como contador, necesito guardar las facturas emitidas en una base de datos local, para tener acceso a ellas incluso cuando el SRI no esté disponible.

Registro y Seguimiento de Facturas y Retenciones:

Como contador, quiero registrar las facturas emitidas y poder buscarlas por RUC, para hacer un seguimiento adecuado de los pagos.

Como usuario, necesito modificar los campos de retención para mantenerlos actualizados según las normativas nacionales, y así evitar inconsistencias en la contabilidad.

Historial Financiero y Tributario:

Como administrador, deseo visualizar el historial de retenciones aplicadas a cada proveedor, para asegurar la coherencia y precisión en la gestión tributaria.

Como administrador, quiero ver el historial de facturas emitidas a cada cliente, para tener una visión clara de las retenciones aplicadas y garantizar una gestión fiscal coherente.

Como usuario, quiero que el sistema valide automáticamente las normativas fiscales vigentes al emitir retenciones, garantizando el cumplimiento legal.

Como administrador, necesito registros detallados de auditoría para revisar y verificar todas las operaciones relacionadas con las retenciones.

Como administrador, quiero que el sistema genere automáticamente los XML de las facturas y los envíe al SRI, asegurando el cumplimiento fiscal.

Conciliación y Reportes Contables:

Como contador, necesito una función de conciliación automática de asientos contables relacionados con proveedores, para verificar la coherencia entre los registros financieros y las transacciones reales.

Como administrador, deseo generar reportes contables específicos sobre proveedores, como salidas de capital y retenciones de IVA, para facilitar la presentación de informes financieros.

Como administrador, quiero integrar los registros de activos fijos relacionados con transacciones de proveedores, para mantener un control detallado de los activos adquiridos.

Gestión de Pagos y Recordatorios:

Como contador, quiero priorizar las facturas pendientes según la fecha de vencimiento, para asegurarme de atender primero las que están por vencer y evitar sanciones por retrasos.

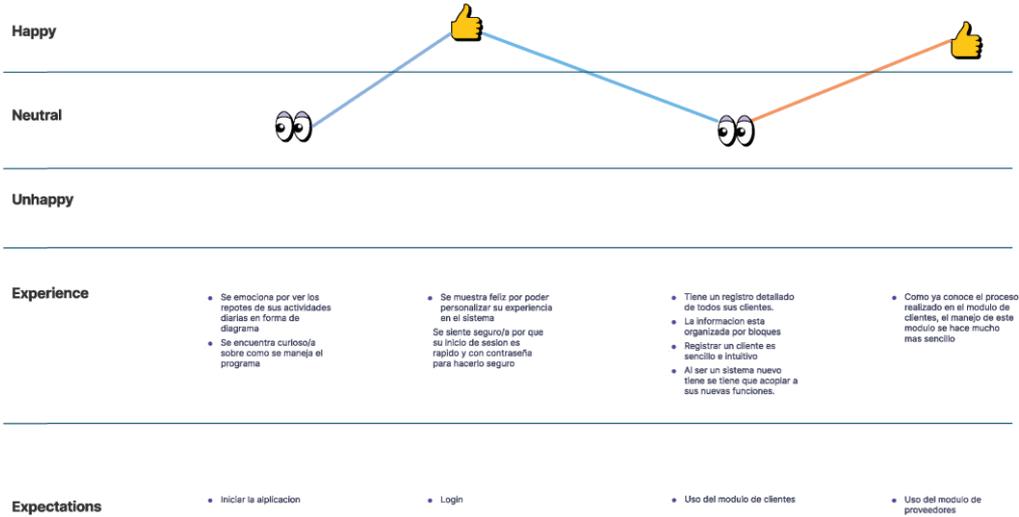
Como contador, deseo recibir notificaciones automáticas sobre las facturas próximas a vencer, para asegurar que los pagos se realicen puntualmente.

Como contador, necesito gestionar disputas o reclamaciones sobre facturas pendientes, para resolver problemas con los proveedores antes de realizar el pago.

9.3 ANÁLISIS UX/UI



Journey Map



C
ACCESCONT

Home

Clientes

Proveedores

Registros Contables

Registro de Clientes

Registro de documentos

Facturar

9.4 DESARROLLO MODULAR



Principal

Clientes

Lista de Clientes

Facturación

Inventario

Proveedores

Lista de Proveedores

Retenciones

Registros contables

Administrador

Ajustes

Usuarios

9.5 LOGIN



Sistema Contable



Ingresar

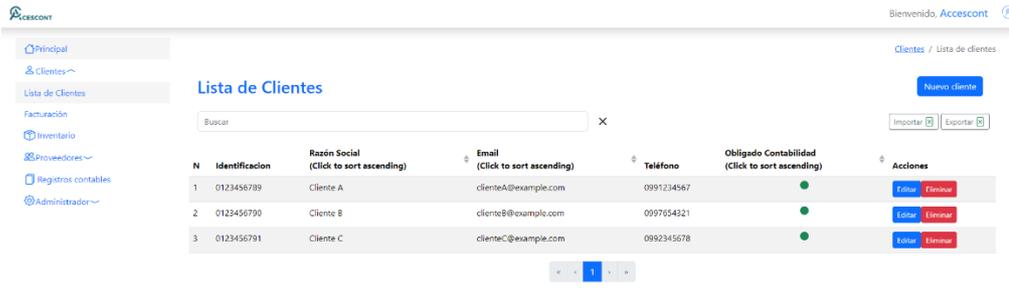
Inicia sesión y comienza a gestionar tus tareas!

Ingrese su correo

Ingrese su contraseña

Ingresar

9.6 MÓDULO CLIENTES

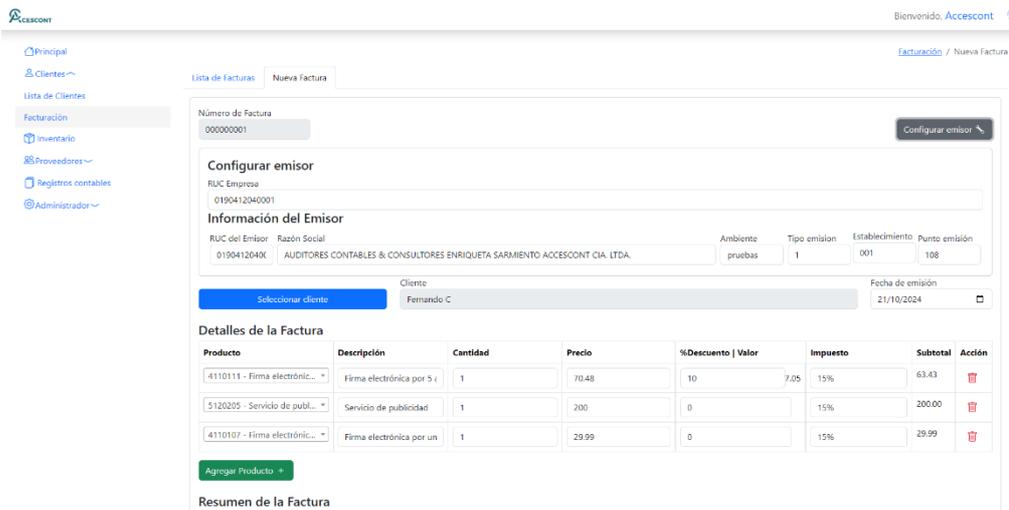


Lista de Clientes

Buscar X Importar Exportar

N	Identificación	Razón Social (Click to sort ascending)	Email (Click to sort ascending)	Teléfono	Obligado Contabilidad (Click to sort ascending)	Acciones
1	0123456789	Cliente A	clienteA@example.com	0991234567	●	Editar Eliminar
2	0123456790	Cliente B	clienteB@example.com	0997654321	●	Editar Eliminar
3	0123456791	Cliente C	clienteC@example.com	0992345678	●	Editar Eliminar

9.7 FACTURACIÓN ELECTRÓNICA



Configurar emisor

RUC Empresa: 0190412040001

RUC del Emisor: 0190412040001

Razón Social: AUDITORES CONTABLES & CONSULTORES ENRIQUETA SARMIENTO ACCESCONT CIA. LTDA.

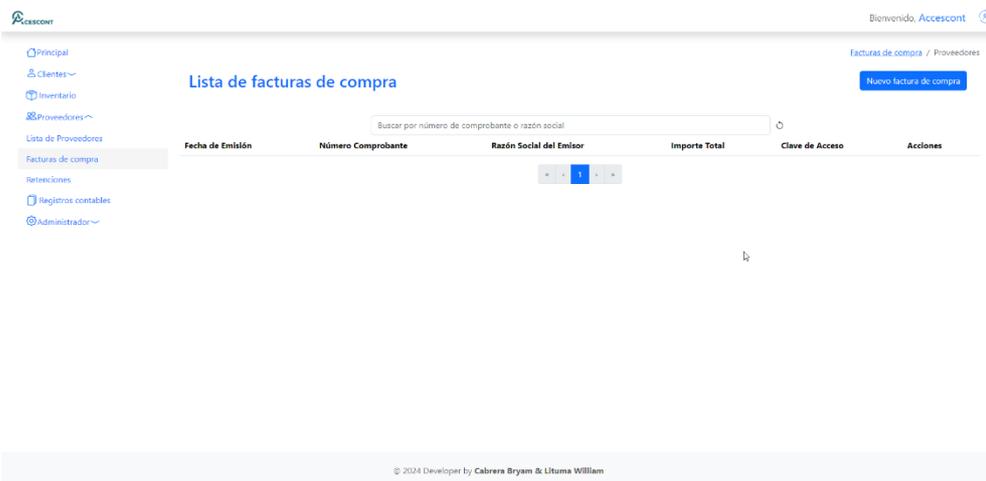
Ambiente: pruebas | Tipo emisión: 1 | Establecimiento: 001 | Punto emisión: 108

Cliente: Fernando C | Fecha de emisión: 21/10/2024

Detalles de la Factura

Producto	Descripción	Cantidad	Precio	%Descuento Valor	Impuesto	Subtotal	Acción
4110111 - Firma electróni...	Firma electrónica por S x	1	70.48	10 7.05	15%	63.43	
5120205 - Servicio de publ...	Servicio de publicidad	1	200	0	15%	200.00	
4110107 - Firma electróni...	Firma electrónica por un	1	29.99	0	15%	29.99	

9.8 MÓDULO DE PROVEEDORES



Lista de facturas de compra

Buscar por número de comprobante o razón social

Fecha de Emisión	Número Comprobante	Razón Social del Emisor	Importe Total	Clave de Acceso	Acciones
------------------	--------------------	-------------------------	---------------	-----------------	----------

Bienvenido, Accescont

Facturas de compra / Proveedores

Lista de facturas de compra

Buscar por número de comprobante o razón social

Fecha de Emisión	Número Comprobante	Razón Social del Emisor	Importe Total	Clave de Acceso	Acciones
16/10/2024	002-202-000404403	SUPERMERCADO Y COMIS. HORTENCIA VARGAS DE JARA E HIJOS LTDA.	\$3.95	161020240106900669710012002202004044031183460111	Retener

✓

Factura agregada

La factura ha sido procesada exitosamente.

© 2024 Developer by Cabrera Bryam & Lituma William

Bienvenido, Accescont

Número de Retención: 00000002 | Fecha de emisión: 23/10/2024

Cargar factura de compra a retener

Seleccionar factura

Seleccionar factura de compra

Buscar por número de comprobante o razón social

Fecha de Emisión	Número Comprobante	Razón Social del Emisor	Importe Total	Clave de Acceso	Acciones
16/10/2024	002-202-000404403	SUPERMERCADO Y COMIS. HORTENCIA VARGAS DE JARA E HIJOS LTDA.	\$3.95	161020240106900669710012002202004044031183460111	Retener

Cancel OK Emitir Retención

© 2024 Developer by Cabrera Bryam & Lituma William

Bienvenido, Accescont

Número de Retención: 00000002 | Fecha de emisión: 23/10/2024

Cargar factura de compra a retener

002-202-000404403

Detalles de la retención

Comprobante	Número	Fecha Emisión	Ejercicio Fiscal	Base Imponible para la retención	Impuesto	Porcentaje de retención	Acción
FACTURA -	002-202-000404403	16-10-2024	10-2024	150	IVA	70	Retener
FACTURA -	002-202-000404403	16-10-2024	10-2024	1000	RENTA	2.75	Retener

Agregar Retención +

Información Adicional

Nombre	Valor	Acciones
Observación	Retención de ejemplo	Retener

Agregar información Adicional +

Enviar retención

© 2024 Developer by Cabrera Bryam & Lituma William

9.9 FORMULARIO DE VALIDACIÓN

Formulario de Pruebas de Funcionalidad del Sistema

Nombre del Proyecto: Análisis, desarrollo y migración de los módulos de proveedores y clientes del sistema contable de la empresa Accescont para agilizar los procesos relacionados al manejo de la facturación para el departamento financiero de la empresa.

Fecha de Prueba:

Nombre del Tester:

1. Funcionalidad General

Funcionalidad probada:

- Registro de clientes
- Modificación de proveedores
- Emisión de facturas
- Emisión de retenciones

Resultados esperados:

Resultados obtenidos:

2. Usabilidad

Tiempo promedio de ejecución de tareas (antes/después de implementación):

Comentarios de los usuarios sobre la interfaz:

3. Seguridad

Implementación de tokens JWT:

Correcta

Incorrecta

Comentarios:

4. Cálculos Fiscales

Cálculos de impuestos y retenciones:

Correctos

Incorrectos

5. Integración con el SRI

Conexión con el SRI:

Funciona

No funciona

Documentos electrónicos autorizados:

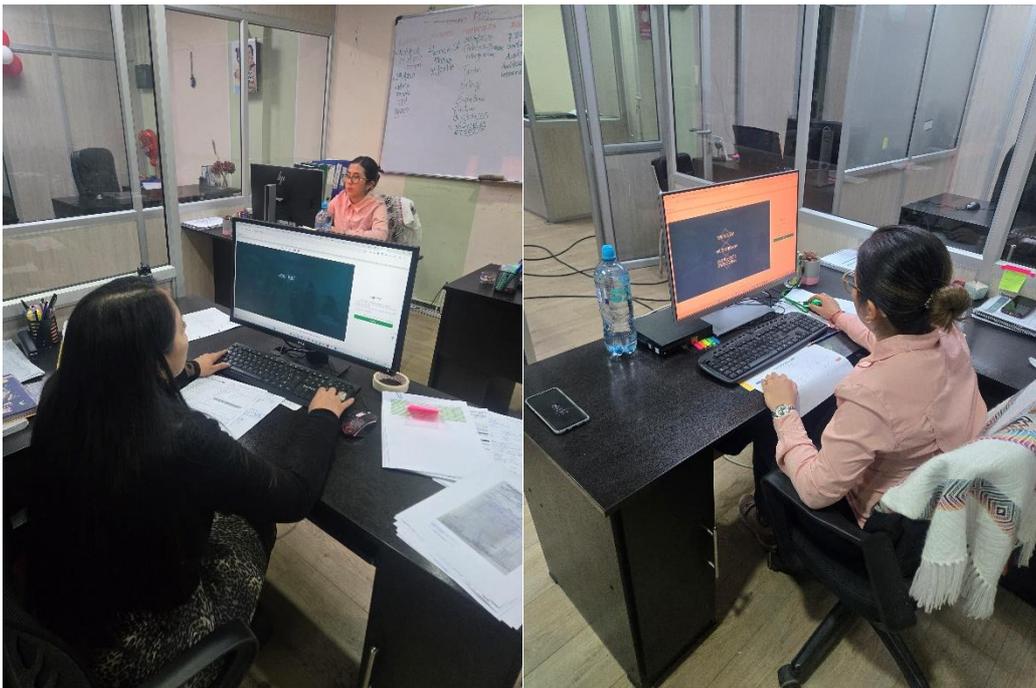
Correctos

Incorrectos

6. Observaciones Generales:

9.10 SATISFACCIÓN DEL PERSONAL DE ACCESCONT DURANTE LAS PRUEBAS DEL SISTEMA

En las imágenes se puede ver al personal administrativo de Accescont participando activamente en las pruebas del sistema. Aunque no son parte del equipo técnico, están revisando y validando que los módulos de **clientes y proveedores** funcionen correctamente desde su perspectiva como usuarios. Están comprobando que todo sea fácil de usar y que las funciones principales, como la gestión de datos y la emisión de comprobantes, se realicen sin problemas. Su colaboración es clave para asegurar que el sistema cumpla con las necesidades diarias del equipo y sea eficiente en el trabajo administrativo.



9.11 REPOSITORIO

El proyecto "**Análisis, desarrollo y migración de los módulos de proveedores y clientes del sistema contable de Accescont**" tiene como objetivo mejorar y agilizar los procesos de facturación del departamento financiero de la empresa. Se han desarrollado módulos clave para la **gestión de clientes y proveedores**, así como la **emisión de facturas y retenciones**, integrados con el SRI para optimizar los tiempos y garantizar el cumplimiento de las normativas fiscales.

El código y la documentación del proyecto están disponibles en el siguiente repositorio de GitHub:

Repositorio: [Tesis - Accescont](#)

Este repositorio contiene todos los microservicios desarrollados, incluyendo la facturación y el firmador de comprobantes, junto con la información técnica para su implementación.