

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DESARROLLO DE MÓDULOS DE APLICATIVOS MÓVILES DE
SISTEMA ERP UTILIZANDO PRÁCTICAS DEVOPS SEGURAS
PARA MANTICORE LABS – MÓDULO DE FACTURACIÓN

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
CIENCIAS DE LA COMPUTACIÓN

ARCINIEGA LEMA JOSÉ DAVID

jose.arciniega@epn.edu.ec

DIRECTOR: ING. ADRIÁN EGÜEZ, MSC.

adrian.eguez@epn.edu.ec

Quito, febrero 2023

CERTIFICACIONES

Yo, José Arciniega declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



José David Arciniega Lema

Certifico que el presente trabajo de integración curricular fue desarrollado por José Arciniega, bajo mi supervisión.



MSc. Vicente Adrián Eguez Sarzosa

DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

JOSÉ DAVID ARCINIEGA LEMA

VICENTE ADRIAN EGUEZ SARSOZA

AGRADECIMIENTO

Agradezco a mi madre, por el esfuerzo puesto en cada día de mi vida, para que no me faltaran las cosas indispensables. Su constancia y empeño en demostrar el amor a su familia, han sido mi motivación para intentar ser una mejor persona. Le agradezco por su incondicional apoyo en mi vida académica, que superaban mis intentos de tener una vida más fácil, pero mucho menos satisfactoria. Le agradezco por todos los sacrificios que tuvo que hacer, para que sus hijos puedan disfrutar de lujos que ella nunca pudo tener. No puedo expresar completamente la gratitud que siento a mi madre con estas palabras, sé que nunca podré igualar ese amor que me da, pero me esforzaré cada día de mi vida para demostrárselo.

Agradezco a mi hermana, por sus consejos y su cariño, fueron suficientes para sacarme de muchos momentos difíciles en los que creí haber dado todo de mí, con su ayuda pude enfrentar de nuevo las adversidades, y con motivación, seguir disfrutando cada momento de esfuerzo y constancia.

Agradezco a mi universidad, me ha brindado los momentos más enriquecedores de mi vida académica, aparte de darme los conocimientos necesarios para crear soluciones técnicas dentro y fuera de la vida laboral, me brindó la oportunidad de conocer compañeros que hicieron más entretenida esta trayectoria, y profesores en los que encontré ejemplos de excelentes profesionales y admirables seres humanos. Sé que puedo contar con muchos de ellos para regresar a decirles que fue un gusto ser su alumno o compañero y que tienen un amigo con el que pueden contar.

Finalmente, agradezco a la vida, por colocarme en los momentos y lugares exactos para conocer a personas maravillosa, gracias a ellos, no solo tuve las oportunidades necesarias para alcanzar todos los logros de mi vida hasta ahora, sino que también fueron quienes influyeron de algún modo para que tome las decisiones adecuadas para llegar hasta este punto de mi vida.

Jose David Arciniega

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
ÍNDICE DE FIGURAS.....	VI
ÍNDICE DE TABLAS.....	VII
RESUMEN.....	VII
ABSTRACT.....	VIII
1.1 Objetivo general	2
1.2 Objetivos específicos	2
1.3 Alcance.....	2
1.4 Marco Teórico.....	2
Términos de Facturación electrónica	2
Lenguajes de programación.....	5
Herramientas de desarrollo	6
Metodologías ágiles.....	9
2 METODOLOGÍA.....	12
2.1 Implementación de SCRUM.....	13
2.2 Implementación de DevOps.....	13
2.3 Configuraciones del Flujo DevOps.....	14
2.4 Requerimientos iniciales.....	15
2.5 Historias de Usuario.....	17
2.6 Prototipo de mediana fidelidad	19
2.7 Modelo de Base de Datos	25
2.8 Implementación/Pruebas de Funcionalidad	27
Sprint 1.....	28
Sprint 2.....	29
Sprint 3.....	32

	Sprint 4.....	34
3	RESULTADOS	37
3.1	Pruebas de usabilidad	37
3.2	Resultados de las Pruebas	39
4	CONCLUSIONES Y RECOMENDACIONES	40
4.1	Conclusiones.....	40
4.2	Recomendaciones	41
5	REFERENCIAS BIBLIOGRÁFICAS	42
6	ANEXOS.....	44
	ANEXO I. REPOSITORIO DIGITAL DEL FRONT-END.....	44
	ANEXO II. REPOSITORIO DIGITAL DEL BACK-END.....	44
	ANEXO III. ENLACE AL PROTOTIPO DE LA BASE DE DATOS	44
	ANEXO IV. PROTOTIPO DE MEDIANA FIDELIDAD	44

ÍNDICE DE FIGURAS

	Figura 2.3.1. Repositorio remoto en GitLab	14
	Figura 2.3.2 Carpeta objetivo como repositorio local	15
	Figura 2.6.1 Diagrama de Interfaces (Vista General)	20
	Figura 2.6.2 Interfaz – Inicio de Sesión.....	21
	Figura 2.6.3 Interfaz – Registro de Usuario	22
	Figura 2.6.4 Interfaz – Menú Principal	23
	Figura 2.6.5 Formulario de registro/actualización de la información del emisor	24
	Figura 2.6.6 Pantalla de catálogo y gestión de facturas.....	25
	Figura 2.7.1 Esquema de la relación de la base de datos Firebase con la aplicación .	26
	Figura 2.7.2 Proyecto de base de datos de Firebase vinculado con la aplicación	26
	Figura 2.7.3 Archivo google-services.json generado por Firebase	27

Figura 2.8.1 Implementación de la interfaz de inicio de sesión	29
Figura 2.8.2 Ejecución de la interfaz de Inicio de Sesión	30
Figura 2.8.3 Implementación de la interfaz de Creación de usuario	30
Figura 2.8.4 Ejecución de la interfaz de creación de usuario	31
Figura 2.8.5 implementación de la interfaz del menú principal	32
Figura 2.8.5 Desarrollo de las actividades que integran los layouts	33
Figura 2.8.6 Implementación de la Interfaz de recuperación de cuenta	34
Figura 2.8.7 Implementación de la Interfaz de verificación de correo.....	35
Figura 2.8.8 Implementación de la Interfaz de eliminación de cuenta	35
Figura 2.8.9 Implementación de la Interfaz de acceso a tu cuenta.....	36
Figura 2.8.9 Implementación de la Interfaz de menú principal.....	36
Ecuación 3.1.1 Fórmula para calcular el puntaje SUS	38
Figura 3.1.2 Promedios por pregunta/ítem de SUS.....	39

ÍNDICE DE TABLAS

Tabla 2.1.1 Resumen de requerimientos consensuados con las partes interesadas ..	13
Tabla 2.4.1 Resumen de requerimientos consensuados con las partes interesadas ..	16
Tabla 2.5.1 Resumen de historias de usuario 001	17
Tabla 2.5.2 Resumen de historias de usuario 002	17
Tabla 2.5.3 Resumen de historias de usuario 003	18
Tabla 2.5.4 Resumen de historias de usuario 004	18
Tabla 2.5.5 Resumen de historias de usuario 005	19
Tabla 3.1.1. Cuestionario SUS.....	37
Tabla 3.1.1. Resultados del cuestionario SUS	38

RESUMEN

El presente trabajo de integración curricular, presenta una aplicación móvil que permite gestionar de forma más eficiente los comprobantes electrónicos, que emita la empresa Manticore Labs en sus procesos comerciales rutinarios. El desarrollo de la aplicación se resolvió utilizando las metodologías DevOps y Scrum, siendo estas, dos de las más utilizadas actualmente como metodologías de desarrollo y gestión. El trabajo presenta la integración de las dos metodologías en el desarrollo de una aplicación móvil utilizando flujos de DevOps seguros, también se muestra el uso de la metodología Scrum y su adaptación en el proyecto. Resaltamos los resultados del uso de los flujos DevOps y la aplicación de la metodología Scrum para verificar la eficiencia de las metodologías en el uso de este trabajo. El propósito de este proyecto es automatizar el proceso de generación, actualización y eliminación de documentos como facturas, guías de remisión, notas de débito, notas de crédito y comprobantes de retención, los cuales reflejan las operaciones llevadas a cabo con clientes, transportistas y productos de una organización.

PALABRAS CLAVE: Facturación electrónica, App de facturación, Manticore Labs

ABSTRACT

The This curricular integration work presents a mobile application that allows to manage more efficiently the electronic receipts issued by the company Manticore

Labs in its routine business processes. The development of the application was solved using DevOps and Scrum methodologies, being these, two of the most currently used as development and management methodologies. The paper presents the integration of the two methodologies in the development of a mobile application using secure DevOps flows, and also shows the use of the Scrum methodology and its adaptation in the project. We highlight the results of the use of DevOps flows and the application of the Scrum methodology to verify the efficiency of the methodologies in the use of this work. The purpose of this project is to automate the process of generating, updating and deleting documents such as invoices, waybills, debit memos, credit memos and retention slips, which reflect the operations carried out with customers, carriers and products of an organization.

KEYWORDS: Electronic invoicing, Invoicing App, Manticore Labs

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El módulo de Facturación es un componente del sistema ERP de Manticore Labs, el cual se encarga de los procesos de generar, visualizar, actualizar y eliminar comprobantes electrónicos. Para mejorar la eficiencia y rapidez en la gestión de información sobre comprobantes de venta, actividades comerciales, movimientos contables y documentos complementarios de una organización, se sugiere desarrollar este componente. Su función será almacenar de forma automática comprobantes electrónicos y datos relacionados, a fin realizar las operaciones comerciales de una manera más ágil.

La guía de contribuyentes del Sistema de Rentas Internas (SRI) Ecuador establece varios tipos de comprobantes, como la Factura, Guía de Remisión, Nota de crédito, Nota de débito y Comprobante de retención de impuestos. Estos documentos son importantes porque contienen información relevante sobre las transacciones monetarias con los clientes, los productos vendidos y los impuestos asociados, así como los transportistas encargados de la entrega y traslado de productos durante la actividad comercial diaria de una organización.

En el proceso de facturación de una organización, es necesario definir y crear distintos tipos de comprobantes, como las facturas y documentos asociados que registran los movimientos contables. Por ejemplo, las notas de crédito y débito reflejan los saldos a favor y en contra de los clientes, mientras que las retenciones indican los impuestos que el cliente debe pagar. Por último, las guías de remisión se utilizan para definir la entrega de los pedidos correspondientes a las facturas emitidas. Por lo tanto, es fundamental que en el área de venta de bienes y servicios de una organización se definan y controlen los procesos involucrados en la generación de facturas.

Para generar comprobantes electrónicos de manera automática y en cumplimiento con la normativa del SRI, se establecerán junto con la organización formatos para cada tipo de documento. Esto permitirá reutilizar plantillas base y crear en masa los comprobantes electrónicos de manera estandarizada. Por lo tanto, este sistema se convierte en una herramienta valiosa para apoyar la gestión de ventas.

1.1 Objetivo general

El proceso de desarrollo del módulo de Facturación abarca la identificación de los requisitos, el diseño de la solución, la implementación de la misma y la realización de pruebas para garantizar su correcto funcionamiento.

1.2 Objetivos específicos

1. Realizar las historias de usuarios para definir los requisitos del módulo de facturación del sistema.
2. Crear bocetos virtuales y esquemas de base de datos para desarrollar la aplicación móvil del módulo de facturación.
3. Desarrollar el módulo de facturación utilizando la metodología SCRUM y varias herramientas de desarrollo de software.
4. Evaluar y comprobar las funciones del módulo de facturación para garantizar la funcionalidad del aplicativo.

1.3 Alcance

La elaboración del módulo de facturación en el sistema ERP incluye etapas de análisis de requisitos, diseño, codificación y pruebas, y se llevará a cabo mediante metodologías ágiles, en colaboración con la empresa Manticore Labs.

1.4 Marco Teórico

Términos de Facturación electrónica

Comprobantes electrónicos

- **Factura:** Se trata de un documento que certifica una venta y está dirigido a empresas o personas naturales que tienen derecho a un crédito tributario y que participan en operaciones de exportación, ya sea para bienes o servicios [1].
- **Liquidación de compra de bienes y prestación de servicios:** Este documento se refiere a un comprobante de venta expedido por sociedades, personas naturales y sucesiones indivisas en el caso de adquisiciones o servicios [1].

- **Nota de crédito:** Se trata de un documento adicional que autoriza a la empresa a cancelar transacciones, aceptar devoluciones y otorgar descuentos o bonificaciones a sus clientes [1].
- **Nota de débito:** Se trata de un documento adicional que autoriza a la empresa a aplicar intereses por mora y recuperar costos y gastos adicionales incurridos por el vendedor después de la emisión del comprobante [1].
- **Guía de remisión:** Las guías de remisión son documentos que sirven para registrar el traslado de bienes o mercancías de un lugar a otro, ya sea dentro del mismo territorio o hacia el extranjero. Estas guías contienen información relevante sobre los productos que se trasladan, como su cantidad, descripción, origen, destino, fecha de envío, entre otros datos [1].

Las guías de remisión son utilizadas por empresas que se dedican al transporte de mercancías, y su emisión suele ser obligatoria para cumplir con las regulaciones tributarias y aduaneras. Además, también son importantes para la gestión de inventarios y el control de stock, ya que permiten llevar un registro de los productos que se están movilizand o en tiempo real [1].

- **Retención:** Este es un tipo de comprobante que es expedido por personas o empresas que actúan como agentes de retención y certifican la retención de impuestos [1].

Sistema de planificación de recursos empresariales (ERP)

Las empresas han utilizado la tecnología como base para desarrollar diversas actividades, como la administración de productos, el contacto con el cliente, la gestión de información y servicios, entre otros. Al principio, las empresas solían utilizar estas herramientas de forma individual, pero con el tiempo se definió un concepto que englobara estos servicios para administrar integralmente varios aspectos dentro de una organización empresarial. Así nació el concepto de Planificación de Recursos Empresariales o ERP en inglés, al menos desde la perspectiva de las tecnologías informáticas [2].

Un ERP es un sistema de software que ayuda a las organizaciones a administrar múltiples procesos y servicios. Este sistema funciona de forma modular, lo que significa que cada uno de los módulos se enfoca en un área de negocios específica para administrar esos servicios. En resumen, un ERP es una herramienta diseñada para ayudar a las organizaciones a gestionar y mejorar sus procesos y servicios de manera

efectiva y eficiente mediante la implementación de diferentes módulos que se adaptan a las diferentes áreas de negocio [2].

No existe una definición exacta para este tipo de sistemas, ya que los módulos que conforman el sistema y la estructura interna de cada módulo pueden variar dependiendo de las necesidades y capacidades de la organización que lo implemente. En otras palabras, la estructura y los módulos de un sistema de este tipo pueden ser personalizados y adaptados según las necesidades específicas de cada organización [3]. En ese sentido, la empresa SAP (Systemanalyse Programmentwicklung) define algunos módulos comunes dentro de dichos sistemas, así:

- Finanzas.
- Abastecimiento y Compras.
- Gestión de Recursos Humanos.
- Ventas.
- Fabricación.
- Cadena de Suministro.
- Servicios.
- Ingeniería de Información más Desarrollo.
- Gestión de Activos Empresariales [3].

Aplicación móvil

Una aplicación móvil, también conocida como "app", es un programa informático diseñado para ser ejecutado en un dispositivo móvil, como un teléfono inteligente o una tableta. Las aplicaciones móviles pueden ser descargadas desde tiendas de aplicaciones en línea, como la App Store de Apple o Google Play para dispositivos Android. Las aplicaciones móviles pueden ser de diferentes tipos y tener diferentes funcionalidades, desde juegos y entretenimiento hasta herramientas de productividad, comunicación, compras en línea y redes sociales [4]. Las aplicaciones móviles se han convertido en una parte integral de la vida cotidiana de muchas personas, permitiendo el acceso a servicios y contenido en cualquier momento y lugar, siempre y cuando se tenga conexión a Internet.

Lenguajes de programación

Kotlin

Kotlin es un lenguaje de programación de código abierto diseñado para el desarrollo de aplicaciones móviles, especialmente dentro del ambiente de desarrollo Android Studio. Aunque es relativamente nuevo, este lenguaje se beneficia de las mejores características de otros lenguajes, como C# y Java, especialmente en lo que se refiere a su sintaxis y funciones internas. Además, Kotlin es capaz de trabajar con diferentes paradigmas de programación, como la Programación Funcional y la Programación Orientada a Objetos, entre otros [5].

XML

XML, que significa Extensible Markup Language, es un tipo de lenguaje que se utiliza para describir diferentes formatos de intercambio de información entre aplicaciones. Esta tecnología permite definir un formato consistente para transmitir información, incluso si se trata de un formato ejecutable, independientemente de la plataforma en la que se encuentre el documento. En otras palabras, XML proporciona un medio para que las aplicaciones intercambien información de una manera estandarizada y confiable [6].

YAML

YAML, también conocido como Yet Another Markup Language, es un lenguaje de serialización de datos que tiene como objetivo principal permitir la configuración de diversos entornos que puedan comprender su sintaxis. De esta manera, se utiliza la configuración para describir los requisitos, opciones y comandos que deben ejecutarse en el entorno donde se va a ejecutar, de acuerdo con lo que el usuario ha descrito en ella. En resumen, YAML proporciona una forma de definir y configurar diferentes entornos y aplicaciones de manera clara y estructurada mediante un lenguaje fácil de entender y usar [7].

Herramientas de desarrollo

GitLab

GitLab es una plataforma que se enfoca en el desarrollo de software utilizando la metodología DevOps. La plataforma se ofrece a través de la nube y proporciona una serie de herramientas necesarias para el correcto funcionamiento de DevOps, principalmente aquellas relacionadas con la Integración Continua y la Entrega Continua [8].

En relación con el trabajo que será mostrado en la sección de Metodología, es necesario describir algunas de las herramientas a ser usadas en el presente trabajo para comprender mejor su ejecución:

- GitLab CI/CD
- GitLab Runners.
- GitLab Repositories [8].

GitLab CI/CD

Utilizando las definiciones previamente mencionadas de Integración Continua y Entrega Continua (CI/CD), GitLab CI/CD provee la capacidad de ejecutar estos procesos de manera automática. La plataforma permite la automatización de los procesos de construcción, pruebas, lanzamiento y monitoreo de cada producto o porción de producto que haya sido entregado a través de su portal [9].

GitLab Runners

Para ejecutar los procesos automatizados dentro de CI/CD se requiere al menos un sistema operativo que permita su ejecución en un nivel básico, dependiendo del producto que se esté desarrollando. Los Runners permiten la ejecución de estos procesos, ya sea dentro de máquinas virtuales proporcionadas por GitLab o en máquinas que pueden ser configuradas por el cliente a través de contenedores, máquinas virtuales u otros medios similares [10].

GitLab Repositories

Para poder ejecutar CI/CD es necesario tener una aplicación y esta debe estar descrita en uno o varios archivos que la representen. Para almacenar estos archivos y permitir

su acceso, GitLab Repositories es un contenedor que puede almacenar diferentes tipos de archivos, principalmente de programación o configuración, que pueden ser accedidos por otros desarrolladores o por otras funciones de la plataforma [11].

Git

Git es una plataforma de control de versiones de código abierto y gratuita, que permite a los usuarios gestionar la configuración de proyectos de software. Proporciona una variedad de herramientas para el desarrollo distribuido, lo que permite la creación de artefactos de software escalables y mantenibles [12].

Figma

Se trata de una herramienta que posibilita la colaboración en el diseño de distintos tipos de figuras o documentos, destacando su funcionalidad para trabajar directamente desde un navegador web. Su versatilidad permite diseñar diversos objetos, siendo uno de ellos la creación de interfaces para programas o aplicaciones de software, lo cual es una de sus funcionalidades más populares [13].

Firebase

Google proporciona una variedad de herramientas para desarrolladores, incluyendo Firebase, una plataforma diseñada para ofrecer servicios y herramientas para desarrolladores de aplicaciones móviles, especialmente para el sistema operativo Android [14].

Firebase proporciona múltiples utilidades, desde bases de datos hasta la capacidad de realizar pruebas de aplicaciones. Esto permite a los desarrolladores aprovechar diversas opciones para ayudar en el ciclo de vida de desarrollo de sus aplicaciones. En este trabajo, se utilizarán dos de las herramientas ofrecidas por Firebase [14].

Cloud Firestore

Cloud Firestore es una base de datos NoSQL que permite almacenar información sin una estructura rígida. En lugar de eso, ofrece flexibilidad al desarrollador para organizar los datos de acuerdo con las necesidades específicas del negocio [15].

Dentro de la estructura de Cloud Firestore, la información es guardada a través de una estructura de tipo jerárquica, donde:

- Lo más bajo son los datos, que representan en la base de datos la información que se desea almacenar.
- Cada uno de los datos es a su vez almacenado dentro de un documento, que representa la entidad a la cuál cada uno de esos conjuntos de datos pertenecen.
- Y en lo más alto, se encuentran las colecciones, que no son más que el conjunto de varios documentos, que permite almacenar a aquellos que, desde el punto de vista de desarrollo, poseen características similares [16].

Firestore App-Distribution

La capacidad de lanzamiento o distribución que ofrece esta herramienta facilita a los desarrolladores la tarea de publicar su aplicación de manera más sencilla. En el contexto de este proyecto, se ha utilizado dicha capacidad para distribuir una aplicación móvil desarrollada para el sistema operativo Android, de manera que se pueda lanzar automáticamente el archivo ejecutable de dicha aplicación a un repositorio [17].

Docker

Docker es una plataforma de uso libre que tiene como objetivo principal permitir que los usuarios desarrollen y ejecuten aplicaciones en un ambiente virtual aislado y emulado del resto de los recursos existentes en la computadora [18].

Docker, permite crear, administrar y ejecutar servicios en entornos dentro del computador, pero aislados de los recursos de este. Esto lo hace por medio de dos recursos, así:

- Imágenes.
- Contenedores [18].

Imagen

Una imagen se puede contemplar como almacenar la información de un sistema operativo o una aplicación en un instante determinado dentro de un recurso, en este caso, un archivo. Esta imagen, a grandes rasgos, guarda todos los archivos que hacen referencia a dicho sistema o aplicación, de modo que pueda ser instalado y usado en otros lugares con las herramientas necesarias [18].

En el caso de Docker, estas imágenes se ejecutan en lo que se conocen como contenedores.

Contenedor

Un contenedor se refiere a un espacio que se crea para alojar y ejecutar una imagen. En términos simples, se podría comparar a una especie de "computadora" virtual aislada del resto del software del equipo local [18].

La principal ventaja de los contenedores es su capacidad para permitir la ejecución de aplicaciones en un entorno controlado sin interrumpir ni interferir con el resto del sistema. Por ejemplo, durante la ejecución de pruebas que podrían dañar la máquina, los contenedores permiten aislar dichas pruebas para proteger el entorno [18].

Android Studio

Existen diversos entornos de desarrollo disponibles en el mercado para programar de manera convencional, pero también hay otros que se han especializado en el desarrollo de aplicaciones para ambientes o propósitos específicos. Uno de estos entornos es Android Studio, que está diseñado específicamente para ayudar a los desarrolladores a crear proyectos de aplicaciones para dispositivos móviles basados en el sistema operativo Android. Android Studio es un entorno de desarrollo que se basa en IntelliJ IDEA y ofrece una amplia gama de características que facilitan la configuración e integración de proyectos en dispositivos móviles Android [19].

Metodologías ágiles

Las metodologías ágiles son un conjunto de enfoques y prácticas utilizadas en el desarrollo de software y otros proyectos complejos para mejorar la eficiencia y la calidad del proceso de desarrollo. Se basan en el trabajo en equipo, la colaboración y la iteración continua para adaptarse a los cambios que puedan surgir durante el desarrollo del proyecto y para lograr un mejor resultado final. Las metodologías ágiles se caracterizan por un enfoque en la entrega de software funcional en ciclos cortos, conocidos como iteraciones o sprints. Estas iteraciones suelen durar de una a cuatro semanas y, al final

de cada una, se presenta una versión del software que puede ser probada y evaluada por el cliente o usuario final.

Las metodologías ágiles también hacen hincapié en la comunicación abierta y frecuente entre los miembros del equipo y con el cliente, y en la adaptación continua del plan de desarrollo en función de los cambios en los requisitos o en las circunstancias del proyecto. Entre las metodologías ágiles más conocidas se encuentran Scrum, Kanban, Extreme Programming (XP), Lean y Crystal. [20].

SCRUM

Scrum es una forma de trabajar que se enfoca en las personas, equipos y organizaciones para crear soluciones adaptativas en contextos complejos, con el objetivo de generar valor. Este enfoque se basa en las metodologías ágiles, las cuales aprovechan los modelos incrementales, la diversidad de habilidades del equipo y la toma de decisiones basada en la experiencia para obtener los mejores resultados posibles. [21].

En este enfoque se aplica un conjunto de principios ágiles para la gestión de proyectos. Según Joseph Monte, estos principios son [22].

- **Satisfacción del cliente** ante sus deseos y necesidades
- **Adaptabilidad** a los cambios de los requerimientos
- Enfoque funcional del **proyecto, producto o servicio**
- **Desarrollo Sostenible**
- **Cooperación** plena entre desarrolladores del proyecto y el negocio
- **Comunicación directa** entre personas
- **Motivación** de los miembros de un equipo
- Enfoque a la **mejora continua**
- **Simplicidad**
- **Autoorganización** de los equipos de trabajo

Por otra parte, para cumplir con estos principios ágiles de SCRUM, se implementa una cultura organizacional cimentada en diferentes valores ágiles. Según Joseph Monte, estos valores son [22].

- **Compromiso** para trabajar en equipo

- **Enfoque** en el problema asumible para el equipo
- **Organización abierta** del equipo de trabajo
- **Respeto** al trabajo propio y ajeno
- **Coraje** para enfrentarse a los retos

Al aplicar esta metodología, se establecen actividades, artefactos y roles que coordinan el ciclo de vida de Scrum. En un proyecto de software con Scrum, las principales actividades que se llevan a cabo son:

- **Sprint:** Son marcos de tiempo iterativos que permiten la ejecución de todo el trabajo necesario para generar valor y lograr la meta del producto [21].
- **Release:** Es una agrupación de Sprint que permite definir puntos de entrega de un producto terminado en la planificación de un proyecto de software [21]. De esta forma, al culminar la ejecución de todos los Sprints que han sido planificados, se lanza un producto final de valor para el cliente

Por otra parte, los principales artefactos utilizados en SCRUM son:

- **Product Backlog:** Constituye la lista de requerimientos que permiten satisfacer las necesidades del cliente y mejorar el producto [21].
- **Sprint Backlog:** Constituye la lista de requerimientos que han sido seleccionadas y extraídas del Product Backlog para un determinado Sprint [21]. Esto permite planificar y priorizar los requerimientos que los desarrolladores deberán implementar durante la ejecución del Sprint

Dentro de los ciclos de trabajo, el trabajo está coordinado y distribuido dentro del equipo de trabajo en base a roles importantes como:

- **Product Owner:** Se encarga de la creación, priorización y comunicación del Product Backlog [21]. permite definir en orden de prioridad y esfuerzo las funcionalidades que se necesita implementar para generar valor al cliente.
- **Scrum Team:** Es el equipo de trabajo SCRUM compuesto por el Product Owner, los desarrolladores y el Scrum Máster. Los desarrolladores se encargan de planificar el Sprint y adaptar el trabajo constantemente hacia el objetivo del producto [21]. De esta forma, los desarrolladores en cada uno de los Sprints

transforman los requerimientos definidos por el Product Owner, en un incremento funcional que genera valor al cliente.

- **Scrum Master:** Se encarga de entrenar, dirigir y apoyar al Scrum Team en el aprendizaje y práctica de la metodología SCRUM [21]. De esta forma, se ofrece una guía constante durante el trabajo de los desarrolladores y el Product Owner sobre la aplicación de valores y principios ágiles en un proyecto.

Según Joseph Monte, un Sprint tiene las siguientes fases principales:

- **Planificación del Sprint:** En esta fase el Scrum Team define las funcionalidades que se incluirán en el Sprint Backlog, es decir, los ítems seleccionados del Product Backlog para el Sprint actual [22]. Luego de este proceso, el equipo de desarrollo define y estima las tareas necesarias para cada una de las funcionalidades.
- **Reuniones diarias:** Esta fase consiste en el seguimiento, control y sincronización diaria de las tareas del equipo de desarrollo [22].
- **Revisión del Sprint:** Esta fase permite al Scrum Team presentar los resultados del Sprints actual al cliente. De esta forma, este puede aceptar o no el producto creado [22].
- **Retrospectiva SCRUM:** Consiste en una reunión final del Sprint donde el SCRUM Máster define con el equipo de desarrollo áreas de mejora en el proceso interno [22]. Esto permite corregir malas prácticas y redireccionar al equipo hacia el objetivo del producto final si es necesario en el inicio del nuevo Sprint.

2 METODOLOGÍA

En la implementación del módulo dentro de una aplicación móvil, se utilizó la metodología DevOps, con un enfoque principal en la integración continua. Mientras tanto, la gestión del proyecto de desarrollo se llevó a cabo utilizando los artefactos, roles y eventos proporcionados por la metodología Scrum. Sin embargo, debido al alcance del proyecto y al equipo de desarrollo seleccionado, no fue posible aplicar todas las prácticas y directrices ofrecidas por estas metodologías en su totalidad. Por lo tanto, es importante describir cómo se aplicaron estas metodologías en el desarrollo del proyecto.

2.1 Implementación de SCRUM

Se han definido los siguientes roles para el equipo de Scrum a fin de aplicar metodologías ágiles en el desarrollo de este proyecto:

Tabla 2.1.1 Resumen de requerimientos consensuados con las partes interesadas

Rol	Persona	Función
Scrum Master	Msc: Adrián Egüez	En su rol de líder del proyecto, estableciendo y revisando los progresos requeridos para alcanzar los objetivos del mismo.
Product Owner	Msc: Adrián Egüez	En calidad de evaluador, garantizando que el resultado final del proyecto proporcione un valor real al cliente, en este caso, la compañía Manticore Labs.
Desarrollador	José Arciniega	Responsable de llevar a cabo las diversas tareas necesarias para la realización del proyecto, que incluyen análisis, diseño e implementación.

2.2 Implementación de DevOps

Durante la implementación de la metodología DevOps, se enfatizó en la característica fundamental de la Integración Continua. Para lograr este objetivo, se emplearon varias herramientas que en conjunto facilitaron la ejecución del flujo de trabajo de DevOps de manera rápida y sencilla. Se utilizó una imagen que describe de manera general el proceso y las actividades involucradas en este trabajo.



Figura 2.2.1. Diagrama de ejecución de DevOps

2.3 Configuraciones del Flujo DevOps

Las configuraciones requeridas para operar en el Flujo DevOps surgen del segundo paso que se indica en la figura 2-1, es decir, en la herramienta Git. Para utilizar Git, es necesario tener un repositorio remoto previamente creado, al cual se le indique dónde se deben guardar los cambios. Este repositorio se encuentra en el Anexo I, que se muestra en la figura 2-2

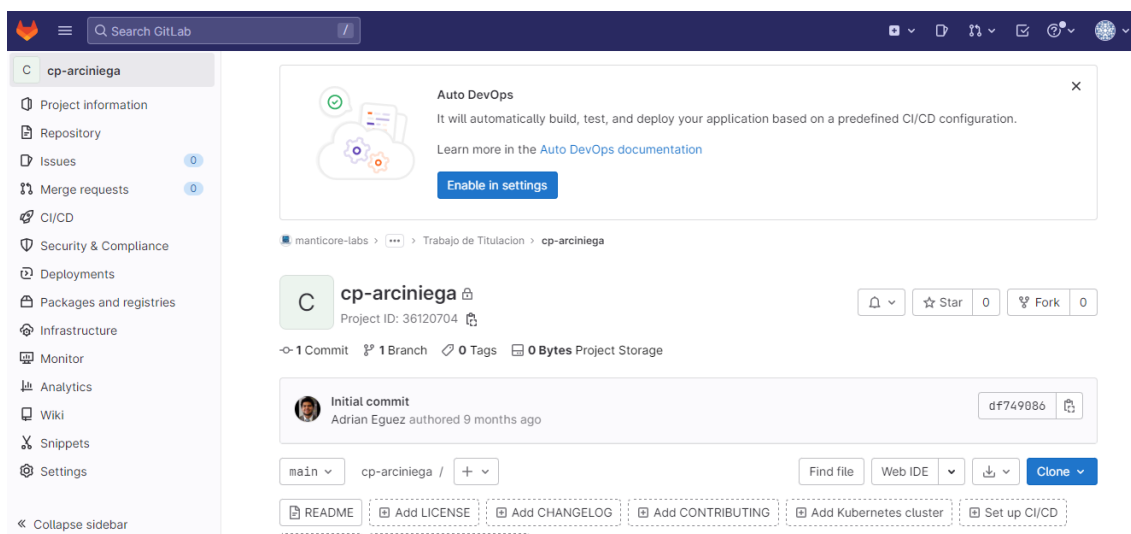


Figura 2.3.1. Repositorio remoto en GitLab

Una vez que el repositorio remoto ha sido creado, es necesario establecer una conexión con un "Repositorio Local", es decir, una carpeta en alguna ubicación dentro del ordenador. En este caso, se optó por utilizar una carpeta llamada "Proyecto", como se puede apreciar en la figura 2.3.

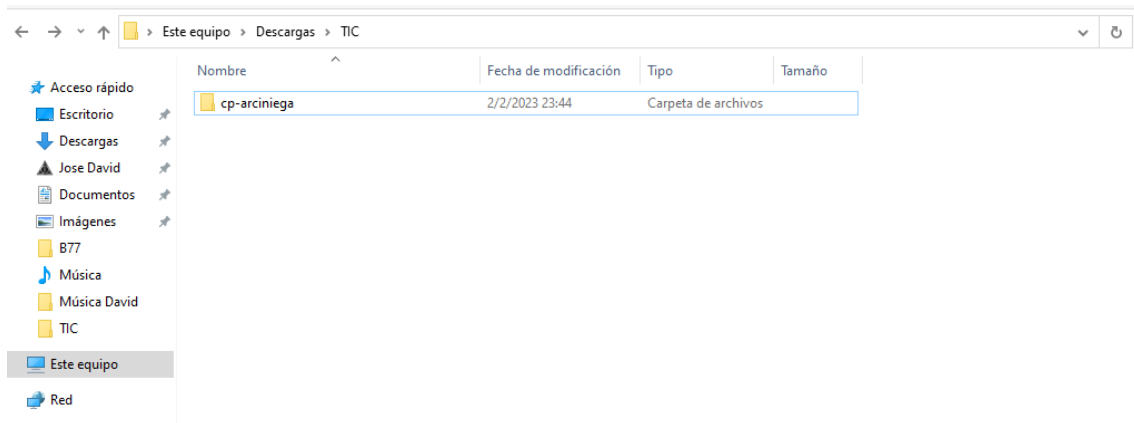


Figura 2.3.2 Carpeta objetivo como repositorio local

2.4 Requerimientos iniciales

Durante la etapa de análisis, se llevó a cabo una revisión informal del problema para identificar las especificaciones requeridas para cumplir con las funcionalidades esperadas del producto final. Los resultados de esta revisión se documentaron en tres artefactos clave que sirvieron como base para la implementación posterior. Estos artefactos fueron fundamentales para la implementación exitosa y constaron de:

- Modelo de Base de Datos (No relacional).
- Modelo de Interfaces.
- Historias de Usuario (Requerimientos)

Se llevaron a cabo reuniones virtuales con el Product Owner utilizando Microsoft Teams para realizar la toma de requerimientos inicial. También se utilizó la técnica de benchmarking para hacer referencia a aplicaciones similares de facturación electrónica del Servicio de Rentas Internas del Ecuador (SRI) y otros. Además, se utilizaron maquetas visuales y modelos de datos que fueron compartidos con el Product Owner. Como resultado de todo esto, se obtuvo un conjunto de requerimientos iniciales que se detallan en la Tabla 2 2.

Tabla 2.4.1 Resumen de requerimientos consensuados con las partes interesadas

Módulo		Requerimientos
Código	Descripción	
USR	Administración de Usuarios	Creación de cuentas de usuario
		Acceso de usuarios al sistema
		Administración de usuarios con rol de administrador.
CLI	Administración de Clientes	Creación, actualización y eliminación de clientes
TRA	Administración de transportistas	Creación, actualización y eliminación de transportistas
PRO	Administración de Productos	Creación, actualización y eliminación de productos
FAC	Administración de Facturas	Visualización de información en formato de factura
		Creación, actualización y deshabilitar de facturas
GDR	Administración de Guías de Remisión	Visualización de información en formato de guía de remisión
		Creación, actualización y deshabilitar de guías de remisión
NDC	Administración de Notas de crédito	Visualización de información en formato de nota de crédito
		Creación, actualización y deshabilitar de notas de crédito
NDD	Administración de Notas de débito	Visualización de información en formato de nota de débito
		Creación, actualización y deshabilitar de notas de débito
RET	Administración de Comprobantes de retención	Visualización de información en formato de comprobante de retención
		Creación, actualización y deshabilitar de comprobantes de retención
EDA	Ejecución de la aplicación	Ejecución de backend
		Ejecución de frontend
BDD	Implementación de Base de datos	Implementación de base de datos por medio de Firebase
CDF	Control de formularios	Validación de campos obligatorios y opcionales en formularios
		Control de tipos de datos y restricciones en valores numéricos

La tabla previa muestra dos aspectos importantes que deben destacarse. Por un lado, se observa que una sola persona ocupó los roles de Scrum Master y Product Owner, lo que también la convierte en un representante de la empresa cliente debido a su participación en esta. Esta situación hace que los eventos del proyecto sean más ágiles, ya que los resultados administrativos y los avances del producto se analizan y revisan por la misma persona. Esto permite que los enfoques administrativos y de desarrollo

apunten en la misma dirección con mayor facilidad. Por otro lado, es relevante mencionar que el equipo de desarrollo está compuesto por una sola persona, lo cual es un factor crucial en ambas metodologías, ya que generalmente se destinan a equipos de trabajo múltiples, donde la colaboración y la interacción entre los miembros del equipo son fundamentales.

2.5 Historias de Usuario

Como se mencionó previamente, en la fase anterior se realizó un análisis para definir los requerimientos necesarios para el proyecto. En este caso, se establecieron los requerimientos a partir de Historias de Usuario. A continuación, se resumirán brevemente los resultados obtenidos de las Historias de Usuario, explicando su propósito y los escenarios contemplados.

Tabla 2.5.1 Resumen de historias de usuario 001

Código	CDF1	Prioridad	Media
Nombre	Formato de campos de información estandarizados		
Descripción	Como usuario, deseo ver entradas de datos genéricas en los formularios de la aplicación para que se mantenga un formato uniforme.		
Antecedentes	Ninguno		
Escenarios			
N°	Nombre	Descripción del escenario	
1	Visualización de campos de texto y numérico	La visualización de los campos de texto y numéricos deben siempre el mismo formato, y este debe ser entendible e intuitivo	
2	Visualización de campos de dirección	El campo de dirección de un formulario debe estar estandarizado, de forma que el ingreso de una dirección sea correcta y entendible.	
3	Visualización de campos de selección y casilla de verificación	Si un formulario incluye uno o varios campos de selección o casillas de verificación, se utilizará un formato uniforme para mostrar dichos campos.	

Tabla 2.5.2 Resumen de historias de usuario 002

Código	CDF2	Prioridad	Media
Nombre	Retroalimentación de formularios		
Descripción	Como usuario, deseo recibir mensajes de error y retroalimentación en los campos del formulario para poder corregir y comprender los errores de validación.		
Antecedentes	Ninguno		
Escenarios			
N°	Nombre	Descripción del escenario	
1	Ingreso de valores vacíos en campos obligatorios	Si se deja algún campo del formulario de la aplicación vacío o se introduce información incorrecta, se sustituye el texto de ayuda correspondiente por un mensaje de error que indica que ese campo es obligatorio.	
2	Ingreso de valores inválidos	Cuando ingreso un valor invalido en alguno de los campos del formulario, entonces se muestra un mensaje de error de acuerdo con el campo afectado.	

Tabla 2.5.3 Resumen de historias de usuario 003

Código	BDD01	Prioridad	Media
Nombre	Implementación de base de datos con Firebase		
Descripción	Como desarrollador, quiero alojar una base de datos NoSQL por medio de Firebase para acceder a ella a través de la web por medio de comandos realizados en Android Studio		
Antecedentes	Crear una cuenta en Firebase y realizar la conexión con la aplicación creada.		
Escenarios			
N°	Nombre	Criterios de aceptación	
1	Revisión de datos en Firebase	<ul style="list-style-type: none"> Se puede acceder a los datos que se registren en Firebase, de tal forma que se crea un registro de las transacciones realizadas. 	
2	Configuración de conexión con Backend	<ul style="list-style-type: none"> Firebase tiene una guía detallada sobre el uso de su base de datos y como conectar con una aplicación móvil. 	
3	Pruebas de conexión	<ul style="list-style-type: none"> La aplicación backend debe traer datos de prueba desde la base de datos de Firebase. 	

Tabla 2.5.4 Resumen de historias de usuario 004

Código	USR1	Prioridad	Media
Nombre	Ingreso del usuario al sistema		
Descripción	Como usuario registrado, deseo poder ingresar al sistema utilizando mi dirección de correo electrónico y una contraseña, para poder acceder a todas las funcionalidades del sistema.		
Antecedentes	Ingresar a la pantalla de Login		
Escenarios			
N°	Nombre	Descripción del escenario	
1	Ingreso exitoso del usuario	Si ingreso correctamente mi correo electrónico y contraseña en la pantalla de inicio de sesión, ingresaré al sistema y se mostrará mi nombre de usuario en la parte superior del menú principal.	
2	Ingreso de correo electrónico o contraseña incorrecto	Si ingreso un correo electrónico y/o contraseña incorrectos en la pantalla de inicio de sesión, se mostrará una notificación de error en la misma pantalla de inicio de sesión.	

Tabla 2.5.5 Resumen de historias de usuario 005

Código	USR2	Prioridad	Media
Nombre	Registro de usuario		
Descripción	Como usuario nuevo quiero registrarme en el sistema para acceder a todas las funcionalidades del sistema.		
Antecedentes	Dar clic en la opción "Regístrate" de la pantalla de Login e ingresar a la pantalla de Registro de usuario.		
Escenarios			
N°	Nombre	Descripción del escenario	
1	Registro exitoso del usuario	Si ingreso mi información personal y una contraseña válida en la pantalla de registro de usuario, y luego hago clic en el botón "Registrar", el sistema me registrará como un nuevo usuario y me redireccionará a la pantalla de inicio de sesión.	
2	Ingreso de confirmación de contraseña no válida	Si ingreso una confirmación de contraseña que no coincide con lo que ingresé en el campo "Contraseña" y hago clic en el botón "Registrar", entonces se mostrará un mensaje de error.	

2.6 Prototipo de mediana fidelidad

Una vez que los requerimientos del aplicativo móvil de facturación fueron definidos conforme al nuevo alcance, se procedió a realizar un diseño de mediana fidelidad mediante la generación de un prototipo de aplicación móvil con la herramienta Figma. En este prototipo se utilizó un diseño que presenta un alto grado de realismo en sus interfaces, similares a las que se utilizarán en el sistema real. También se definió la navegación que el usuario podrá realizar en su experiencia de uso de la aplicación móvil. El enlace al prototipo de la aplicación móvil, del sistema de facturación electrónica se encuentra en el ANEXO IV. PROTOTIPO DE MEDIANA FIDELIDAD.



Figura 2.6.1 Diagrama de Interfaces (Vista General)

A partir de este punto se detallan todas las pantallas que integran la totalidad de las interfaces de la aplicación, y se explica su función.

La primera pantalla, que se muestra en la figura 2.6.2, es la pantalla de inicio de sesión. Esta pantalla funciona como una interfaz para que el usuario pueda autenticarse y autorizarse en la aplicación.



Figura 2.6.2 Interfaz – Inicio de Sesión

En la pantalla de inicio de sesión, aparte de ser la pantalla de autenticación para el ingreso de usuarios permitidos, existe un botón de registro de usuario, este botón habilita la posibilidad de registrare con un correo electrónico y una contraseña, además de algunos datos requeridos para el emisor. Los detalles de la pantalla de registro podemos visualizar en la Figura 2.6.3



Figura 2.6.3 Interfaz – Registro de Usuario

Una vez que se ingresan los datos y la aplicación autentifica la información de usuario, se visualiza la pantalla principal del menú, donde se encuentran opciones para gestionar: Facturas, guías de remisión, notas de débito, notas de crédito y comprobantes de retención. Se encuentra también un menú inferior para gestionar comprobantes, clientes, transportistas y productos. Podemos visualizar las opciones mencionadas en la Figura 2.6.4



Figura 2.6.4 Interfaz – Menú Principal

La pantalla de menú principal cuenta, con una cabecera, en la que se encuentra el nombre del usuario dueño de las credenciales de acceso. Y al lado derecho está un botón que habilita una pantalla para ingresar la información necesaria del emisor. Los detalles de la pantalla mencionada podemos visualizar en la Figura 2.6.5 presentada a continuación.

Android Large - 25

< José Arciniega Información

Ingrese su RUC

Ingrese su nombre o razón social

Ingrese la dirección de la matriz

Ingrese código para su establecimiento

Ingrese su número de contribuyente

Tipo de Ambiente Pruebas

Ingrese el nombre de su empresa

Ingrese la dirección del establecimiento

Ingrese un código - punto de emisión

Obligado a llevar contabilidad

Guardar

Comprobantes Clientes Transportistas Productos

Figura 2.6.5 Formulario de registro/actualización de la información del emisor

La primera opción para la gestión de documentos electrónicos, que se encuentra en el menú principal, es el de facturación. Este botón habilita la pantalla de gestión de usuarios, donde encontramos un listado de las facturas, clasificadas por fecha, id, y nombre. En esta pantalla también se encuentra un botón para ingresar nuevas facturas. La pantalla mencionada la visualizamos en la Figura 2.6.6 que se encuentra a continuación.

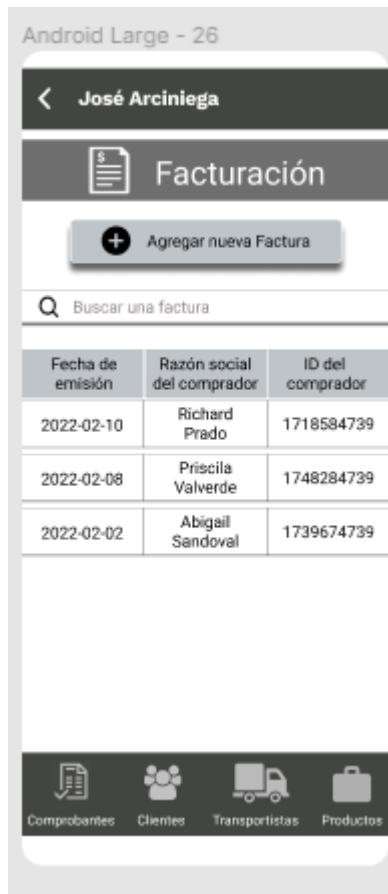


Figura 2.6.6 Pantalla de catálogo y gestión de facturas

Las figuras de las pantallas mostradas, nos dan una visualización general de las interfaces de la aplicación, su estética y el flujo de navegación por las diferentes opciones con las que cuenta el sistema de facturación.

2.7 Modelo de Base de Datos

Con base en la información previa, se puede avanzar hacia una etapa de desarrollo de modelo enfocada en la implementación, lo cual está dentro del diseño de la base de datos necesaria para respaldar la aplicación. Dado que la prioridad de la aplicación no es la velocidad ni la cantidad de escrituras, sino más bien la presentación de datos, se optó por utilizar una base de datos no relacional. El esquema de la base de datos utilizada se encuentra a detalle en el ANEXO III. El proyecto implementó el servicio Firebase Cloud Firestore proporcionado por Google. En la Figura 2.7.1 se encuentra un esquema básico del funcionamiento de la plataforma que proporciona una base de datos para la aplicación.



Figura 2.7.1 Esquema de la relación de la base de datos Firebase con la aplicación

Se creó una cuenta el Firebase para acceder al servicio de base de datos. Posteriormente se creó el proyecto Facturación-ML con las especificaciones requeridas, en el proceso la plataforma genera un archivo de nombre google-services.json, que sirve para realizar la conexión con la base de datos y la aplicación móvil.

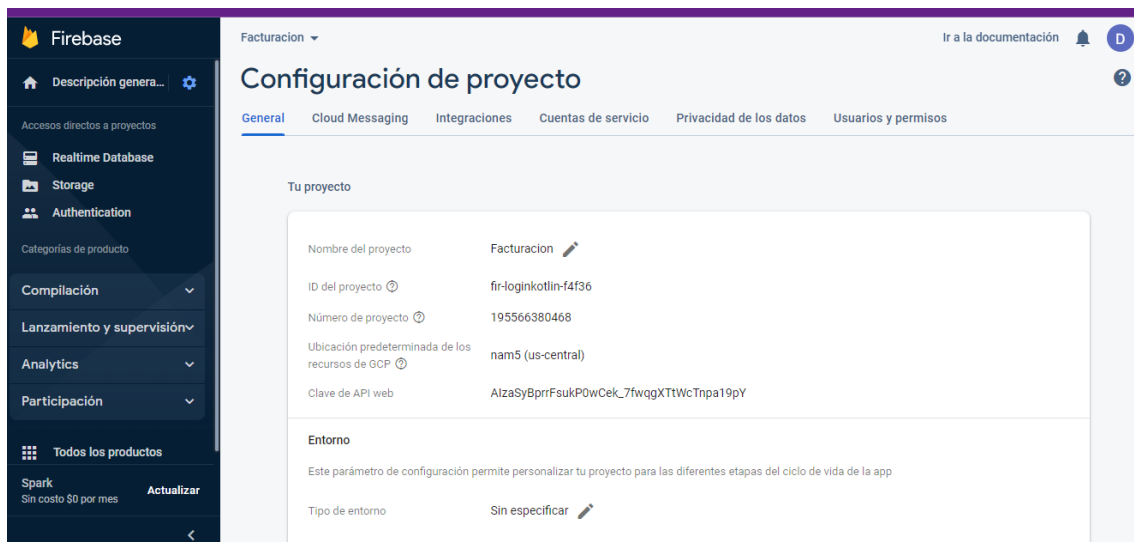


Figura 2.7.2 Proyecto de base de datos de Firebase vinculado con la aplicación

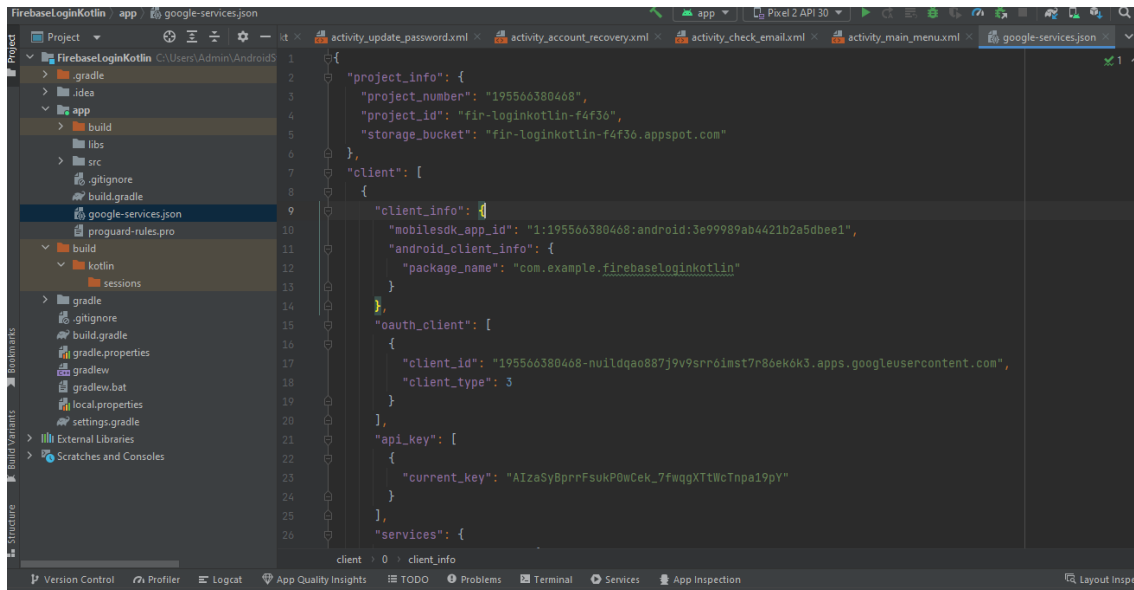


Figura 2.7.3 Archivo google-services.json generado por Firebase

2.8 Implementación/Pruebas de Funcionalidad

Antes de comenzar la implementación, se presentarán algunos aspectos relevantes del proceso, entre ellos:

- La implementación del producto se realizó en el entorno de Android Studio, lo que significa que la aplicación está diseñada para ser ejecutada en sistemas operativos Android o similares capaces de ejecutar archivos APK.
- Se utilizó la plataforma de servicios Firebase, tanto para definir la base de datos a través de Firebase Cloud Firestore como para lanzar y distribuir el producto final mediante Firebase App-Distribution.

Una vez se han establecido los modelos y los requisitos del producto final, así como se han aclarado los puntos anteriores, podemos analizar el progreso en el desarrollo de la aplicación. Para describir el proceso de implementación de la aplicación, se utilizó la metodología Scrum y se dividió el proceso en cada uno de los Sprints ejecutados, los cuales se describirán a continuación. Además, como se definieron los eventos y artefactos, cada uno de los Sprints incluirá una sección que analizará los resultados obtenidos en comparación con lo planificado al inicio del mismo.

Sprint 1

En los primeros Sprints, la implementación se enfocó en establecer la integración continua del entorno y completar las etapas de diseño previamente mencionadas.

Sprint Planning

Los objetivos establecidos para el primer ciclo de trabajo son:

- Completar la descripción de las historias de usuario.
- Realizar ajustes en las interfaces de usuario.

Ejecución del Sprint

Durante las primeras etapas del proyecto, se enfocaron los esfuerzos en la comprensión y práctica de las herramientas nuevas que se utilizarían:

Se completó la descripción detallada de los modelos de Bases de Datos, Interfaces y Historias de Usuario, como se ha visto en capítulos previos, para poder comenzar con la codificación. También se revisó información sobre el uso y configuración de la herramienta GitLab, especialmente para la implementación de la Integración Continua en proyectos de desarrollo. Se consultó diversas fuentes, principalmente del propio GitLab, además de fuentes propias de CircleCI/Android.

Daily Meeting

No se han presentado problemas durante la ejecución del Sprint 1

Sprint Review

Tareas realizadas en el sprint actual:

- Descripción de las Historias de Usuario.
- Ajustes y Mejoras en las Interfaces de Usuario.
- Revisión de la Configuración de CI/CD en GitLab.

Problemas del Sprint:

- Falta de práctica y conocimiento del entorno y las capacidades de GitLab.

Tareas para el siguiente sprint:

- Configurar CI/CD en GitLab con soporte de Docker.

- Configurar el entorno de Android Studio con Firebase.
- Inclusión del proyecto en GitLab.

Sprint Retrospective

El detalle de las historias de usuario se realizó tomando en cuenta trabajos anteriores sobre el mismo proyecto de integración, las historias de usuario tiene las mismas características, de funcionalidad, es decir que tienen la misma lógica de negocio. La implementación de la base de datos con el entorno de Android Studio fue facilitado por la documentación encontrada en la misma plataforma de Firebase.

En el siguiente Sprint se subirá el proyecto a los repositorios de GitHub.

Sprint 2

Sprint Planning

Durante el Segundo Ciclo de trabajo, se establecen las siguientes tareas:

- Desarrollo de las Interfaces de Inicio de Sesión
- Desarrollo de las Interfaces de Gestión de Cuentas

Ejecución del Sprint

Para lograr cumplir con el plan establecido para el ciclo de trabajo, se inició con la implementación de la interfaz de Inicio de Sesión de la aplicación, como se puede observar en la figura 2.8.1

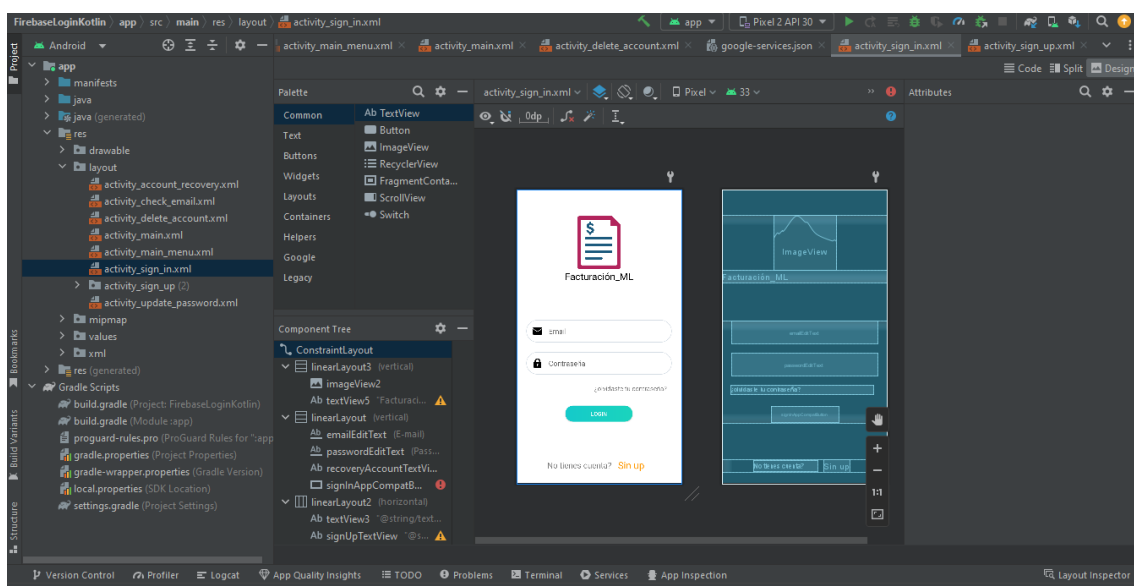


Figura 2.8.1 Implementación de la interfaz de inicio de sesión



lemad217@gmail.com

.....

¿olvidaste tu contraseña?

LOGIN

Email o contraseña o incorrectos.

No tienes cuenta? [Sin up](#)

Figura 2.8.2 Ejecución de la interfaz de Inicio de Sesión

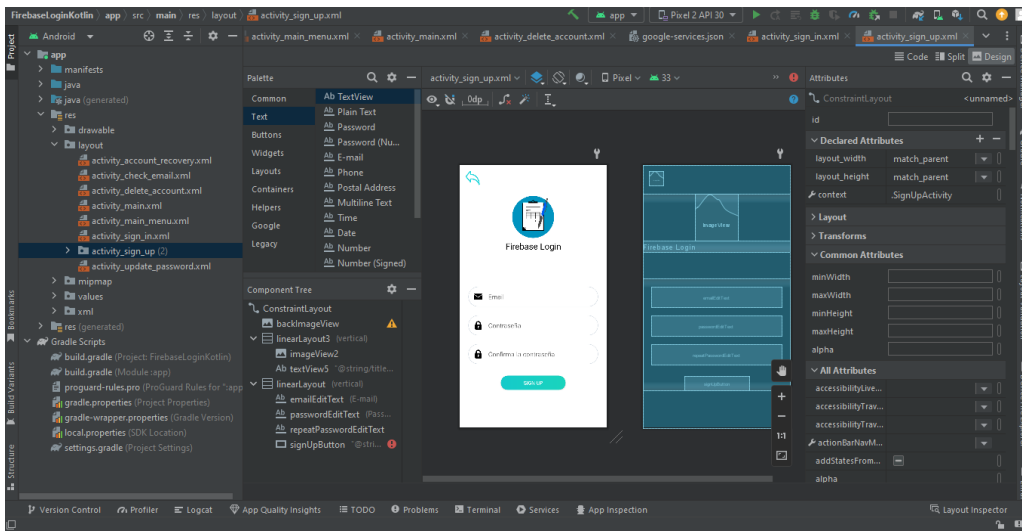


Figura 2.8.3 Implementación de la interfaz de Creación de usuario

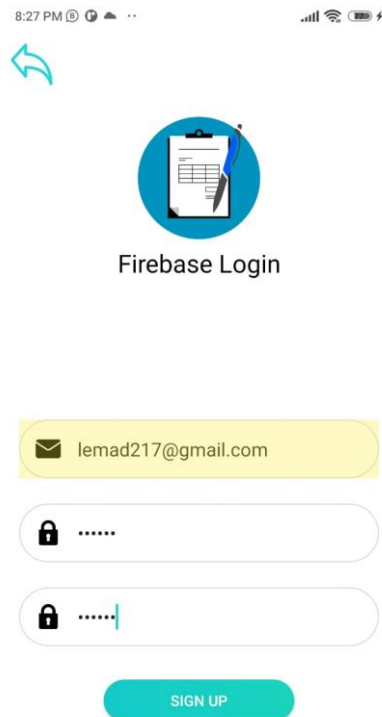


Figura 2.8.4 Ejecución de la interfaz de creación de usuario

Sprint Review

Tareas realizadas en el sprint actual:

- Desarrollo de la Interfaz de Inicio de Sesión
- Desarrollo de la Interfaz de creación de usuario

Problemas del Sprint:

- Falta de práctica y conocimiento del entorno Android Studio.

Tareas para el siguiente sprint:

- Desarrollo de las interfaces y su flujo en Android Studio.
- Desarrollo del flujo completo de cuentas base.

Sprint Retrospective

Durante el Sprint 2, se priorizó el levantamiento de la arquitectura en Android Studio, se elaboró los Layouts necesarios para cumplir los objetivos de este sprint. Se elaboró las actividades que ejecutarán la conexión con la base de datos.

Sprint 3

Sprint Planning

En el tercer ciclo de trabajo se establecen las siguientes tareas:

- Implementar la lógica correspondiente a la interfaz de Inicio de Sesión y a la Página Principal.

Ejecución del Sprint

Después de comenzar con la implementación, se tomó la decisión de diseñar todas las interfaces de la aplicación antes de llevar a cabo la implementación mediante un conjunto de interfaces.

Luego de diseñar la interfaz de Inicio de Sesión, se procedió a diseñar la interfaz correspondiente a la Página Principal, como se muestra en la figura 2.8.1

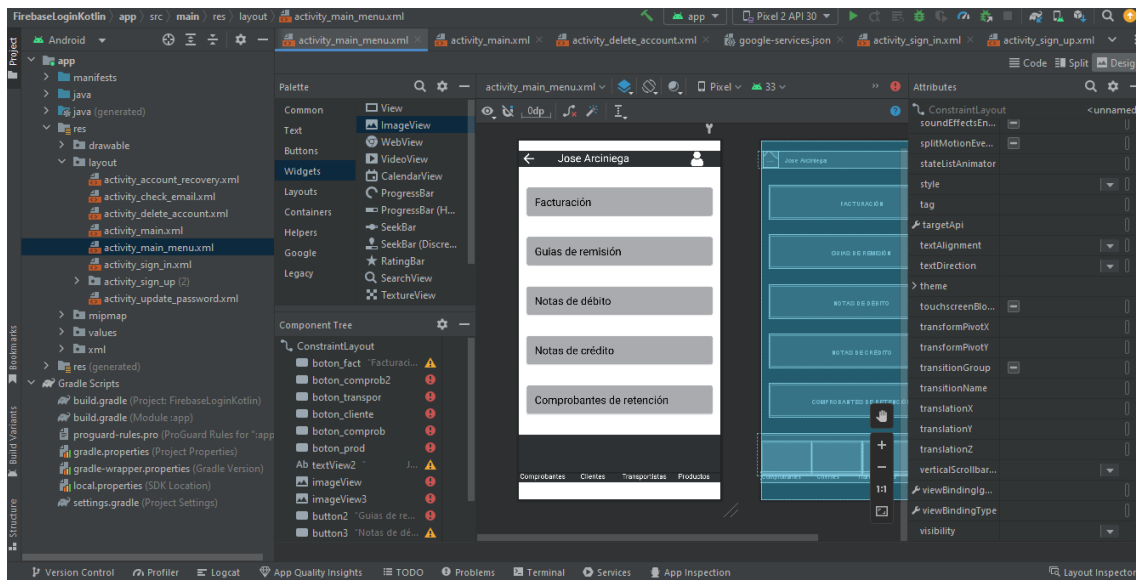


Figura 2.8.5 implementación de la interfaz del menú principal

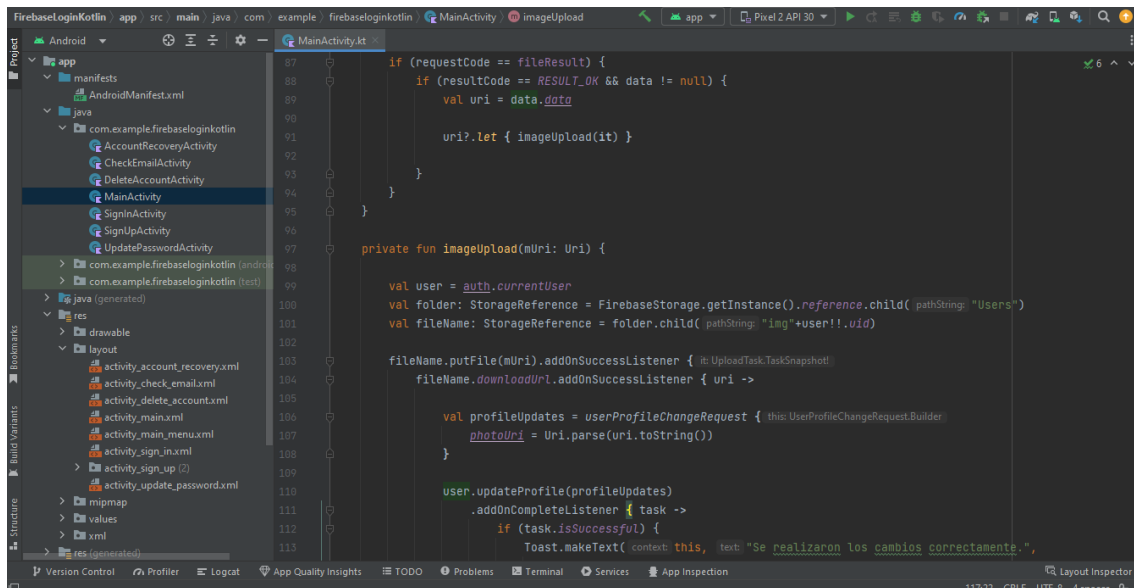


Figura 2.8.5 Desarrollo de las actividades que integran los layouts

Sprint Review

Tareas realizadas en el sprint actual:

- Desarrollo de la Interfaz del menú principal.
- Desarrollo del código de integración de layout.

Problemas del Sprint:

- Sin novedades.

Tareas para el siguiente sprint:

- Validar al usuario con su correo electrónico.
- Elaborar las pantallas auxiliares del inicio de sesión.

Sprint Retrospective

Se encontraron errores fácilmente solucionables en la etapa de codificación de las actividades de la aplicación. Por lo que no se reportó con un problema en el Sprint. Sin embargo, retrasó la entrega del mismo llegando a tiempo de revisión con un tiempo muy ajustado. El inconveniente se tomará en cuenta para las futuras tareas.

Sprint 4

Sprint Planning

Dentro del cuarto Sprint, se definen las actividades descritas a continuación:

- Desarrollo de las interfaces de la aplicación.
- Desarrollo de la lógica de la interfaz de Inicio de Sesión y de la Página Principal.

Ejecución del Sprint

En este Sprint se realizó las interfaces restantes de la aplicación y se desarrollaron las interfaces auxiliares para la autenticación. La autenticación fue una tarea realizada también en este sprint, por lo que la conexión con Firebase era necesario para esta tarea.

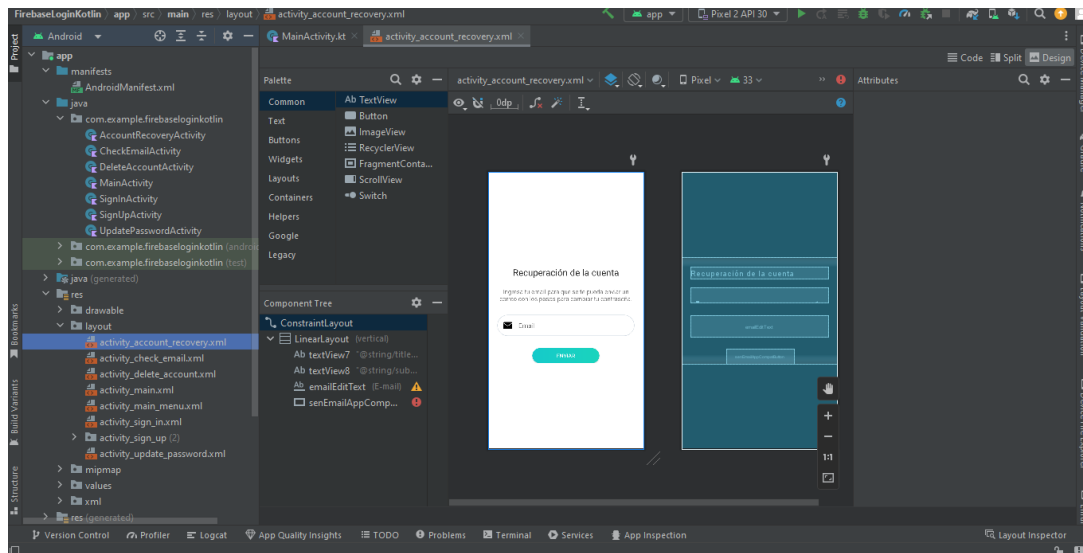


Figura 2.8.6 Implementación de la Interfaz de recuperación de cuenta

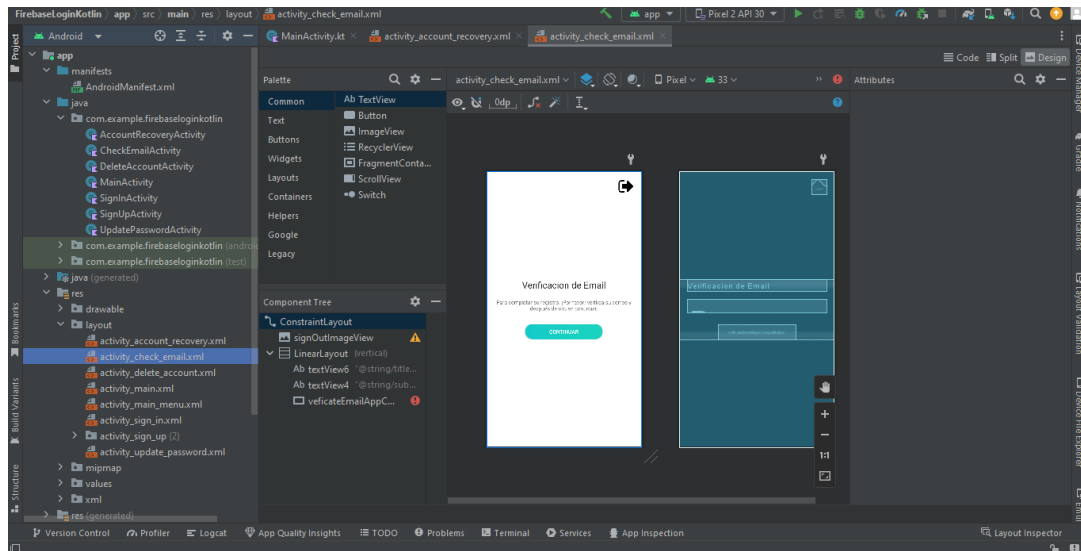


Figura 2.8.7 Implementación de la Interfaz de verificación de correo

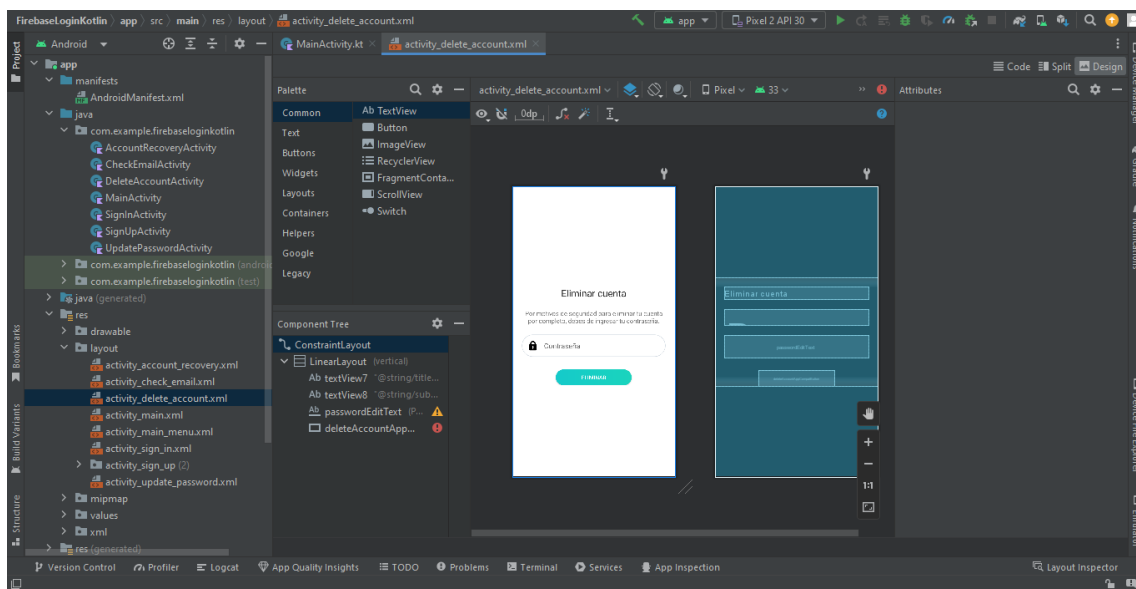


Figura 2.8.8 Implementación de la Interfaz de eliminación de cuenta

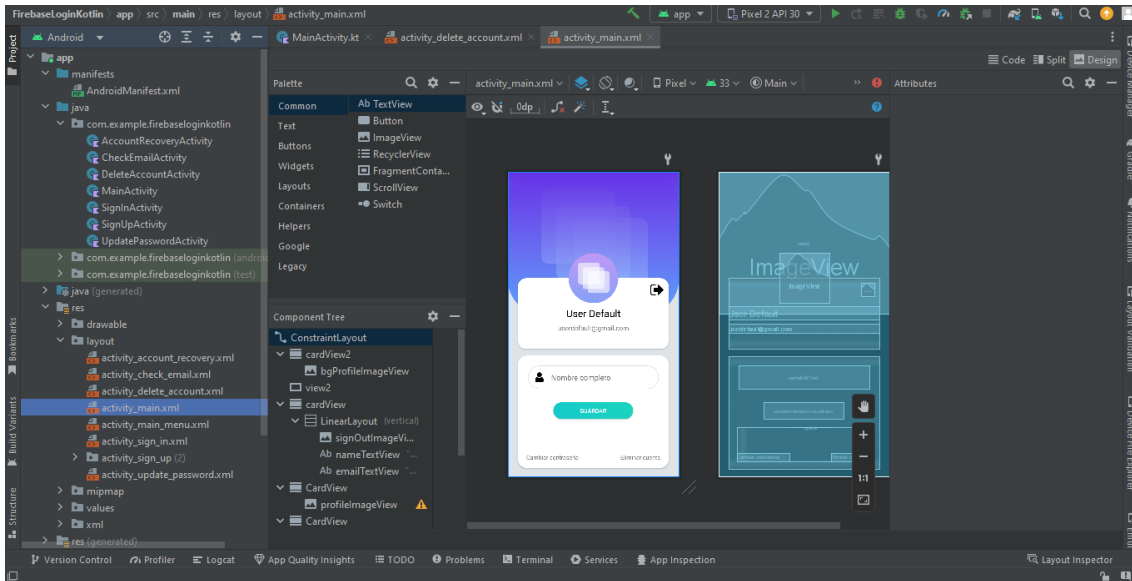


Figura 2.8.9 Implementación de la Interfaz de acceso a tu cuenta

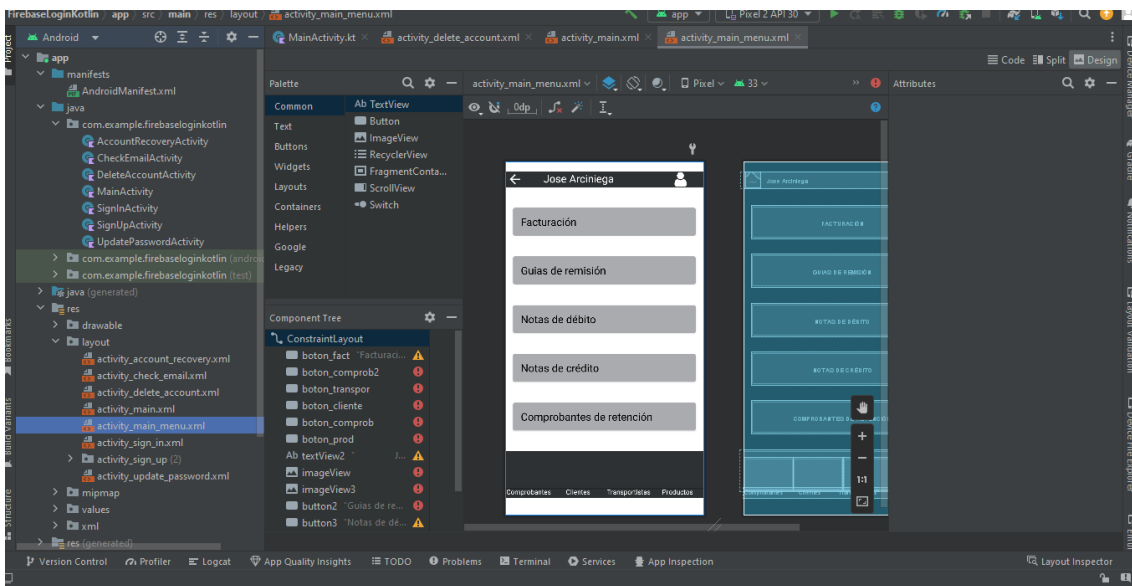


Figura 2.8.9 Implementación de la Interfaz de menú principal

Sprint Review

Una vez finalizado el diseño de las interfaces, se procedió a la implementación de la funcionalidad de algunas pantallas, específicamente la interfaz de inicio de sesión y la página principal de la aplicación. De esta manera, se logró que la funcionalidad de la interfaz de inicio de sesión, esté lista para cumplir sus funciones, las cuales son permitir la identificación del usuario y autenticar y autorizar su acceso a la aplicación. Además, se pudo implementar la funcionalidad de la página principal, que incluye acceso a la

página inicial y flujo de cuentas, acceso a la página inicial y flujo de informe contable, agregar un nuevo periodo contable y acceder a un periodo contable vigente. Aunque se había planificado ejecutar el flujo DevOps, se decidió posponer su implementación hasta completar al menos uno de los flujos, por lo que la inclusión de estas funcionalidades se integró en el siguiente Sprint.

Sprint Retrospective

La integración de las interfaces y sus funcionalidades se realizaron por medio de las actividades desarrolladas para la ejecución de la aplicación.

3 RESULTADOS

3.1 Pruebas de usabilidad

Se evaluará la usabilidad de la aplicación móvil de Facturación electrónica utilizando la metodología del System Usability Scale (SUS). La Tabla 3.1.1 muestra las 10 preguntas del cuestionario SUS y su escala de valoración, donde se otorga una puntuación de 1 a "Totalmente en desacuerdo" y de 5 a "Totalmente de acuerdo".

Tabla 3.1.1. Cuestionario SUS

N°	Pregunta	Valoración				
		1	2	3	4	5
1	Creo que me gustaría usar este sistema con frecuencia.					
2	Encontré el sistema innecesariamente complejo.					
3	Pensé que el sistema era fácil de usar.					
4	Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema.					
5	Descubrí que las diversas funciones de este sistema estaban bien integradas.					
6	Pensé que había demasiada inconsistencia en este sistema.					
7	Me imagino que la mayoría de la gente aprendería a usar este sistema muy rápidamente.					
8	Encontré el sistema muy engorroso de usar.					
9	Me sentí muy confiado usando el sistema.					
10	Necesitaba aprender muchas cosas antes de poder ponerme en marcha con este sistema.					

Según la metodología SUS, es necesario calcular la puntuación final del sistema por medio de la Ecuación 3.1, donde \bar{P}_i es la valoración promedio de la pregunta i del cuestionario SUS.

$$Puntaje\ SUS = 2.5 \times \sum_{i=1}^{10} \bar{P}_i$$

Ecuación 3.1.1 Fórmula para calcular el puntaje SUS

Por otro lado, se establece que, si el puntaje es mayor a 68 sobre 100, entonces se considera mayor al promedio, es decir, tiene un nivel aceptable de usabilidad.

En este estudio, se utilizó una muestra de 15 personas con un rango de edad entre 25 y 36 años. Entre los encuestados se encuentran estudiantes de la Escuela Politécnica Nacional y externos relacionados académicamente a diversas áreas del conocimiento. Se utilizaron canales y herramientas digitales como Google Meets y Google Forms para que los usuarios encuestados puedan comprender el contexto de la aplicación móvil a través de reuniones virtuales síncronas, y llenar el cuestionario SUS en línea, respectivamente. La Tabla 3.1.1 muestra la tabulación de los resultados del cuestionario SUS.

Tabla 3.1.1. Resultados del cuestionario SUS

N° de encuestados	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1	5	1	4	1	5	2	2	2	1	3
2	4	1	2	3	3	5	1	2	1	5
3	5	1	5	2	4	5	2	2	1	5
4	2	3	5	3	1	4	3	2	2	4
5	4	1	5	2	2	5	2	4	2	5
6	5	4	2	3	3	2	1	3	3	3
7	4	3	4	5	4	3	1	3	1	5
8	5	5	5	3	5	3	1	5	1	4
9	3	2	5	4	5	4	3	3	4	4
10	5	2	5	3	2	5	3	5	2	3
11	3	4	2	2	5	5	3	3	2	3
12	5	2	3	5	4	4	2	6	1	3
13	4	3	3	4	5	4	3	5	3	4
14	4	5	4	3	4	5	2	2	1	3
15	4	2	5	1	3	4	5	5	1	2
Promedio	4,13	2,60	3,93	2,93	3,67	4,00	2,27	3,47	1,73	3,73

Al realizar el cálculo del puntaje SUS se obtiene el valor de 81.15. Por lo cual, estos resultados muestran que el nivel de usabilidad del sistema es aceptable.

3.2 Resultados de las Pruebas

Volviendo al propósito del proyecto que se centra en comparar el desarrollo a través de flujos DevOps seguros versus aquellos sin el uso de ellos, se han presentado cuatro métricas o formas de medir los resultados obtenidos. Estas son: el número de flujos completados exitosamente, con error o cancelados; el tiempo de ejecución de los flujos; el número de pasos necesarios para completar un flujo; y el porcentaje de tareas completadas dentro de un Sprint en comparación con las tareas planificadas. Utilizando estas medidas, se han cuantificado los resultados de los flujos DevOps ejecutados y se han categorizado en función de su resultado, es decir, flujos completados de manera exitosa, flujos que han fallado debido a errores en la configuración o durante la ejecución, y flujos que han sido cancelados.

En su mayoría, los flujos que fallan son aquellos en las etapas iniciales de su implementación. De hecho, de los últimos 12 flujos ejecutados, solo uno tuvo un resultado fallido. Esto sugiere que aunque la configuración y la implementación correcta de los flujos pueden causar problemas y confusiones en el equipo de trabajo al principio, a medida que las personas se familiarizan con el proceso, se convierte en una herramienta muy útil. En lugar de ser una preocupación, el flujo de trabajo se convierte en un aliado poderoso del equipo que puede ser aprovechado en gran medida.

La figura 3.1.2 muestra el promedio de calificación otorgado a cada una de las preguntas o ítems de la tabla 3.1.2 correspondientes a la escala de la metodología SUS.

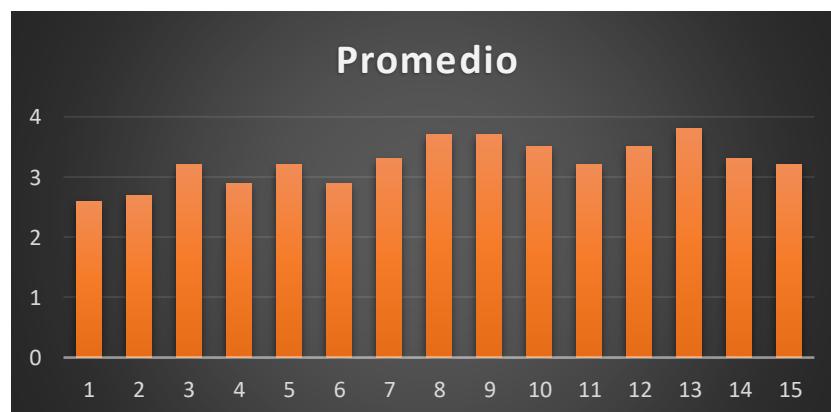


Figura 3.1.2 Promedios por pregunta/ítem de SUS

4 CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Antes de decidir si se debe utilizar flujos DevOps en un proyecto, es necesario realizar un análisis exhaustivo de varios componentes del proyecto, incluyendo el alcance, el tamaño, las herramientas o servicios utilizados, el equipo de trabajo y otros factores relevantes. Tomar en cuenta cada uno de estos aspectos ayuda a evitar el esfuerzo innecesario de implementar flujos DevOps en proyectos que no lo requieren, o la omisión de flujos DevOps en procesos que podrían beneficiarse de su implementación.

En el desarrollo de aplicaciones móviles, es probable que los flujos de trabajo incluyan la compilación del proyecto para la generación de un archivo ejecutable, como un APK para dispositivos Android. Este archivo es esencial para realizar pruebas en entornos reales o virtuales, así como para distribuir la aplicación a través de tiendas de aplicaciones o servicios propios o externos. Por lo tanto, la compilación del proyecto es una parte crucial en el proceso de desarrollo de flujos de trabajo para aplicaciones móviles.

A través de este proyecto de integración curricular, se lograron alcanzar los objetivos establecidos gracias al uso de diversas herramientas y métodos de evaluación. La implementación del marco de trabajo SCRUM permitió abordar cambios inesperados y generar valor al final de cada iteración, al producir incrementos funcionales que pudieron ser probados y presentados a los interesados durante el desarrollo del proyecto.

Mediante la combinación de sesiones virtuales con el Product Owner, análisis de sistemas similares, maquetas visuales y modelos de datos, se pudo realizar exitosamente la toma de requerimientos. Sin embargo, se presentaron dificultades en la comprensión de los requerimientos técnicos de los comprobantes electrónicos, así como en su alcance para el proyecto y la naturaleza de los procesos tributarios relacionados con ellos. A pesar de estos inconvenientes, se pudo solucionar mediante la comunicación constante con el Product Owner en sesiones virtuales y aprendiendo sobre los procesos tributarios a través de herramientas similares. Como

resultado, se refinaron y priorizaron los requerimientos del módulo de facturación para cumplir con las necesidades de Manticore Labs y ofrecer un producto de calidad que sea competitivo en el mercado.

El módulo de facturación por medio de una aplicación móvil, fue diseñado con éxito antes de la ejecución del Sprint 0, gracias a la creación de prototipos en Figma de las interfaces de usuario y el modelo de base de datos. Estos prototipos ayudaron a definir el alcance del módulo y tener una idea aproximada del producto final y la navegación entre las diferentes rutas. Sin embargo, al tratarse de un prototipo de mediana fidelidad, se realizaron ajustes en la dinámica de navegación y diseño del producto final para cumplir con los requerimientos refinados, como la visualización de información del emisor y el formato de registro de todos los comprobantes. En general, la fase de diseño fue esencial para apoyar la planificación del release y el refinamiento del Product Backlog.

4.2 Recomendaciones

Para que los beneficios de los flujos DevOps centrados en la Integración Continua y/o Despliegue Continuo sean más evidentes y comprensibles, se recomienda implementarlos en proyectos que involucren a dos o más desarrolladores. Esto asegura que el proceso de codificación no sea lineal ni ejecutado por una sola persona, sino que haya múltiples flujos de cambios en el producto en desarrollo. La integración correcta, estructurada y lógica de estos flujos facilita el trabajo de todo el equipo.

La recomendación es agregar y fusionar un módulo adicional de seguridad al sistema con el fin de asegurar la eficacia del aplicativo al ser utilizado por el público general en su entorno de producción. Con esta medida, se puede salvaguardar la privacidad de la información confidencial que se maneja dentro de la aplicación, como los datos personales de los usuarios, las organizaciones, los clientes y los transportistas.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] SRI, «Facturación Física,» [En línea]. Available: <https://www.sri.gob.ec/facturacion-fisica#:~:text=Sirve%20para%20respaldar%20las%20transacciones,venta%2C%20retenci%20y%20documentos%20complementarios>
- [2] Martínez, J. M. A., Fa, M. C., & Elguezabal, I. Z. (2005). Evolución histórica de los sistemas ERP: de la gestión de materiales a la empresa digital. Revista de Dirección y Administración de empresas.
- [3] J. Nestell and D. Olson, Successful ERP Systems : A Guide for Businesses and Executives, 1st ed. Business Expert Press, 2018 [Online]. Available: <https://bibliotecas.ups.edu.ec:2708/lib/upsal/reader.action?docID=5165162&query=978-1-63157-846-5>
- [4] Zapata Barreno, F. J. (2019). Aplicación móvil con implementación de un modelo de inteligencia de negocios para el proceso de facturación de la ferretería “Del Ahorro” usando la herramienta Microsoft Power BI (Bachelor's thesis).
- [5] DeveloperAndroid, “Kotlin overview,” Android Developers. [Online]. Available: <https://developer.android.com/kotlin/overview>
- [6] W3, “Extensible Markup Language (XML),” W3.org, 2016. [Online]. Available: <https://www.w3.org/XML/>
- [7] RedHat, “¿Qué es YAML? YAML y su uso en la automatización,” www.redhat.com, 2021. [Online]. Available: <https://www.redhat.com/es/topics/automation/what-is-yaml>
- [8] GitLab, “About GitLab,” GitLab. [Online]. Available: <https://about.GitLab.com/company/>
- [9] GitLab, “GitLab CI/CD | GitLab,” docs.GitLab.com. [Online]. Available: <https://docs.GitLab.com/ee/ci/>

- [10] GitLab, "GitLab Runner | GitLab," docs.GitLab.com. [Online]. Available: <https://docs.GitLab.com/runner/>
- [11] GitLab, "Repository | GitLab," docs.GitLab.com. [Online]. Available: <https://docs.GitLab.com/ee/user/project/repository/>
- [12] Git, "Git - About Version Control," Git-scm.com, 2019. [Online]. Available: <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>
- [13] Theme Junkie, «What is Figma? (And How to Use Figma for Beginners),» 2022. [En línea]. Available: <https://www.theme-junkie.com/what-is-figma/>
- [14] Google, "Documentation | Firebase," Firebase, 2019. [Online]. Available: <https://firebase.google.com/docs>
- [15] Google, "Cloud Firestore | Firebase," Firebase, 2019. [Online]. Available: <https://firebase.google.com/docs/firestore>
- [16] Google, "Cloud Firestore Data model," Firebase. [Online]. Available: <https://firebase.google.com/docs/firestore/data-model>
- [17] Google, "Firebase App Distribution," Firebase. [Online]. Available: <https://firebase.google.com/docs/app-distribution>
- [18] Docker, "Docker overview," Docker Documentation, Apr. 09, 2020. [Online]. Available: <https://docs.docker.com/get-started/overview/>
- [19] Google Developers, "Meet Android Studio | Android Developers," Android Developers, 2019. [Online]. Available: <https://developer.android.com/studio/intro>
- [20] Atlassian, «What is Agile?,» [En línea]. Available: <https://www.atlassian.com/agile#:~:text=Agile%20is%20an%20iterative%20approach,s mall%2C%20but%20consumable%2C%20increments>
- [21] K. Schwaber y J. Sutherland, The Scrum Guide, Creative Commons, 2020.
- [22] J. Monte, Implantar SCRUM con éxito, Barcelona, España: UOC, 2016.

6 ANEXOS

ANEXO I. REPOSITORIO DIGITAL DEL FRONT-END

El repositorio digital de Front-End en GitHub del módulo de Facturación se puede observar en el siguiente

<https://github.com/JoseDavid20/FacturacionML.git>

ANEXO II. REPOSITORIO DIGITAL DEL BACK-END

El repositorio digital de Back-End en GitHub del módulo de Facturación se puede observar en el siguiente

<https://github.com/JoseDavid20/FacturacionML.git>

ANEXO III. ENLACE AL PROTOTIPO DE LA BASE DE DATOS

En este enlace se encuentra la base de datos que se diseñó para el proyecto.

<https://www.figma.com/file/7gZJsQpQ7M0DR1st4C6QGr/Base-de-Datos-Facturaci%C3%B3n?node-id=0%3A1&t=V3F2zMnnaHGtYoiT-1>

ANEXO IV. PROTOTIPO DE MEDIANA FIDELIDAD

El prototipo de mediana fidelidad para las interfaces gráficas del módulo de facturación se puede observar en el siguiente.

<https://www.figma.com/proto/Bmm0oA0ZCR81HpVayWNFzi/Facturaci%C3%B3n-ML?node-id=520%3A583&scaling=scale-down&page-id=0%3A1&starting-point-node-id=1%3A3>