



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA DE COMPUTACIÓN

**DESARROLLO DE UN SOFTWARE DE FACTURACIÓN CON INTELIGENCIA
ARTIFICIAL MEDIANTE EL USO DE MICROSERVICIOS EN LA NUBE DE AWS
ENFOCADO PARA MICROEMPRESAS EN ECUADOR**

Trabajo de titulación previo a la obtención
del título de Ingeniero en Ciencias de la
Computación

AUTORES: MILTON RAFAEL CORAL NAVARRO
EDISON ALEXANDER ÑACATO GANCHALA

TUTOR: JULIO RICARDO PROAÑO ORELLANA

Quito - Ecuador
2024

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Milton Rafael Coral Navarro con documento de identificación N.º 1753904257 y Edison Alexander Ñacato Ganchala con documento de identificación N.º 1721524880 manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 05 de marzo del 2024

Atentamente,



Milton Rafael Coral Navarro

1753904257



Edison Alexander Ñacato Ganchala

1721524880

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

Nosotros, Milton Rafael Coral Navarro con documento de identificación N.º 1753904257 y Edison Alexander Ñacato Ganchala con documento de identificación N.º 1721524880, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos los autores del Proyecto Técnico: “Desarrollo de un software de facturación con inteligencia artificial mediante el uso de microservicios en la nube de AWS enfocado para microempresas en Ecuador”, el cual ha sido desarrollado para optar por el título de: Ingenieros en Ciencias de la Computación, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

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

Quito, 05 de marzo del 2024

Atentamente,



Milton Rafael Coral Navarro

1753904257



Edison Alexander Ñacato Ganchala

1721524880

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Julio Ricardo Proaño Orellana con documento de identificación N.º 0103909412, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DESARROLLO DE UN SOFTWARE DE FACTURACIÓN CON INTELIGENCIA ARTIFICIAL MEDIANTE EL USO DE MICROSERVICIOS EN LA NUBE DE AWS ENFOCADO PARA MICROEMPRESAS EN ECUADOR, realizado por Milton Rafael Coral Navarro con documento de identificación N.º 1753904257 y Edison Alexander Ñacato Ganchala con documento de identificación N.º 1721524880, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 05 de marzo del 2024

Atentamente,



Ing. Julio Ricardo Proaño Orellana, PhD
0103909412

DEDICATORIA

A mis padres Silvana y Edison

Por darme su apoyo, cariño, guía y tiempo en todo el transcurso de mi vida. Han hecho esfuerzos titánicos para mantener nuestra familia contenta, este logro hoy en día es totalmente para ustedes que siempre han creído en mí y han visto el potencial para ser alguien mejor. Ellos han sido el ejemplo para mi vida, Dios quiera sea una persona de bien al igual que ellos.

A mi hermano Josué

Gracias por ser aquel que ha estado riéndose a mi lado, aprecio mucho su compañía y al igual que yo espero sigas avanzado no importa cuantas veces tropieces, siempre estará tu familia para apoyarte, escucharte, levantarte y esperarte.

AGRADECIMIENTO

Alexander

A todos aquellos que estuvieron durante mi proceso académico, amigos compañeros, maestros, instructores, jefes y demás. Han aportado tanto para mi vida, me han hecho crecer de manera profesional, emocional y como ser humano. Han guiado mis errores, me han mostrado un camino para poder desarrollarme profesionalmente y sobre todo han sentado las bases éticas sobre las cuales trabajar.

Agradezco a mi familia extendida la cual ha estado en los buenos y malos momentos, siempre apoyándome y compartiendo todo lo que tienen. Os llevo en mi corazón siempre, solo me queda decir que han sido ese motor para jamás rendirme.

Algún día a todos los que me han ayudado y han estado ahí os pagare por su lealtad, bondad y cariño.

¡Gracias totales!

ÍNDICE

RESUMEN	1
ABSTRACT.....	2
INTRODUCCIÓN	3
Problema	3
Antecedentes.....	3
Importancia y alcances	5
Delimitación	5
Objetivos Generales y Específicos.....	5
Objetivo General.....	5
Objetivos Específicos	5
CAPÍTULO I	6
FUNDAMENTOS TEÓRICOS	6
1.1. Microempresas en Ecuador.....	6
1.2. Contabilidad en las Microempresas	6
1.3. Facturación electrónica en Ecuador	6
1.4. Software de facturación para microempresas	6
1.5. Serverless	7
1.5.1. Amazon Web Services	7
1.5.2. AWS Amplify	7
1.5.3. AWS Lambda.....	7
1.5.4. AWS Cognito.....	7
1.5.5. Amazon Relational Database Service.....	8
1.5.6. AWS Elastic Beanstalk	8
1.5.7. AWS CodePipeline	8
1.6. Microservicios.....	8
1.7. Aplicaciones multiplataforma.....	8

1.8.	Inteligencia Artificial	8
1.9.	Procesamiento natural del lenguaje	9
1.9.1.	spaCy	9
1.9.2.	Expresiones regulares	9
1.9.3.	Pandas	9
1.9.4.	Tf-idf.....	9
1.9.5.	Naive Bayes para clasificación de textos	10
1.9.6.	Split de datos, Cross-validation	10
1.9.7.	Multinomial Naive Bayes Sklearn.....	10
1.10.	Metodología cascada.....	10
1.10.1.	Requerimientos del sistema	10
1.10.2.	Diseño del Sistema	10
1.10.3.	Implementación	11
1.10.4.	Integración y Prueba.....	11
1.10.5.	Despliegue del sistema	11
1.10.6.	Mantenimiento.....	11
1.11.	Metodología CRISPDM.....	11
1.11.1.	Comprensión del negocio	11
1.11.2.	Comprensión de datos.....	11
1.11.3.	Preparación para uso de datos.....	12
1.11.4.	Modelado	12
1.11.5.	Evaluar los modelos.....	12
1.11.6.	Despliegue	12
CAPÍTULO II.....		14
MARCO METODOLÓGICO.....		14
2.1.	Especificación de Requerimientos.....	14
2.1.1.	Introducción.....	14

2.1.1.1.	Propósito	14
2.1.1.2.	Alcance	14
2.1.1.3.	Definiciones, acrónimos y abreviaturas	14
2.1.2.	Descripción general del sistema	14
2.1.2.1.	Perspectiva	15
2.1.2.2.	Funcionalidades	15
2.1.2.3.	Características de los usuarios	15
2.1.2.4.	Restricciones	15
2.1.2.5.	Evolución a futuro del software	15
2.1.3.	Requisitos específicos.....	15
2.1.3.1.	Requisitos funcionales	15
2.1.3.2.	Limitantes	17
2.1.3.3.	Requisitos no funcionales	17
2.1.3.4.	Interfaces de usuario	18
2.2.	Modelado de Sistema	24
2.2.1.	Registro de usuarios.....	24
2.2.2.	Registro de clientes.....	25
2.2.3.	Registro de proveedores	26
2.2.4.	Registro de productos	26
2.2.5.	Facturación	28
2.2.6.	Registrar cuentas contables	29
2.2.7.	Registrar movimientos en el Kardex simplificado	30
2.2.8.	Registrar datos en el libro diario.....	30
2.2.9.	Registrar datos en el libro mayor.....	31
2.2.10.	Esquema de la base de datos.....	32
2.3.	Arquitectura de Sistema	33
2.4.	Uso de la metodología	36

2.4.1. Uso de metodología de datos.....	36
2.4.1.1. Comprensión del negocio	36
A. Objetivo del proyecto.....	36
B. Requisitos y limitaciones del negocio.....	36
2.4.1.2. Comprensión de datos.....	36
A. Adquisición de datos.....	36
B. Exploración de datos	37
C. Descripción de los diferentes datos.....	38
2.4.1.3. Preparación para uso de los datos	40
A. Curación y limpieza de datos	40
B. Transformación de los datos.....	41
2.4.1.4. Modelado	42
A. Técnicas utilizadas para entrenar el modelo	42
B. División de datos.....	42
C. Desarrollo de los modelos	44
2.4.1.5. Evaluar los modelos.....	44
A. Rendimiento de los algoritmos	44
B. Interpretación de los resultados.....	44
2.4.1.6. Despliegue.....	44
A. Implementación del modelo.....	44
B. Proceso de despliegue	45
C. Uso del modelo.....	45
CAPÍTULO III.....	46
RESULTADOS	46
3.1. Formulario para crear factura – vista desde la aplicación móvil	46
3.2. Funciones desplegadas en Amazon Lambda.....	48
3.3. Modelo de inteligencia artificial desplegado en Elastic Beanstalk.....	48

3.4. Pruebas.....	49
CONCLUSIONES	61
RECOMENDACIONES.....	61
REFERENCIAS.....	63

ÍNDICE DE TABLAS

Tabla 1: Productos de consumo masivo	37
Tabla 2: Productos de ferretería	37
Tabla 3: Medicamentos de venta libre	38
Tabla 4: Fertilizantes y productos agrícolas.....	38
Tabla 5: Variables de los datos sobre productos de consumo masivo.....	39
Tabla 6: Variables de los datos sobre productos de ferretería	39
Tabla 7: Variables de los datos sobre medicamentos de venta libre	40
Tabla 8: Productos escogidos por subcategoría	40

ÍNDICE DE FIGURAS

Figura 1: Interfaces móviles.....	19
Figura 2: Interfaz Contabilidad.....	20
Figura 3: Interfaz plataformas de pago	20
Figura 4: Interfaz datos del cliente para la factura.....	21
Figura 5: Interfaz lista de clientes	21
Figura 6: Interfaz proveedores	22
Figura 7: Interfaz de inventario.....	22
Figura 8: Interfaz inicio de sesion.....	23
Figura 9: Diagrama registro de usuarios	24
Figura 10: Diagrama registro de clientes	25
Figura 11: Diagrama registro de proveedores	26
Figura 12: Diagrama para el registro de productos.....	26
Figura 13: Diagrama de la facturación.....	28
Figura 14: Diagrama del registro de cuentas contables	29
Figura 15: Diagrama para el registro del Kardex simplificado.....	30
Figura 16: Diagrama del registro en el libro diario.....	31
Figura 17: Diagrama del registro en el libro mayor.....	32
Figura 18: Esquema de tablas de la BD	33
Figura 19: Diagrama general de la arquitectura de la aplicación.....	35
Figura 20: Expresión regular	40
Figura 21: Ecuación para elección aleatoria	42
Figura 22: Despliegue del modelo de IA	45
Figura 23: Apartados “Datos del cliente” y “Selección de productos”	46
Figura 24: Apartado “Métodos de pago” y “Método en efectivo”	47
Figura 25: Apartado “Resumen de factura”	47
Figura 26: Funciones Lambda desplegadas	48
Figura 27: Carga desde GitLab a CodePipeline.....	49
Figura 28: Despliegue del modelo en Elastic Beanstalk.....	49

RESUMEN

Este trabajo presenta el desarrollo de un software que parte desde una investigación acerca del uso de este tipo de software para empresas ecuatorianas, las cuales por mandato de ley deben realizar toda factura de manera electrónica, exceptuando negocios populares en los cuales nos enfocamos. Como resultado se desarrolla un software que permite la facturación a pequeña escala, lo cual es un beneficio para este tipo de negocios permitiéndoles mantener un mejor registro de las transacciones para su posterior uso ya sea en una gestión administrativa o para verificar datos tributarios. Se tiene en cuenta que conforme ha avanzado la tecnología y se han desarrollado cada vez más negocios digitales, la facturación física puede llegar a ser compleja de mantener, por ello la tecnología nos ayuda a simplificar procesos que normalmente supondrían un esfuerzo extra el mantener. Así como nos ayuda en el almacenamiento, gestión, acceso y envío de este tipo de documentos priorizando así el servicio a los clientes.

También es importante recalcar que gracias a la nueva popularidad de la inteligencia artificial podemos recurrir a esta para ayudarnos en procesos como la clasificación de transacciones comerciales, esto se debe a que podemos entrenar modelos para extraer palabras clave que nos ayuden a la clasificación, esto supone un impacto y mejora a la hora de tener una ley u ordenanza que cambie la forma de clasificar estas transacciones. Como arquitectura basada en microservicios alojados en la nube, se utilizó un proveedor mundialmente reconocido como es Amazon Web Services. Que nos permite agilizar el proceso de desarrollo sin preocuparnos por la infraestructura, además sus costos son asequibles y nos permite alojar modelos de inteligencia artificial de manera sencilla.

- Palabras claves: AWS, Microservicios, Facturación Electrónica, Inteligencia Artificial

ABSTRACT

This work presents the development of a software that starts from an investigation about the use of this type of software for Ecuadorian companies, which by law must make all invoices electronically, except for popular businesses in which we focus. As a result, software developed that allows invoicing on a small scale, which is a benefit for this type of business, allowing them to keep a better record of transactions for later use either in administrative management or to verify tax data. It is considered that as technology has advanced and increasingly digital businesses have developed, physical invoicing can become complex to maintain, so technology helps us to simplify processes that would normally be an extra effort to maintain. As well as helping us in the storage, management, access and sending of such documents, thus prioritizing the service to customers.

It is also important to emphasize that thanks to the new popularity of artificial intelligence we can resort to this to help us in processes such as the classification of commercial transactions, this is because we can train models to extract keywords that help us to classify, this is an impact and improvement when having a law or ordinance that changes the way to classify these transactions. As an architecture based on microservices hosted in the cloud, we used a globally recognized provider such as Amazon Web Services. That allows us to streamline the development process without worrying about the infrastructure, also its costs are affordable and allows us to host artificial intelligence models in an effortless way.

- Key words: AWS, Microservices, Electronic Invoicing, Artificial Intelligence.

INTRODUCCIÓN

Las MiPymes son fundamentales para la economía global, ya que constituyen la mayoría de las empresas en todo el mundo, teniendo una importancia sustancial en la generación de empleo y la contribución al PIB. En Ecuador, se ha implementado una estrategia dirigida a promover prácticas fiscales internacionales adecuadas y combatir la evasión tributaria, inicialmente enfocada en los Grandes Contribuyentes. Sin embargo, la ley orgánica de desarrollo económico y sostenibilidad fiscal desarrollada después de la pandemia mundial manifiesta que existe un interés creciente en ejercer un mayor control sobre todos los contribuyentes.

La facturación electrónica se vuelve obligatoria, lo que permitirá un seguimiento en tiempo real y controles más efectivos. Esta normativa aplica a negocios que facturen más de \$300.000 anuales. Si bien aún no es obligatorio para pequeñas y microempresas, es mejor empezar a adoptar en estos niveles la facturación electrónica. No solo porque en un futuro la normativa lo va a exigir, si no también es una forma más eficaz de mantener los balances y el control de un emprendimiento.

Problema

Un estudio desarrollado por un medio de comunicación ecuatoriano, determino que en Ecuador hay cerca de 2,27 millones de contribuyentes con RUC, de los cuales aproximadamente el 11,8% utiliza sistemas electrónicos para emitir facturas, dejando a dos millones que deben adaptarse a esta normativa. Esta transición enfrenta desafíos como la desinformación, falta de conocimientos para utilizar plataformas digitales y la ausencia de una cultura tributaria arraigada en algunos segmentos de contribuyentes. (Tapia, 2023)

En Ecuador, aproximadamente el 50,03% de las MiPymes no emplea sistemas informáticos para la facturación o contabilidad, lo que supone una dificultad en la toma de decisiones a nivel gerencial y el cálculo de costos-gastos a nivel de producción. (Arellano Cepeda et al., 2017)

Por eso, se plantea desarrollar un software de facturación con inteligencia artificial para el apartado contable como lo es la clasificación del IVA en los productos o bienes, utilizando la arquitectura de microservicios en la nube de AWS.

Aunque las microempresas no están obligadas a emitir comprobantes electrónicos en todos los casos, mantener un registro digital de transacciones se propone como una práctica recomendada para un control eficaz de sus operaciones diarias, según el Servicio de Rentas Internas (SRI).

Antecedentes

Según el código orgánico de la producción, comercio e inversiones (COPCI), las MiPymes son personas, ya sean naturales o jurídicas, que desarrollan una actividad económica (producción, servicios o comercio) y tienen un límite específico de personal y ventas por año establecido en las categorías incluidas dentro de este, según lo determinado en su reglamento. (Reglamento de Inversiones Del Código Orgánico de La Producción, 2019)

En el caso de las microempresas, este reglamento establece criterios claros para su correcta identificación, separándolo así de otro tipo de empresas:

- Número de empleados: Máximo de nueve.
- Ingresos Brutos Anuales: Depende del régimen en el que se encuentren. Por ejemplo, hasta trescientos mil dólares bajo el régimen emprendedor o por debajo de veinte mil dólares en negocios populares.

El Servicio de Rentas Internas (SRI) presenta el Régimen Simplificado para Emprendedores y Negocios Populares (RIMPE), dividido en dos categorías:

- **Emprendedores:** persona o empresa cuyos ingresos brutos anuales son de hasta trescientos mil dólares.
- **Negocios populares:** persona o empresa con ingresos brutos anuales inferiores a veinte mil dólares.

A menudo, las microempresas caen en la categoría de negocios populares, y aunque no están obligadas a llevar contabilidad detallada según las especificaciones del RIMPE, se les recomienda mantener una contabilidad simplificada.

La Superintendencia de Compañías, Valores y Seguros (SuperCias) indica, aunque estas empresas o más bien dicho emprendimientos no tienen obligación alguna de rendir cuentas públicamente, podría ser necesario o recomendable mantener estados financieros generales para uso interno o, si se requiere, para usuarios externos. (Casinelli, 2011)

Muchas microempresas encuentran dificultades en su propia administración por la falta de herramientas y procesos tecnológicos eficientes. Por eso, se propone desarrollar un prototipo que ofrezca una solución asequible para estas empresas, que a menudo carecen de recursos para invertir en software costoso y, a la vez, no aprovechan al máximo las características de los programas existentes. (Coello et al., 2021)

Este proyecto busca simplificar la facturación, que es clave para registrar las transacciones de compraventa de las microempresas, y proporcionar herramientas modernas para aumentar su eficiencia operativa, basándose en los resultados de estas transacciones.

El enfoque en microservicios permite la escalabilidad y adaptabilidad de las soluciones tecnológicas, lo que brinda a las organizaciones la flexibilidad necesaria para ajustar y mejorar sus sistemas según las cambiantes demandas del mercado. (Vera-Rivera et al., 2019)

La introducción de la inteligencia artificial ha transformado la manera de llevar a cabo tareas de diversa índole. La implementación del aprendizaje automático en contabilidad reduce los incidentes de fraude, aumenta la confiabilidad y optimiza el cumplimiento normativo, según lo señalado por especialistas en el campo. (Moncayo, 2020)

Importancia y alcances

Bajo las premisas previas, esta aplicación resalta cómo la implementación de un software encargado de facturar electrónicamente optimiza el método de compra, venta, administración y cumplimiento. Además de ofrecer beneficios medioambientales al reducir el uso de papel. Desarrollada con el objetivo de convertirse en un apoyo para aquellos que ofrecen bienes o servicios, incluso si no tienen la obligación actual de generar documentos electrónicos. Aunque las regulaciones podrían cambiar en el futuro, la aplicación sigue siendo valiosa para mantener una administración eficiente. Con un enfoque específico en microempresas dirigidas por individuos bajo el régimen RIMPE negocios populares.

Delimitación

Esta aplicación es un prototipo en el cual se busca integrar la inteligencia artificial en una arquitectura de microservicios alojada en la nube.

Objetivos Generales y Específicos.

Objetivo General

Desarrollar un software de facturación con inteligencia artificial mediante el uso de microservicios en la nube de AWS enfocado para microempresas en Ecuador.

Objetivos Específicos

Definir los requisitos funcionales y no funcionales necesarios para el desarrollo del software contable inteligente.

Diseñar la arquitectura del software basado en microservicios, considerando la integración de servicios de AWS y su funcionamiento en plataformas web y móviles.

Integrar la inteligencia artificial en el software, permitiendo la automatización para la identificación, reconocimiento y clasificación de datos contables como la clasificación del IVA en base al producto.

Realizar pruebas de funcionamiento del prototipo.

CAPÍTULO I

FUNDAMENTOS TEÓRICOS

1.1. Microempresas en Ecuador

Las microempresas en Ecuador están creciendo notablemente, pero muchos de sus dueños carecen de conocimientos financieros y contables, lo que subraya la necesidad de herramientas estratégicas para ofrecer información coherente sobre su situación económica. (Crespo et al., 2020)

1.2. Contabilidad en las Microempresas

El Instituto Nacional de Economía Popular y Solidaria (IEPS) enfatiza la importancia de un desarrollo estratégico continuo para las microempresas, dada su rápida expansión (Crespo et al., 2020). Además, el 50,03% de las MiPymes en Ecuador carecen de un software informático para la facturación, dificultando la gestión y comprobación del rendimiento. (Arellano Cepeda et al., 2017)

1.3. Facturación electrónica en Ecuador

La obligatoriedad de la facturación electrónica en Ecuador desde el 29 de noviembre de 2022 tiene múltiples ventajas: ahorro de costos, reducción del impacto ambiental, agilidad en procesos de recepción, incremento de productividad, mayor eficiencia operativa, minimización de errores y acortamiento de plazos. (Becerra & Ojeda, 2022)

1.4. Software de facturación para microempresas

La adopción de un software contable adecuado permite a las microempresas tener un control preciso sobre sus operaciones diarias, fortaleciendo su capacidad competitiva a través de la transparencia en sus estados financieros y una toma de decisiones gerenciales más dinámica (Crespo et al., 2020)

1.5. Serverless

Las arquitecturas serverless son un modelo en el que los servidores tradicionales pasan a un segundo plano para el desarrollador, permitiendo que el código se ejecute en entornos sin estado y efímeros, gestionados por proveedores de la nube como AWS Lambda, Azure Functions, Cloud Functions de Google, IBM o Alibaba. Esto implica que el programador solo inserta su código en una función pública, dejando que el proveedor maneje toda la infraestructura. Este enfoque permite un escalado flexible y se paga únicamente por el uso de las funciones, a diferencia de mantener un servidor activo constantemente, lo que ahorra recursos y costos. (Encinas Cortés, 2020)

1.5.1. Amazon Web Services

Amazon Web Services o AWS, es un proveedor de servicios en la nube, siendo una de las plataformas más completas y ampliamente utilizadas a nivel mundial. Ofrece tanto infraestructura y plataforma como servicio (IaaS y PaaS). A través de sus servicios, proporciona soluciones escalables en áreas como cómputo, almacenamiento, alojamiento de base de datos, análisis y varias otras funcionalidades adicionales. (AWS, 2023b)

1.5.2. AWS Amplify

Amplify representa una solución integral para el desarrollo de software web o móvil alojado en AWS. Permite la creación y hospedaje de interfaces, la incorporación de funcionalidades como autenticación y almacenamiento, la conexión a fuentes de datos en tiempo real, además de facilitar la implementación y escalabilidad, incluso para audiencias de millones de usuarios. (AWS, 2023c)

1.5.3. AWS Lambda

Es un servicio de computación en la nube, el cual nos facilita la ejecución de código sin infraestructura de servidores. Se encarga de ejecutar el código en una infraestructura altamente disponible, gestionando aspectos como el mantenimiento del servidor, el sistema operativo, la escalabilidad automática y el registro de funciones. Al organizar el código en funciones de Lambda, estas se ejecutan únicamente cuando son invocadas, escalando automáticamente según la demanda. El servicio solo cobra por el tiempo de computación utilizado, sin costos adicionales cuando el código no está en ejecución. (AWS, 2023a)

1.5.4. AWS Cognito

Es una solución de identidad diseñada para aplicaciones web y móviles que combina múltiples funciones. Sirve como un repositorio de usuarios, un servidor de autenticación y un sistema de autorización para las credenciales de acceso de AWS basadas en OAuth 2.0. Ofrece

la capacidad de autenticar y autorizar usuarios desde un directorio de usuarios integrado, directorios empresariales y a través de proveedores de terceros como Google y Facebook. (AWS, 2023a)

1.5.5. Amazon Relational Database Service

Es un servicio ofrecido por AWS que ayuda en la simplificación de la gestión, configuración y escalabilidad de bases de datos relacionales en la nube. Permite la capacidad de almacenamiento adaptable y rentable en bases de datos relacionales estándar. A su vez es encargado de la administración de estas mismas. (AWS, 2023a)

1.5.6. AWS Elastic Beanstalk

Elastic Beanstalk simplifica y agiliza la implementación y gestión de aplicaciones en la nube de AWS, liberando al usuario de la preocupación por la infraestructura subyacente. Este servicio automatiza tareas como la capacidad de aprovisionamiento, balanceo de carga de trabajo, escalado y monitoreo de la aplicación, reduciendo la complejidad sin limitar las opciones ni el control del usuario. (AWS, 2023a)

1.5.7. AWS CodePipeline

Es una herramienta de entrega continua que facilita el esquematizar, visualizar y automatizar las fases requeridas para el despliegue de software. Facilita la configuración ágil de diversas etapas en el proceso de lanzamiento. CodePipeline se encarga de automatizar los pasos cruciales para implementar cambios de software de manera continua. (AWS, 2023a)

1.6. Microservicios

Los microservicios ofrecen una escalabilidad y adaptabilidad óptimas para ajustar los sistemas empresariales según las necesidades cambiantes del mercado. (Vera-Rivera et al., 2019)

1.7. Aplicaciones multiplataforma

Las aplicaciones híbridas, como las desarrolladas en React Native, combinan la versatilidad del desarrollo web con la adaptabilidad de las aplicaciones nativas en varios dispositivos, mejorando significativamente tanto el proceso de desarrollo como el rendimiento. (Quisaguano Collaguazo et al., 2022)

1.8. Inteligencia Artificial

La inteligencia artificial ha revolucionado la contabilidad al reducir el fraude, aumentar la confianza y mejorar el cumplimiento normativo a través de modelos de aprendizaje automático que automatizan tareas repetitivas y rutinarias, permitiendo un procesamiento eficiente de documentos contables. (Moncayo, 2020)

1.9. Procesamiento natural del lenguaje

El Procesamiento del Lenguaje Natural o NLP es una subdisciplina de la rama informática cuya finalidad es la comprensión del lenguaje en las computadoras de manera "natural", similar a los humanos. Normalmente, esto involucra tareas como comprender el sentimiento en textos, reconocimiento de voz y generar respuestas a preguntas. (AWS, 2023d)

1.9.1. *spaCy*

Es una biblioteca de procesamiento avanzado del lenguaje natural (NLP) en Python, disponible de forma gratuita y con código abierto. Su enfoque se orienta principalmente hacia el uso en entornos de producción, facilitando la creación de aplicaciones capaces de procesar y "entender" extensos textos. Su aplicación abarca varios campos desde el desarrollo de modelos para la extracción de información hasta la comprensión del lenguaje natural, así como ser usado para el preprocesamiento de texto en aplicaciones de aprendizaje profundo. (SPACY Usage Documentation, 2023)

1.9.2. *Expresiones regulares*

Las Expresiones Regulares (ER) son herramientas flexibles y eficientes para procesar texto. Por ejemplo, al buscar palabras en un texto extenso, considerando variaciones en los caracteres, se emplean meta caracteres, generalmente paréntesis, para delimitar las variantes posibles. Una ER se define como una forma abreviada para representar patrones de cadenas de caracteres. Al utilizar ER, se definen los caracteres de un alfabeto específico, llamados literales o caracteres normales, junto con operadores, conocidos como meta caracteres. Mediante estos, se pueden definir secuencias de caracteres más complejas para analizar mensajes escritos y verificar reglas gramaticales y ortográficas en español, siendo una herramienta útil para aprendices del idioma, especialmente cuando se implementa en aplicaciones. Esto facilita el aprendizaje asistido por dispositivos digitales, permitiendo buscar secuencias que puedan contener caracteres alternativos de manera indistinta mediante el uso de meta caracteres que señalan las posibles alternancias en la palabra buscada. (Solis Galindo & Lezama León, 2019)

1.9.3. *Pandas*

Cuando hablamos de ciencia de datos en Python, la biblioteca “**pandas**” es prácticamente omnipresente. Está construida sobre la base de la biblioteca NumPy, que nos permite realizar operaciones matemáticas eficientes en matrices de datos de un solo tipo. Pandas expande esto a dataframes, que pueden considerarse como tablas de datos. (Miller, 2018)

1.9.4. *Tf-idf*

El TF-IDF (Frecuencia del Término - Frecuencia Inversa del Documento) es un indicador que posibilita la creación de modelos avanzados de recuperación de información de tipo vectorial. Este método se distingue por representar conceptos mediante la vectorización de una colección, lo que genera una matriz de datos con columnas equivalentes a los términos y filas que corresponden a los documentos. (Cahyani & Patasik, 2021)

1.9.5. Naive Bayes para clasificación de textos

El clasificador Naive Bayes es ampliamente utilizado en problemas de NLP. Su finalidad es predecir la etiqueta o categoría de una cadena de texto. Calcula la probabilidad de cada etiqueta para un texto dado y luego produce la etiqueta con la probabilidad más alta. (Ruan et al., 2020)

1.9.6. Split de datos, Cross-validation

Cross-validation es una técnica de muestreo de datos que es usado para calificar el rendimiento en modelos predictivos y evitar el sobreajuste. Consiste en separar el conjunto de datos disponible en dos subconjuntos con la finalidad de entrenamiento y prueba de una manera sistemática. Su objetivo principal es validar el desempeño del modelo en datos no vistos, asegurando que este no dependa excesivamente del conjunto de entrenamiento y pueda realizar predicciones precisas en instancias nuevas y no observadas. (Seraj et al., 2022)

1.9.7. Multinomial Naive Bayes Sklearn

El clasificador Naive Bayes multinomial es apto para la categorización que involucra características discretas, como la frecuencia de palabras en la clasificación de texto. Este tipo de distribución multinomial suele necesitar recuentos de características expresados como números enteros. No obstante, en la aplicación práctica, incluso recuentos fraccionarios, como tf-idf, pueden ser eficaces. (Scikit Learn, 2023)

1.10. Metodología cascada

Un proceso lineal e iterativo basado en la verificación de cada fase considerando su predecesor, para poder analizar y mejorar el resultado final. El modelo en cascada según (Hughey, 2017) se compone de las siguientes fases:

1.10.1. Requerimientos del sistema

En esta fase se busca documentar lo necesario para el desarrollo del sistema, por ende, se deben definir los requisitos previos en el proyecto.

1.10.2. Diseño del Sistema

Principalmente se centra en el desarrollo de todo lo relacionado con la arquitectura del proyecto dados los requerimientos se estudia y se analiza el hardware, software necesario para el desarrollo.

1.10.3. Implementación

Se desarrollan las funcionalidades en pequeños bloques que servirán para integrarse en estas fases, pero estos pequeños bloques de código deberán analizarse y testearse para garantizar su funcionalidad.

1.10.4. Integración y Prueba

Las funcionalidades previas se integran en un solo bloque de prueba para conocer posibles fallos, vulnerabilidades.

1.10.5. Despliegue del sistema

Cuando los requisitos funcionales y no funcionales se encuentren completos, se someterán a una prueba de funcionamiento general como prototipo para que se puedan hacer cambios a futuro.

1.10.6. Mantenimiento

Si a lo largo de la prueba como prototipo se presentan diferentes fallos estos se corregirán como también posibles cambios.

1.11. Metodología CRISPDM

La metodología CRISP-DM es ampliamente adoptada en la actualidad en proyectos relacionados a la ciencia de datos. Su inicio se registra en 1997 con el respaldo financiero del Programa de Investigación y Desarrollo en Tecnologías de Información de la Unión Europea (ESPRIT). Actualmente, aunque IBM es la principal promotora de esta metodología, es implementada por profesionales en minería de datos e incluso por académicos que investigan en este ámbito de la ciencia de datos. (Espinosa Zúñiga, 2020)

1.11.1. Comprensión del negocio

En el proceso de comprensión del problema o negocio, se establece la base fundamental para el éxito del proyecto. Esta etapa se divide en actividades principales que incluyen la identificación precisa del problema, la delimitación de la problemática y la comprensión de los requisitos, supuestos, restricciones y beneficios del proyecto en cuestión. Además, se determinan los objetivos clave que se pretenden alcanzar mediante la aplicación de esta metodología.

1.11.2. Comprensión de datos

En la etapa de Comprensión de Datos, se llevan a cabo actividades clave para garantizar una visión clara y completa sobre los datos a utilizar. La recopilación de datos se enfoca en obtener la información necesaria para el proyecto, detallando fuentes, técnicas empleadas en su recopilación, y problemas enfrentados en este proceso, así como las soluciones aplicadas para superar estos obstáculos. Una vez recopilados, se procede a describir los datos, identificando su tipo, formato, volumen y significado. Esta etapa también implica la exploración de los datos mediante pruebas estadísticas básicas, lo que permite adquirir un entendimiento inicial de sus propiedades y su comportamiento general.

1.11.3. Preparación para uso de datos

Esta fase es donde se ejecutan procesos fundamentales que requieren una inversión significativa de tiempo en el proyecto. En esta etapa, se seleccionan y se transforman los datos según lo descubierto en la etapa previa para que puedan ser utilizados en el modelado. Entre las actividades principales, se encuentra la limpieza de datos, que incluye la normalización, la discretización de campos numéricos, el manejo de valores faltantes, la eliminación de valores duplicados y la atribución de datos. Asimismo, se crean marcadores que enriquezcan la capacidad predictiva de los datos existentes, facilitando la detección de patrones relevantes para el modelado. El cambio de datos adapta el estilo o estructura de algunos datos sin alterar su definición, permitiendo la aplicación de técnicas específicas en el modelado.

1.11.4. Modelado

En la etapa de Modelado se crea el modelo. Esto implica una serie de actividades clave. En primer lugar, elige el método más adecuado para el desarrollo, considerando el problema en cuestión, la data, las herramientas de minería y la comprensión del método seleccionado. Luego, en ciertos modelos, se dividen los datos en subconjuntos de entrenamiento y prueba para obtener un modelo más preciso sin llegar al sobre ajuste. Finalmente, se genera el mejor modelo posible a través de un proceso de ajuste de parámetros iterativamente.

1.11.5. Evaluar los modelos

En esta fase se evalúa el modelo mediante métricas estadísticas específicas. Estas métricas ayudan a comparar los resultados obtenidos con resultados anteriores o con el análisis de expertos en el campo del problema. Basándose en estos resultados, se decide si se continúa con la fase final de la metodología, si se regresa a alguna etapa anterior para realizar ajustes o si se comienza de nuevo con un enfoque diferente.

1.11.6. Despliegue

El despliegue o implementación del modelo pone en práctica el conocimiento derivado del modelo, realizando acciones concretas para su ejecución. Es crucial documentar de manera clara los resultados para los usuarios finales y asegurarse de una documentación completa de todas las etapas del proceso. Esta documentación generada nos puede facilitar la revisión del proyecto a posteriori y la identificación de oportunidades de mejora descubiertas. Además, se debe monitorear el proceso implementado para detectar oportunidades de mejora o posibles nuevos desafíos.

CAPÍTULO II

MARCO METODOLÓGICO

Se define el proceso de especificación de requerimientos, los cuales son tomados en cuenta para el desarrollo del software. El siguiente paso es el modelado de los procesos clave del sistema, cuya finalidad es ofrecer una mejor perspectiva y por último el uso de las metodologías para el desarrollo.

2.1. Especificación de Requerimientos

El proceso busca identificar y documentar los requerimientos de las partes interesadas, es importante hacerlo ya que permite entender mejor el objetivo general del proyecto y ajustar el desarrollo de este.

2.1.1. Introducción

Este punto aborda los requerimientos para el desarrollo los cuales se mantendrán como brújula para alcanzar los objetivos planteados para con el proyecto. Estos abordan brevemente las funcionalidades esperadas, el resultado deseado y la perspectiva de la aplicación.

2.1.1.1. Propósito

Este documento pretende ser una guía de requerimientos que se deben cumplir en el desarrollo de este proyecto. Su uso se orienta al equipo de desarrollo. El software tiene como objetivo la creación de comprobantes electrónicos, específicamente facturas, automatizar acciones como la generación de datos contables como el cálculo del IVA, registrar clientes, proveedores y productos, búsqueda mediante palabras para documentos y personas, por último, ver sus detalles correspondientes.

2.1.1.2. Alcance

El diseño del software se basa en la premisa de ser usado como una solución para la venta en microempresas del Ecuador, así como brindar los datos relevantes para las mismas. El usuario destino es la denominada persona natural que se encuentra en el régimen RIMPE negocios populares.

2.1.1.3. Definiciones, acrónimos y abreviaturas

SRI: Servicio de Rentas Internas del Ecuador

BD: Base de datos.

IVA: Impuesto sobre el valor agregado.

API: Interfaz de programación de aplicaciones.

IA: Inteligencia artificial.

2.1.2. Descripción general del sistema

2.1.2.1. Perspectiva

La aplicación esta planificada para desarrollarse mediante una arquitectura de microservicios que trabajarán en la nube del proveedor AWS. Los servicios que se han hecho uso de AWS son:

- Para la autenticación y registro de usuarios, Cognito.
- Base de datos relacional, Amazon RDS.
- GitHub, GitLab y AWS CodePipeline
- AWS Lambda, AWS Api Gateway y AWS Elastic Beanstalk.
- AWS Amplify para el alojamiento.

2.1.2.2. Funcionalidades

El software permitirá emitir los siguientes documentos electrónicos:

- Facturas

Podrán gestionar:

- Clientes y proveedores.
- Productos, bienes y servicios.
- Información del usuario

2.1.2.3. Características de los usuarios

Tipo de usuario	Persona natural que se clasifica para el régimen RIMPE negocios populares
Formación	Ninguna
Habilidades	Conocimiento básico de como interactuar con un teléfono inteligente o computador
Actividades	Limitado a negocios populares

2.1.2.4. Restricciones

El desarrollo hace uso del framework React Native, Python, GitHub, AWS y sus componentes. Cumpliendo con las metodologías previamente propuestas.

2.1.2.5. Evolución a futuro del software

2.1.3. Requisitos específicos

2.1.3.1. Requisitos funcionales

Código	Requerimiento Funcional 1
Nombre	Registrar usuarios
Prioridad	Esencial

Descripción	La aplicación permite el registro de nuevos usuarios del aplicativo. Los campos para proporcionar son: RUC, correo, nombre, apellido y contraseña.
-------------	--

Código	Requerimiento Funcional 2
Nombre	Registrar clientes
Prioridad	Esencial
Descripción	La aplicación permite el registro de nuevos clientes. Los campos para proporcionar son: CI o RUC, nombre, correo, dirección y teléfono.

Código	Requerimiento Funcional 3
Nombre	Registrar proveedores
Prioridad	Esencial
Descripción	La aplicación permite el registro de nuevos proveedores. Los campos para proporcionar son: RUC, nombre, nombre comercial, correo, dirección y teléfono.

Código	Requerimiento Funcional 4
Nombre	Registrar productos
Prioridad	Esencial
Descripción	La aplicación permite el registro de nuevos productos. Los campos para proporcionar son: Nombre producto, proveedor, código del proveedor, categoría, código interno, precio, precio sugerido, stock, unidad y tipo.

Código	Requerimiento Funcional 5
Nombre	Generar factura
Prioridad	Esencial
Descripción	La aplicación permite generar nuevas facturas. Los campos para proporcionar son: Fecha, datos cliente, productos y método de pago.

Código	Requerimiento Funcional 6
Nombre	Registrar cuentas contables
Prioridad	Esperado
Descripción	La aplicación permite el registro de nuevas cuentas contables. Los campos para proporcionar son: Nombre de cuenta y padre de la cuenta.

Código	Requerimiento Funcional 7
Nombre	Registrar libro diario
Prioridad	Esperado
Descripción	La aplicación permite el registro de nuevos asientos contables. Los campos para proporcionar son: fecha, numero de documento, cuenta, debe, haber y total.

Código	Requerimiento Funcional 8
Nombre	Registrar libro mayor
Prioridad	Esperado
Descripción	La aplicación permite el registro de saldos en el libro mayor. Los campos para proporcionar son: Cuenta y saldos.

2.1.3.2. Limitantes

- No autorizara las facturas con el SRI.
- No bajara facturas del SRI para su registro.
- Generación de reportes de ventas compras para realizar análisis de productividad.
Soporte técnico del sistema.
- No enviara las facturas generadas por correo electrónico

2.1.3.3. Requisitos no funcionales

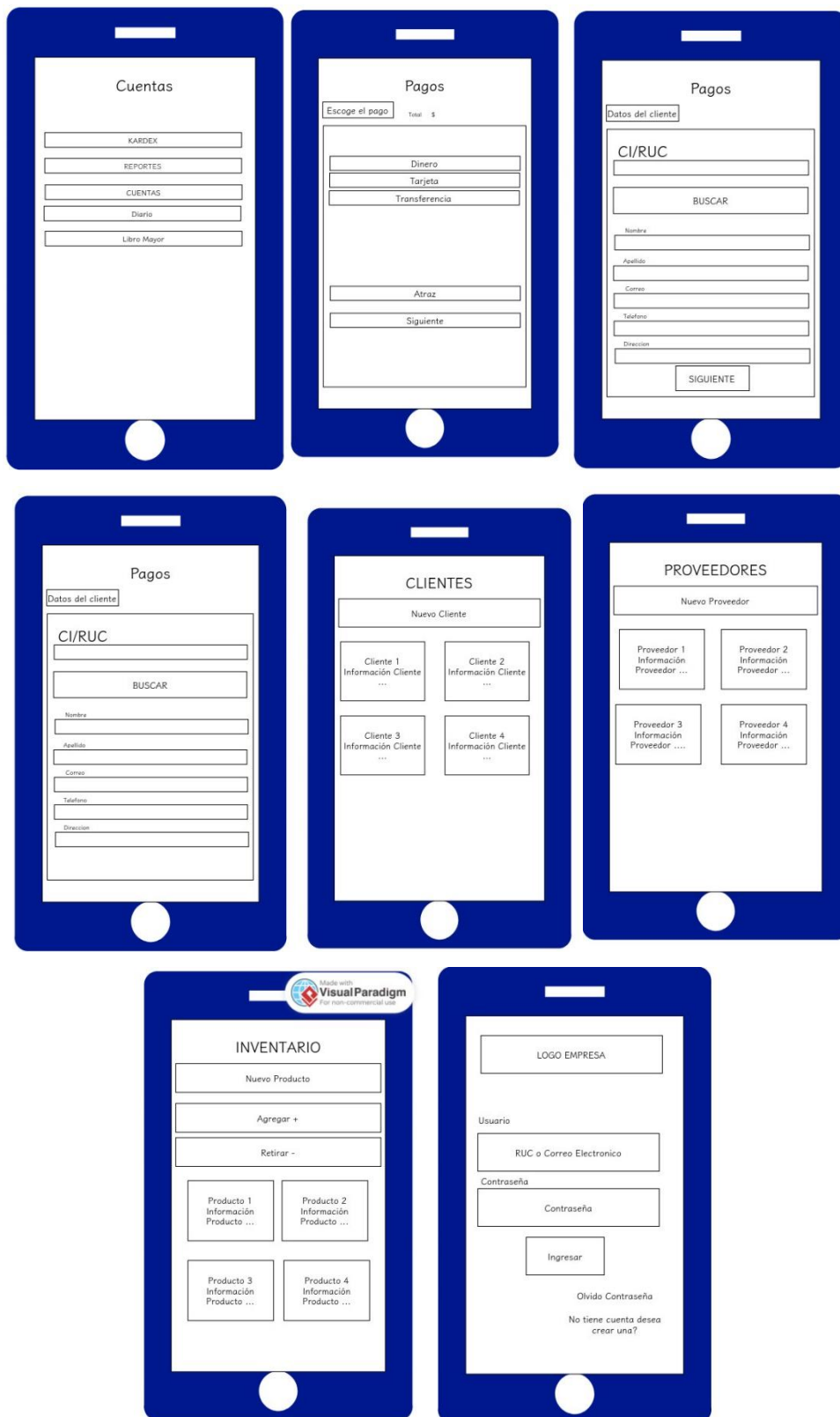
Código	Requerimiento no funcional 1
Nombre	Funcionar 24/7
Prioridad	Esperado
Descripción	La aplicación funcionara de manera ininterrumpida.

Código	Requerimiento no funcional 2
Nombre	Desarrollo en frameworks correspondientes
Prioridad	Esperado
Descripción	La aplicación será desarrollada con los lenguajes electos y sus respectivos frameworks.

2.1.3.4. Interfaces de usuario

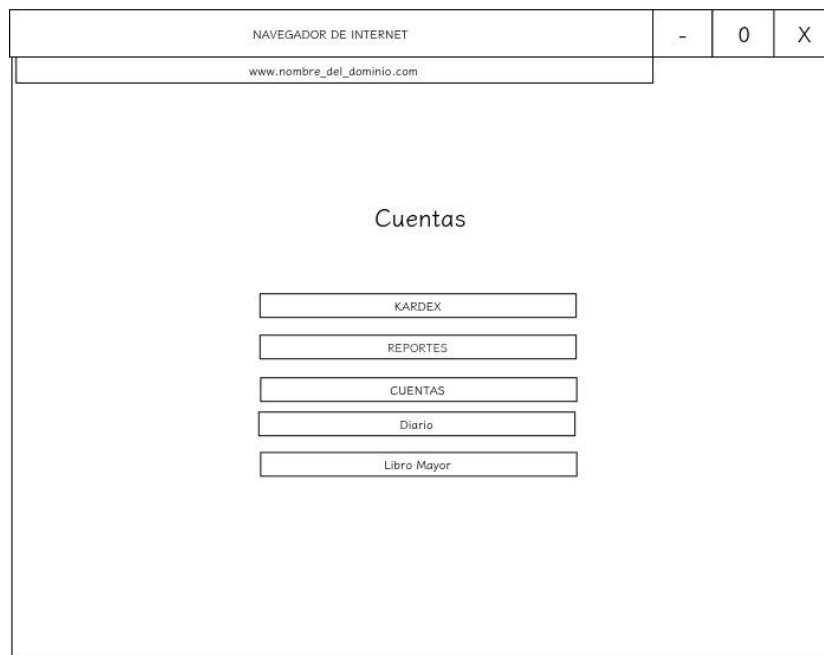
Este punto contiene prototipos de interfaces para el usuario. Estas se tuvieron en cuenta para el desarrollo de la aplicación, al ser multiplataforma el resultado final puede variar en base a cuestiones como el dispositivo y su resolución de pantalla.

Figura 1: Interfaces móviles



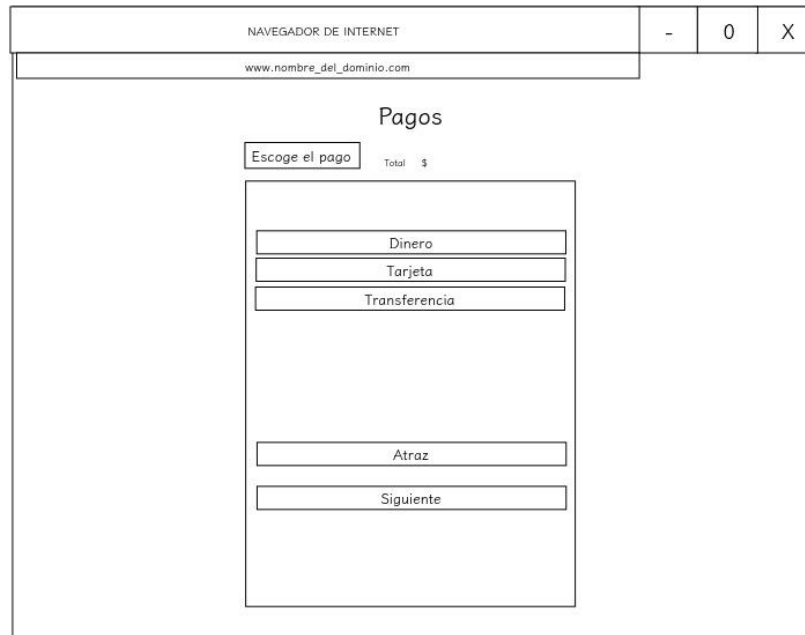
Nota. Prototipos de interfaces para la plataforma móvil. Elaborado por: Los autores.

Figura 2: Interfaz Contabilidad



Nota. Conteo de valores únicos de consumo masivo. Elaborado por: Los autores.

Figura 3: Interfaz plataformas de pago



Nota. Prototipo de interfaz para las formas de pago. Elaborado por: Los autores.

Figura 4: Interfaz datos del cliente para la factura.

Este prototipo de interfaz de usuario muestra una ventana de navegador de Internet con el título "NAVEGADOR DE INTERNET" y la URL "www.nombre_del_dominio.com". El contenido principal de la página es el título "Pagos". Debajo de este título, hay un recuadro con el título "Datos del cliente". Dentro de este recuadro, se encuentra un formulario con los siguientes campos:

- CI/RUC: un campo de entrada de texto.
- BUSCAR: un botón de acción.
- Nombre: un campo de entrada de texto.
- Apellido: un campo de entrada de texto.
- Correo: un campo de entrada de texto.
- Telefono: un campo de entrada de texto.
- Direccion: un campo de entrada de texto.

Nota. Prototipo de interfaz para la plataforma web. Elaborado por: Los autores.

Figura 5: Interfaz lista de clientes

Este prototipo de interfaz de usuario muestra una ventana de navegador de Internet con el título "NAVEGADOR DE INTERNET" y la URL "www.nombre_del_dominio.com". El contenido principal de la página es el título "CLIENTES". Debajo de este título, hay un botón "Nuevo Cliente". A continuación, se muestran cuatro tarjetas de cliente, cada una con el siguiente contenido:

- Cliente 1: Información Cliente ...
- Cliente 2: Información Cliente ...
- Cliente 3: Información Cliente ...
- Cliente 4: Información Cliente ...

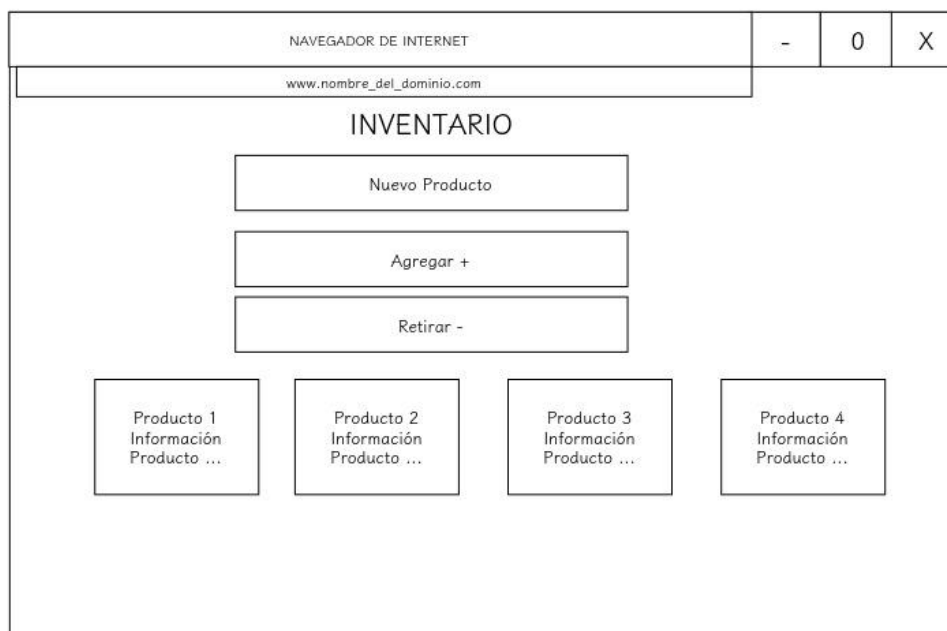
Nota. Prototipo de interfaz para web sobre la lista de clientes. Elaborado por: Los autores.

Figura 6: Interfaz proveedores



Nota. Prototipo de interfaz para web sobre lista de proveedores. Elaborado por: Los autores.

Figura 7: Interfaz de inventario



Nota. Prototipo de interfaz para web del inventario. Elaborado por: Los autores.

Figura 8: Interfaz inicio de sesión

The image shows a wireframe of a login interface within a web browser window. The browser's title bar reads "NAVEGADOR DE INTERNET" and the address bar contains "www.nombre_del_dominio.com". In the top right corner, there is a logo for "VisualParadigm" with the text "Made with VisualParadigm For non-commercial use". The main content area is centered and contains the following elements from top to bottom: a rectangular box labeled "LOGO EMPRESA"; the label "Usuario" followed by a rectangular input field containing the text "RUC o Correo Electronico"; the label "Contraseña" followed by a rectangular input field containing the text "Contraseña"; a rectangular button labeled "Ingresar"; and finally, the text "Olvido Contraseña" and "No tiene cuenta desea crear una?" centered below the button.

Nota. Prototipo de interfaz para web sobre el inicio de sesión. Elaborado por: Los autores.

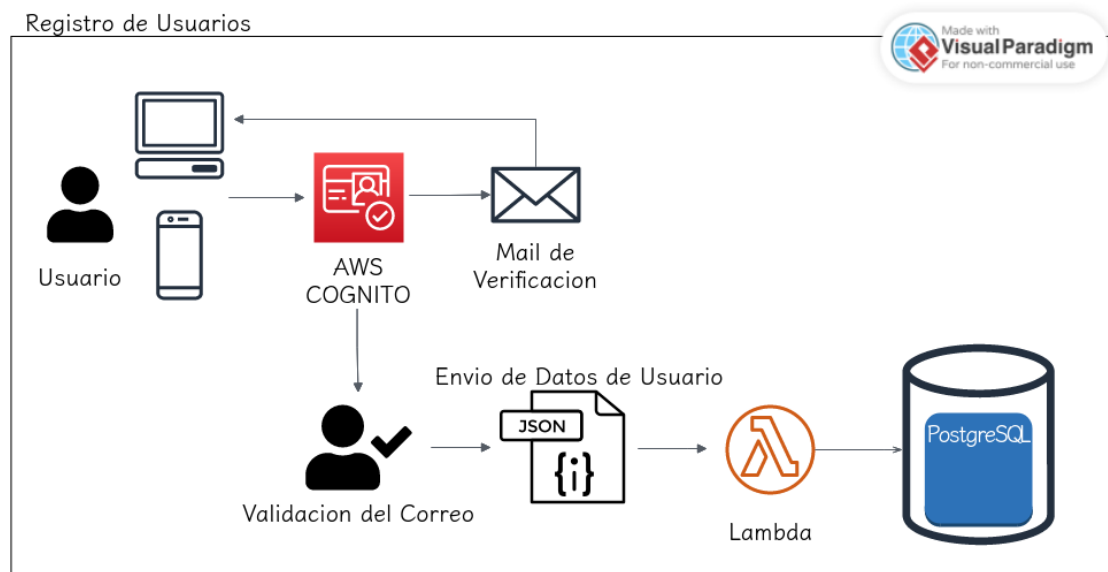
2.2. Modelado de Sistema

El modelado hace posible la visualización clara de cada proceso involucrado en el software, actúa como guía en el desarrollo de este mismo. En los siguientes apartados, se muestran diagramas de cada proceso clave de la aplicación.

2.2.1. Registro de usuarios

Se presenta el diagrama acerca del proceso de crear un nuevo usuario en la aplicación.

Figura 9: Diagrama registro de usuarios



Nota. Diagrama sobre el proceso para el registro de usuarios. Elaborado por: Los autores.

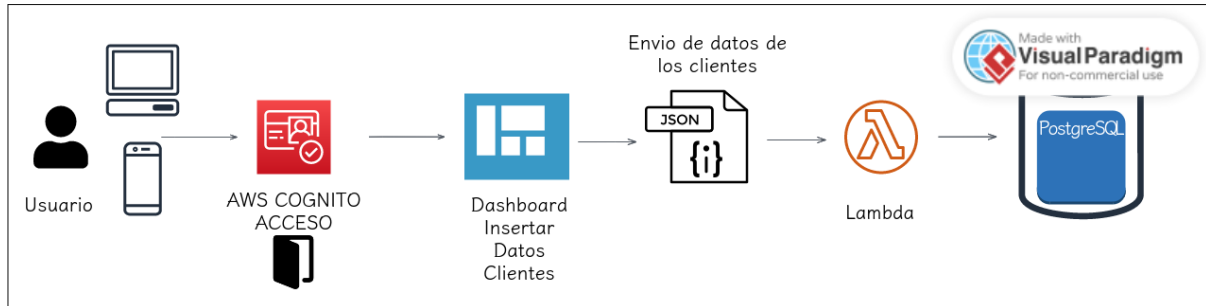
El proceso de registro cuenta con las siguientes fases:

- El usuario se registra con su respectiva información de la empresa
- Cognito recibe esta información y envía un mail de verificación con el correo registrado
- Una vez registrado el código se valida el usuario
- Finalmente se crea el esquema individual en la base de datos del usuario mediante lambda con los datos recibidos del usuario registrado

2.2.2. Registro de clientes

Se presenta el diagrama acerca del proceso de registrar un nuevo cliente en la aplicación.

Figura 10: Diagrama registro de clientes



Nota. Diagrama sobre el proceso para registro de clientes. Elaborado por: Los autores.

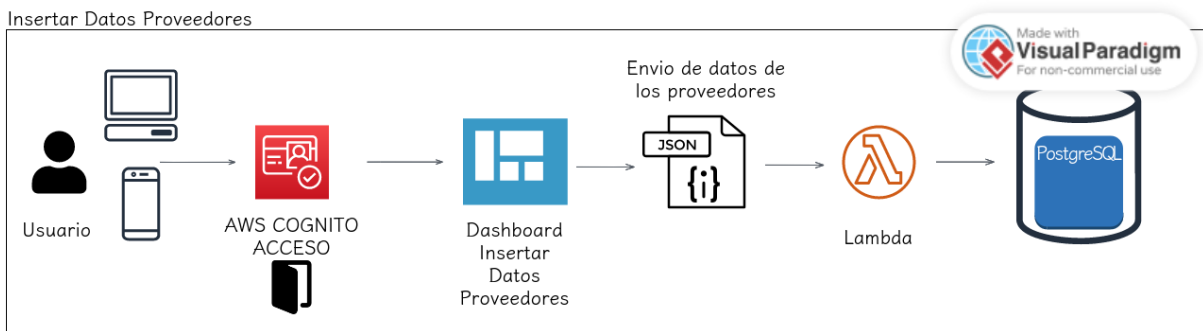
El registro de clientes presenta el siguiente proceso:

- Una vez accedemos al sistema nos dirigimos a la página principal para insertar clientes
- Llenar los datos pertenecientes al cliente y finalmente enviar
- La información será enviada en formato JSON hacia lambda para luego ser insertada en BD.
- Finalmente, en la página de clientes podremos observar el cliente que se acaba de ingresar.

2.2.3. Registro de proveedores

Se presenta el diagrama acerca del proceso de registrar un nuevo proveedor en la aplicación

Figura 11: Diagrama registro de proveedores



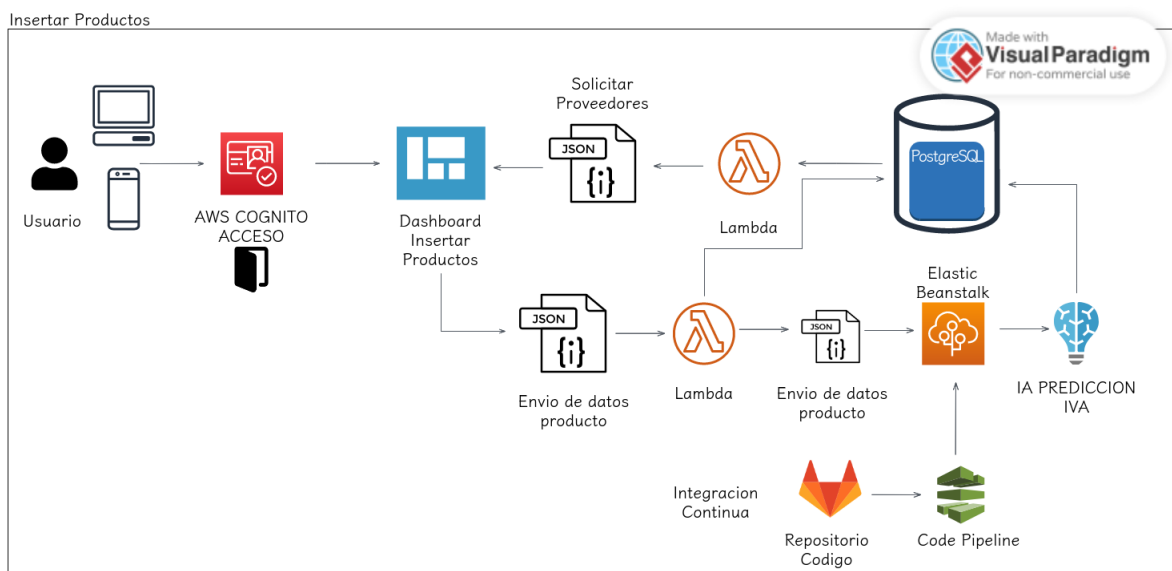
Nota. Diagrama del proceso para el registro de proveedores. Elaborado por: Los autores.

El registro de proveedores funciona de la misma manera que registro clientes el cual fue explicado previamente.

2.2.4. Registro de productos

Se presenta el diagrama acerca del proceso de registrar un nuevo producto en la aplicación

Figura 12: Diagrama para el registro de productos



Nota. Diagrama sobre el proceso de registro de productos en el inventario. Elaborado por: Los autores.

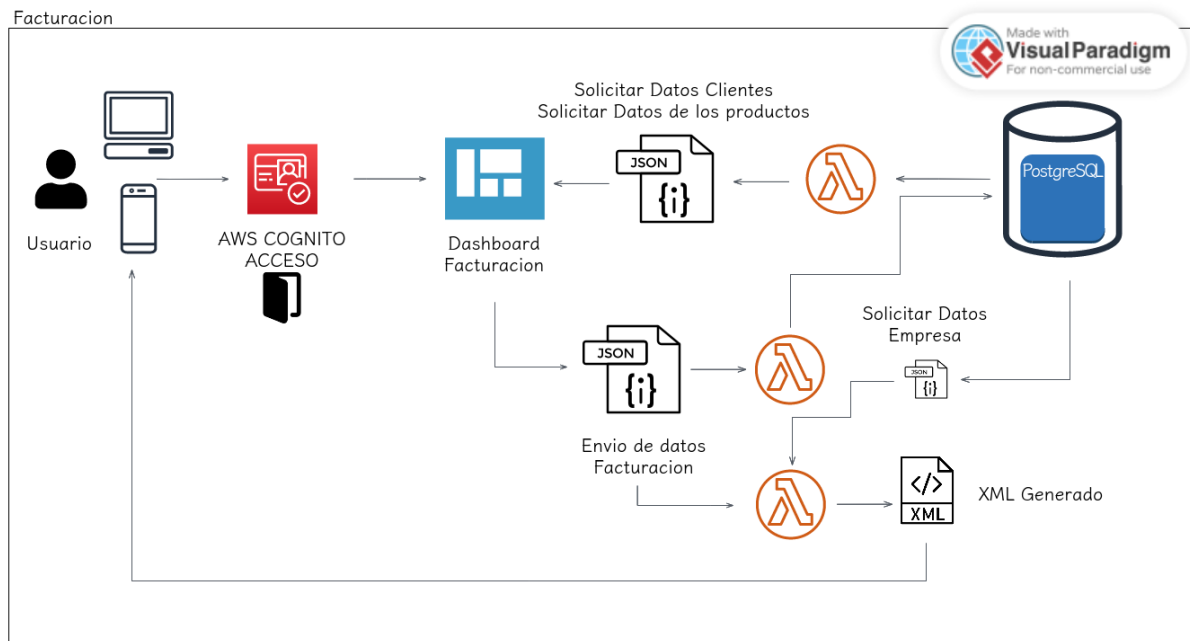
El proceso de Registro de productos consta de las siguientes etapas:

- En la página de registro de productos se llena los datos pertinentes relacionados al producto
- Si existe información dentro de la BD se mostrará los diferentes productos
- Al hacer el envío de datos primero se realiza el registro en la BD para posterior realizar la predicción mediante la Inteligencia Artificial y registrar el producto con su respectiva clase, en este caso utilizamos el api mediante Elastic Beanstalk.

2.2.5. Facturación

Se presenta el diagrama acerca del proceso de generar una nueva factura en la aplicación

Figura 13: Diagrama de la facturación



Nota. Diagrama del proceso para generar una factura. Elaborado por: Los autores.

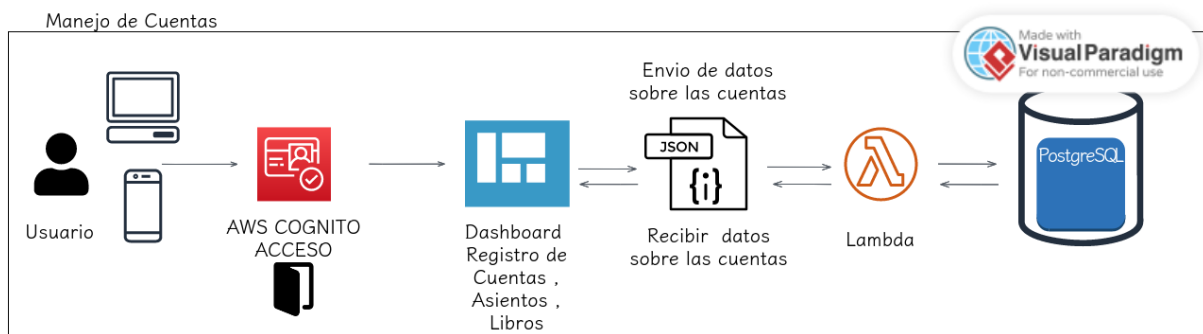
El proceso de registro de facturación cuenta de 3 funciones principalmente

- Se da el llenado de información de la factura con los productos para luego esta factura se registrada en la BD.
- La información tanto de clientes, productos o proveedores se obtiene mediante la base de datos
- Finalmente, para la generación del XML usamos igualmente una función que recibe los datos de la factura además de la empresa para luego devolver al usuario el formato que maneja el SRI.

2.2.6. Registrar cuentas contables

Se proporciona el diagrama relativo al procedimiento de registro de una nueva cuenta contable en la aplicación. En la gestión de cuentas contables, el proceso implica que el usuario introduce el nombre de la cuenta y selecciona la cuenta principal. Estos datos se consolidan en un formato JSON y se transmiten al API correspondiente, donde se ejecuta la función lambda encargada de almacenar la información en la BD. De no ser así y si se produce algún inconveniente, se retorna un mensaje de error.

Figura 14: Diagrama del registro de cuentas contables



Nota. Diagrama del proceso de registro de cuentas contables. Elaborado por: Los autores.

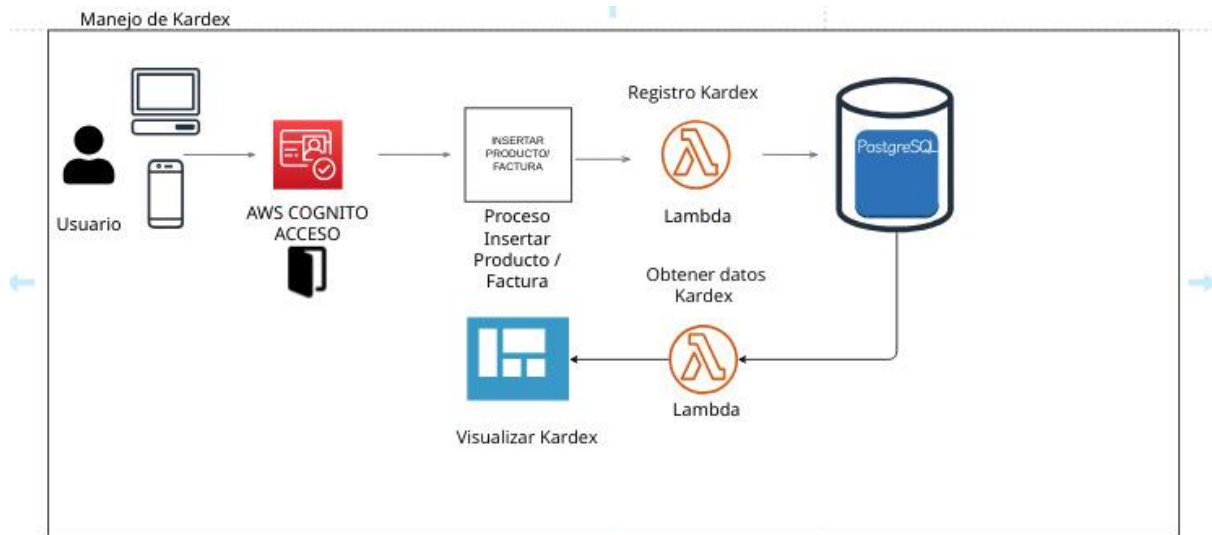
Para el caso del manejo de cuentas se tiene las siguientes consideraciones:

- Si la cuenta es algo relacionado con libro mayor, diario, cuentas o algo relacionado al Kardex, el JSON llevara en su información al hacer un proceso que tipo de operación es.
- La función lambda recibe todos los datos relacionados a la cuenta en conjunto del tipo de operación que es, a continuación, esta procesa e inserta los datos en las respectivas tablas
- Finalmente se puede visualizar en cada dashboard respectivo de las cuentas la información.

2.2.7. Registrar movimientos en el Kardex simplificado

Se presenta el diagrama acerca del proceso de registrar el movimiento de entrada o salida de productos del inventario

Figura 15: Diagrama para el registro del Kardex simplificado



Nota. Diagrama del proceso de registro de movimientos en el Kardex simplificado. Elaborado por: Los autores.

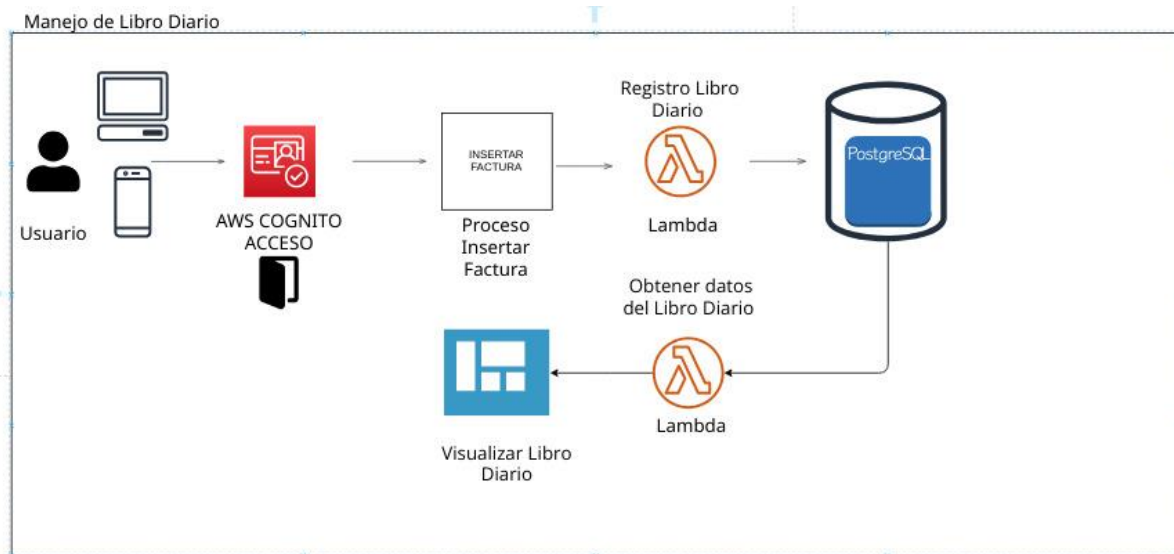
Para el caso del manejo entrada o salida de inventario de los bienes o productos se tiene las siguientes consideraciones:

- La consideración fundamental en el proceso es que se trata de una forma simplificada. Debido al público objetivo no se necesita la complejidad que un proceso de Kardex normal necesitaría, es decir que solo guardara datos como fecha, documento de ser el caso, motivo, tipo de movimiento, cantidad, precio unitario tomado del inventario y valor total.
- El usuario no influye en el proceso, dado que es automático. El proceso varía dependiendo si es salida o entrada por factura o es directamente un movimiento desde el apartado de inventario.
- Si fuera un movimiento de factura, se guardaría el número de documento como referencia y su motivo sería compra o venta dependiendo del caso.
- Si fuera entrada o salida desde el apartado inventario no hay necesidad de número de documento.

2.2.8. Registrar datos en el libro diario

Se presenta el diagrama del proceso de registro de los asientos contables en el libro diario

Figura 16: Diagrama del registro en el libro diario



Nota. Diagrama del proceso para registrar movimientos en el libro diario. Elaborado por: Los autores.

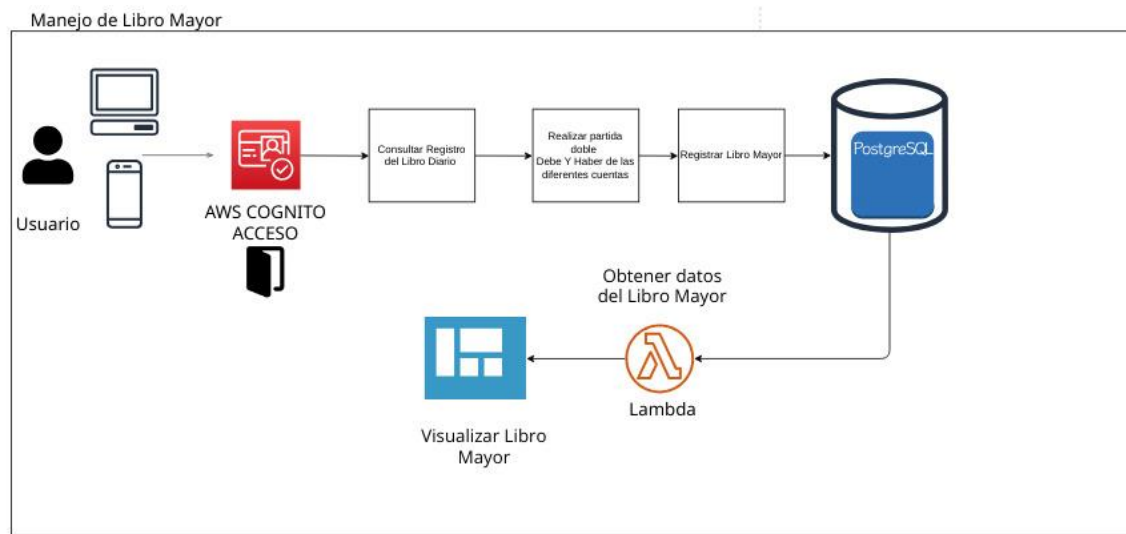
El proceso para el registro en el libro diario tiene las consideraciones:

- Se hace automáticamente sin intervención del usuario, esto solo se realiza después de generar una factura, se toma de las cuentas contables almacenadas para registrar la data contable necesaria.
- Debido a la limitación de ser un prototipo, no registrara un asiento contable en el caso cuando se recibe una factura.
- El proceso de cierre del libro es diario.

2.2.9. Registrar datos en el libro mayor

Se presenta el diagrama sobre el flujo del proceso para el registro de datos en el libro mayor

Figura 17: Diagrama del registro en el libro mayor



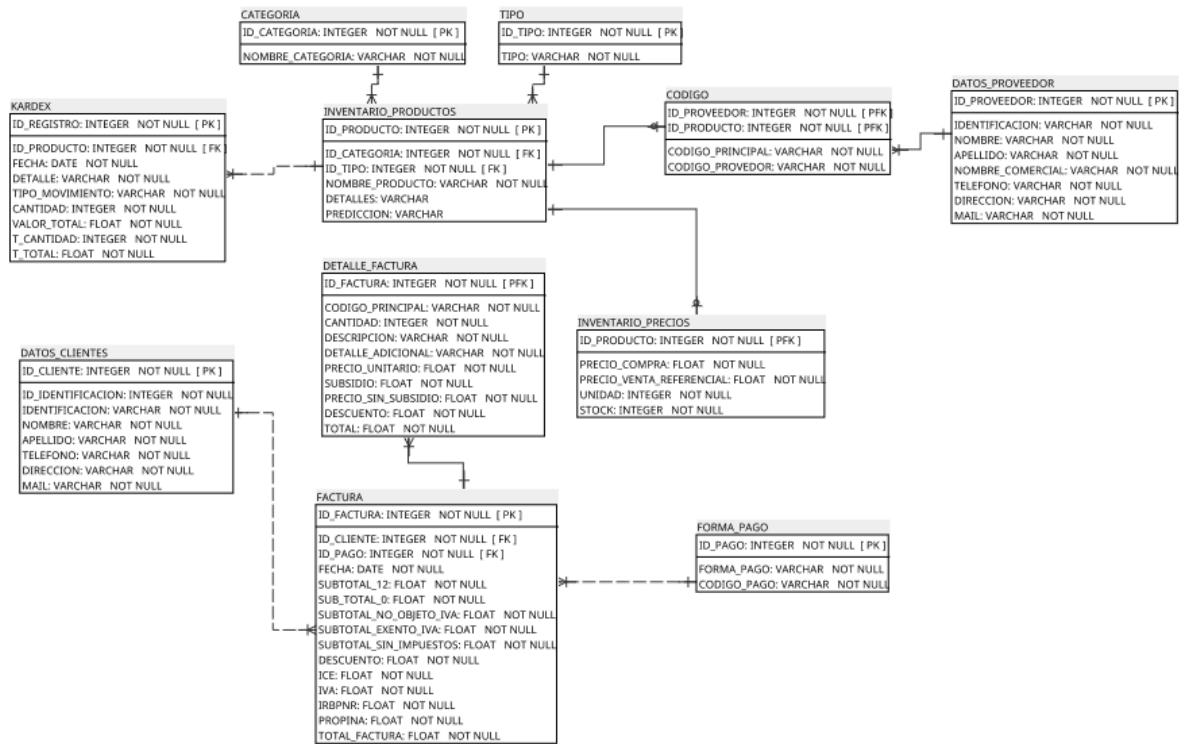
Nota. Diagrama de proceso de registro de movimientos en el libro mayor. Elaborado por: Los autores.

Para este proceso se toman las siguientes consideraciones:

- Al igual que los procesos anteriores es automático y no requiere intervención del usuario.
- Este proceso se ejecuta después de registrar un nuevo registro en el diario. Al ser un libro mayor mantiene los balances de las cuentas contables.
- El proceso de cierre se realizará de manera mensual.
- Al igual que el libro diario, este no recopilara la data de facturas externas ingresadas, debido a la limitante de ser un prototipo.
- Toda información es referencial y no está destinada, debido a la limitante, a ser usada en algún proceso tributario por el momento.

2.2.10. Esquema de la base de datos

Figura 18: Esquema de tablas de la BD



Nota. Diagrama del esquema de la base de datos donde se muestran las tablas y sus relaciones. Elaborado por: Los autores.

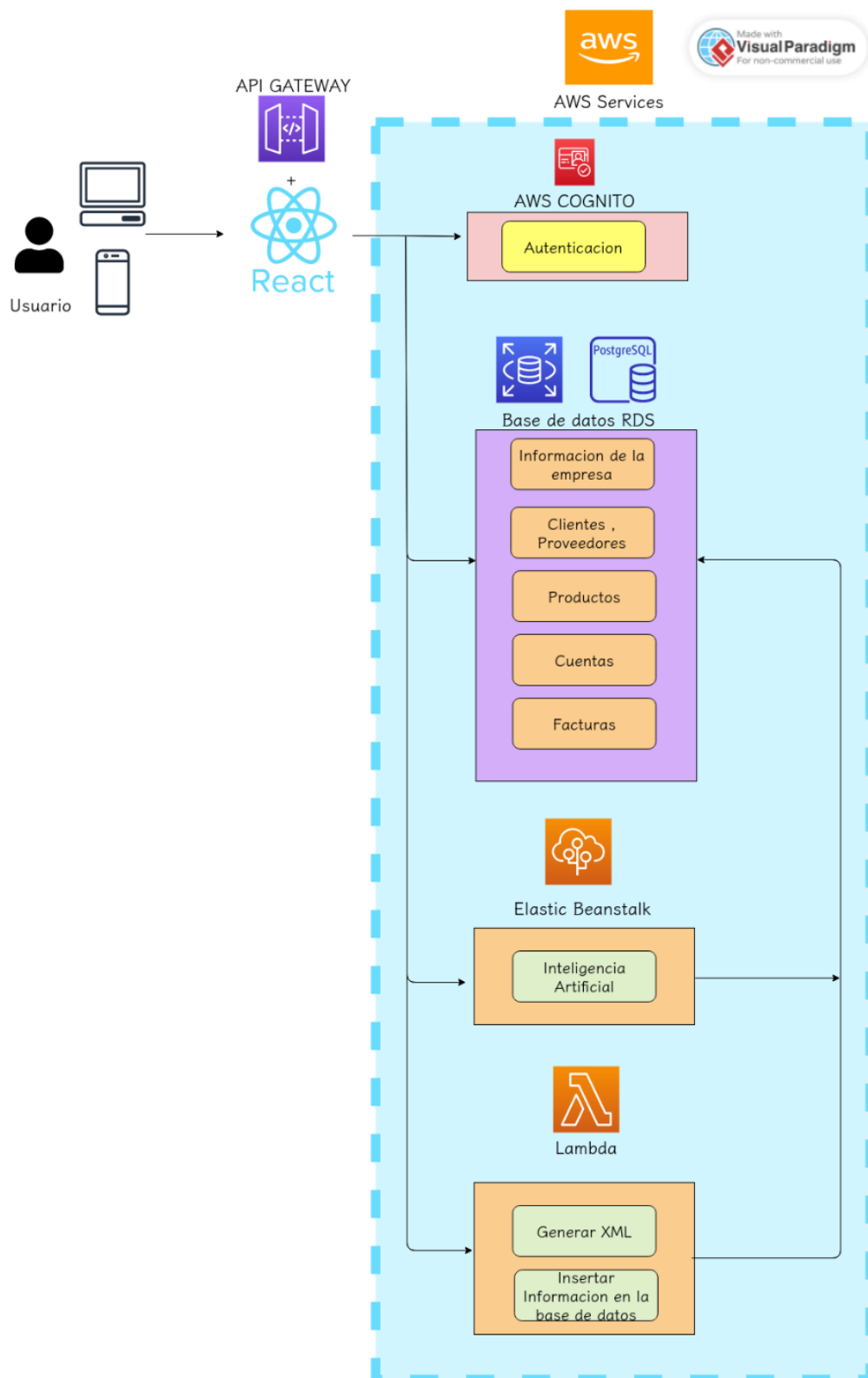
La arquitectura de la BD nos muestra tablas las cuales son una parte fundamental para el funcionamiento de la lógica del software. Aquí se almacenan datos generados por cada cliente. Esta BD trabaja bajo la premisa de múltiple esquema único, se refiere a una arquitectura que mantiene una única instancia de BD con varios esquemas. Cada esquema puede contener múltiples tablas o compartir la misma arquitectura de tablas entre diferentes esquemas. Esta implementación permite optimizar la gestión al utilizar una única arquitectura de tablas, mientras se administran varios esquemas por usuarios. Esta estrategia simplifica operaciones, agiliza la ejecución de scripts específicos para ciertos esquemas y evita la necesidad de realizar modificaciones en toda la base de datos.

2.3. Arquitectura de Sistema

La arquitectura esperada mediante la tecnología serverless de AWS con un desarrollo en microservicios. El uso de este tipo de tecnología es para concentrarse en el desarrollo y no tener que mantenerse pendiente de la infraestructura donde se ejecuta el software, además que permite el total enfoque de los desarrolladores en el proceso de desarrollo. Dependiendo de las funciones que sean requeridas y las herramientas que sean utilizadas el escalamiento y

optimización se hacen automáticamente según corresponda. Por último, permite el ahorro de costos debido a que AWS trabaja de manera que solo pagas por lo que requieras o uses. El uso de AWS y sus servicios, es debido a la gran integración, optimización, disponibilidad y escalamiento.

Figura 19: Diagrama general de la arquitectura de la aplicación



Nota. Diagrama de la arquitectura basada en microservicios de la aplicación. Elaborado por:

Los autores.

2.4. Uso de la metodología

2.4.1. Uso de metodología de datos

2.4.1.1. Comprensión del negocio

A. Objetivo del proyecto

Desarrollar un modelo de inteligencia artificial para la clasificar la existencia de IVA para los diferentes productos de un inventario mediante técnicas de procesamiento de texto.

B. Requisitos y limitaciones del negocio

El modelo debe estar pensado para microempresas del Ecuador. El modelo está limitado únicamente a productos, servicios con IVA y sin IVA. Al ser un prototipo el modelo podrá estar sujeto a cambios a futuro dependiendo el uso.

2.4.1.2. Comprensión de datos

A. Adquisición de datos

Productos de Consumo Masivo

Se recolecto información acerca de los diferentes productos que tienen más uso en este caso se filtró la información por categorías para identificar de forma más eficiente que productos tienen IVA y no. La información fue obtenida mediante Kaggle utilizando un dataset que recolecta productos de uso diario, alimentos, productos de limpieza, etc.

Productos de Ferretería

Para obtener productos que tienen IVA utilizamos datasets relacionados con materiales de construcción, herramientas y varios productos relacionados con línea blanca al igual que con productos de consumo masivo este dataset posee la suficiente información por categorías lo cual facilita el proceso de clasificación de las clases. La información fue obtenida mediante zenodo un servicio que se usa para subir información para investigación, proyectos.

Medicamentos de venta libre

En este caso se usó información de medicamentos que se extrajo de datasets abiertos del Ministerio de Salud del Ecuador, así como datos abiertos al público por del gobierno de Colombia, en este caso se usó el nombre comercial que manejan estos medicamentos y su composición para ubicarlos dentro de la clase no tienen IVA.

Productos Agrícolas y Fertilizantes

Para productos relacionados con la agricultura utilizamos datos abiertos del Ecuador en donde se puede obtener varios datasets sobre los productos más utilizados en la agricultura del

Ecuador, igualmente se utiliza el nombre comercial, la composición que maneja el producto para el modelo.

B. Exploración de datos

Para realizar la búsqueda de datos y clasificarlos dentro de las respectivas clases IVA, no IVA se utiliza funciones para el filtrado de datos en este caso utilizaremos Python ya que cuenta con librerías para un análisis de data extensa. Existen datasets que tienen subcategorías entonces mediante estas se puede asignar si poseen IVA o no los productos ya que engloba una clase general al igual que con productos que poseen IVA.

Tabla 1: Productos de consumo masivo

Nombre del Producto	6676
Nombre Completo del Producto	8017
Marca del producto	1203
Categoría	1
Subcategoría	14
Total Elementos	25638

Nota. Conteo de valores únicos de consumo masivo. Elaborado por: Los autores.

Tabla 2: Productos de ferretería

Títulos	4313
Fabricante	292
Articulo	1037
Presentación	292
Marca Comercial	72
Categoría	6
Subcategoría	70
Total Valores Únicos	5807

Nota. Conteo de valores únicos de ferretería. Elaborado por: Los autores.

Tabla 3: Medicamentos de venta libre

	Principio Activo	Concentración	Forma Farmacéutica
Conteo	960	960	960
Únicos	492	747	169
TOP	SALICILATO DE METILO+ MENTOL	10 MG	Tableta
Frecuencia	17	15	105

Nota. Conteo de medicamentos de venta libre. Elaborado por: Los autores.

Tabla 4: Fertilizantes y productos agrícolas

	Nombre del Producto	Ingrediente Activo	Grupo	Presentación	Tipo de Cultivo
Conteo	99	99	99	99	99
Únicos	48	48	6	39	25
TOP	PARAQUAT VECOL 200 SL	Paraquat 200 g/L	HERBICID AS	Frascos de 1, 4 y 20 Litros	ARROZ
Frecuencia	15	15	34	15	21

Nota. Conteo de productos agrícolas. Elaborado por: Los autores.

C. Descripción de los diferentes datos

Tabla 5: Variables de los datos sobre productos de consumo masivo

Columna	Descripción
date	Fecha de ingreso del producto
prod_id	Id del producto interno
prod_name	Nombre del producto
prod_name_long	Nombre del producto completo
prod_brand	Nombre de la marca del producto
category	Categoría del producto
Subcategory	Subcategorías de productos
Tags	Identificador del producto
prod_unit_price	Precio unitario del producto

Nota. Descripción de variables de los datos sobre productos de consumo masivo. Elaborado por: Los autores.

Tabla 6: Variables de los datos sobre productos de ferretería

Columna	Descripción
url	Información del producto
title	Nombre del producto
Price	Precio del proveedor para el producto
fabricante	El quien fabrica el producto
sku	Código del producto
Articulo	Nombre alterno al producto
Presentación	Unidad de medida
Referencia Proveedor	Código de referencia del proveedor
Diámetro	Unidad de medida relacionada al diámetro del producto
Capacidad	Volumen del producto
Categoría	Categoría en la que el producto se encuentra
Subcategoría	Subcategoría en la que el producto ha sido clasificado

Nota. Descripción de variables de los datos sobre productos de ferretería. Elaborado por: Los autores.

Tabla 7: Variables de los datos sobre medicamentos de venta libre

Columna	Descripción
principioactivo	Ingrediente principal del medicamento
concentracion	Concentración química del medicamento
formafarmaceutica	Terminología para referirse a un medicamento terminado para su dosificación

Nota. Descripción de variables de los datos sobre medicamentos de venta libre. Elaborado por: Los autores.

2.4.1.3. Preparación para uso de los datos

A. Curación y limpieza de datos

En este proceso se hace uso de expresiones regulares dentro de pandas aplicando el proceso a cada columna del dataset, además de transformar todos los datos que tengan que ver con palabras a minúscula ya que estos principalmente se usaran para el proceso NLP.

Figura 20: Expresión regular

```
'[^a-zA-Z\sáéíóúÁÉÍÓÚüÜñÑ]'
```

Nota. Expresión regular usada para el proceso de limpieza. Elaborado por: Los autores.

La siguiente expresión regular tomará todos los caracteres latinos, mayúsculas, minúsculas, eliminando caracteres especiales, puntos y respetando el espacio de la palabra. Luego para eliminar los datos nulos se transforma a lista las diferentes columnas teniendo en cuenta la condición de que sea de cierta subcategoría, esto aplicable para los dataset de ferretería y productos, en este caso se usó las siguientes condiciones.

Tabla 8: Productos escogidos por subcategoría

Contiene	Dataset
Bebidas	Productos Consumo Masivo
Charcutería	Productos Consumo Masivo
Despensa	Productos Consumo Masivo
Dulces y postres	Productos Consumo Masivo
Lácteos, huevos y refrigerados	Productos Consumo Masivo
Panadería y Pastelería	Productos Consumo Masivo

Pasabocas	Productos Consumo Masivo
Productos Congelados	Productos Consumo Masivo
Vinos y Licores	Productos Consumo Masivo
mascotas_perros_alimento_seco	Productos Consumo Masivo
mascotas_gatos_alimento_seco	Productos Consumo Masivo
Cerrajería	Ferretería
Climatización	Ferretería
Ferretería	Ferretería
Fontanería	Ferretería
Herramientas	Ferretería
Jardín y piscinas	Ferretería
Material eléctrico	Ferretería
Mobiliario Hogar	Ferretería
Para el hogar	Ferretería

Nota. Subcategoría de productos elegidos. Elaborado por: Los autores.

En el resto de los datos únicamente se escoge y se limpia con la expresión regular ya que estos se encuentran en una sola categoría por ende se puede asignar directamente a la clase sin IVA ya que involucra medicinas y productos agrícolas.

B. Transformación de los datos

El siguiente paso es escoger de cada una de las categorías valores aleatorios ya que al ser un data set bastante extenso necesitamos una cantidad pareja de datos para cada clase, escogemos los datos mediante la librería random de la siguiente forma:

Figura 21: Ecuación para elección aleatoria

```
random_sample = random.sample(datos, C)
```

Nota. Ecuación para la elección de datos de manera aleatoria. Elaborado por: Los autores.

En este caso C vendría a ser la cantidad de datos que necesitamos escoger de cada clase, el mismo proceso aplica para todos los datos.

Por último, procedemos a eliminar todo dato nulo existente dentro de cada una de las listas mediante un bucle que recorra toda la lista.

```
filtered_list = [value for value in lista if value != ""]
```

En este caso se recorrerá cada valor de la lista data y si existe un dato nulo se lo elimina además se utilizará SpaCy para obtener las palabras claves del texto con el fin de simplificar las palabras para que el entrenamiento del algoritmo sea eficiente, como paso final es guardar cada dataset creado con su respectiva clase tiene IVA, No Tiene IVA en sus respectivos directorios.



2.4.1.4. Modelado

A. Técnicas utilizadas para entrenar el modelo

Al ser un proceso de NLP será necesario el uso de modelados mediante TF-IDF ya que este al utilizar similitud de cosenos ya que tiene en cuenta como se escribe las palabras, además de una puntuación dentro del texto lo cual permite determinar a qué clase pertenece. Además, que al ser un clasificador de texto necesitamos el uso de algoritmos como Naive Bayes ya que estos tienen mejores resultados a la hora de clasificar texto mediante TF-IDF.

B. División de datos

Se particiona los datos en dos subconjuntos pertenecientes a Training y Test para el respectivo entrenamiento en este caso se usará Cross Validation con estado aleatorio cualquiera y las siguientes proporciones.

TRAINING	TEST
70%	30%

C. Desarrollo de los modelos

Luego de realizar el Split de datos será necesario tener que obtener el TF IDF para los textos ya separados en training y test, utilizaremos la librería que ofrece Sklearn para obtener el tf-idf, una vez transformado se procede con el entrenamiento del modelo.

2.4.1.5. Evaluar los modelos

A. Rendimiento de los algoritmos

Una vez entrenado el modelo realizaremos la predicción de la prueba para obtener los puntajes de las medidas de rendimiento para determinar qué tan efectivo es el algoritmo en este caso usaremos las medidas de accuracy, precision, recall. Luego de realizar las pruebas se obtiene los siguientes resultados para cada medida.

Accuracy	0.97
Precision	0.98
Recall	0.95

B. Interpretación de los resultados

Observamos que el modelo obtuvo muy buenos resultados en la evaluación de la prueba por ende es bastante preciso, sin embargo, el mayor desafío que enfrenta este tipo de modelos son la posibilidad de falta de vocabulario a la hora de nuevas instancias, en este caso específico se utilizó lo más usado y esencial que una microempresa manejaría a la hora de registrar un inventario

2.4.1.6. Despliegue

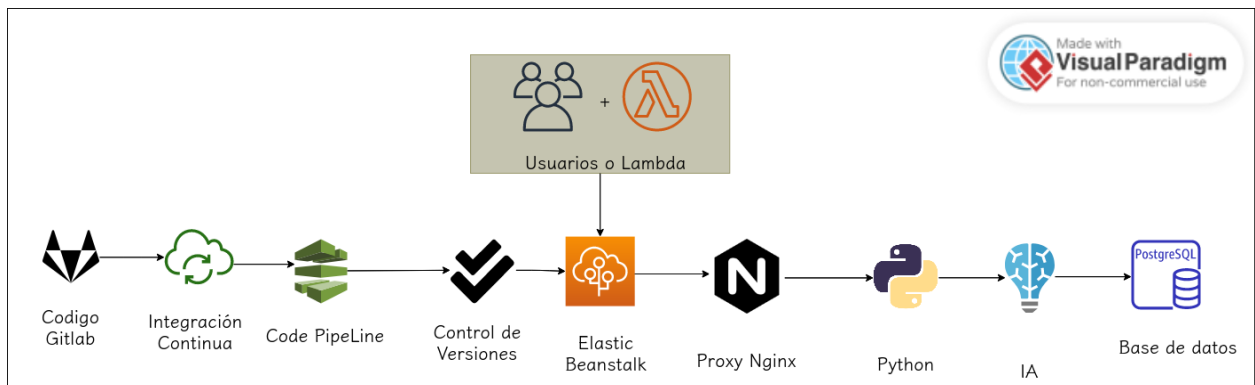
A. Implementación del modelo

Para iniciar la implementación del modelo se utilizó varios servicios de Amazon Web Services entre ellos Elastic Beanstalk y CodePipeline.

B. Proceso de despliegue

1. Creamos el ambiente de ejecución en este caso necesitaremos Python ya que el modelo se entrenó y se guardó en dicho lenguaje de programación.
2. Escogemos las opciones gratuitas para prueba y uso del prototipo
3. Creamos el Pipeline dentro de AWS conectando el repositorio de GitLab
4. Por último, integramos el entorno previamente creado y esperamos a que el proceso de integración finalice

Figura 22: Despliegue del modelo de IA



Nota. Despliegue del modelo entrenado en un servicio de AWS para su uso. Elaborado por:

Los autores.

Podemos observar en el diagrama que mediante estas herramientas el proceso se vuelve autónomo y que mediante el control de versiones con CodePipeline se puede desplegar de forma automatizada las diferentes versiones que se tenga del modelo facilitando la implementación continua.

C. Uso del modelo

Dentro de los usos que tiene es más predecir la clase a la que pertenece cierto producto dado su nombre, posee IVA, NO IVA de esta forma podemos crear un historial y registro de datos para poder reentrenar un modelo dependiendo el usuario y así el proceso sea automático, pero al ser un prototipo solo guarda el historial para un futuro posible reentrenamiento.

En este caso específico, la mayoría de los productos los clasifica bien, pero hay ciertos productos que no y es por la falta de vocabulario, en este caso al hacer el reentrenamiento es necesario corregir mediante la base de datos o el propio sistema cada uno de los productos.

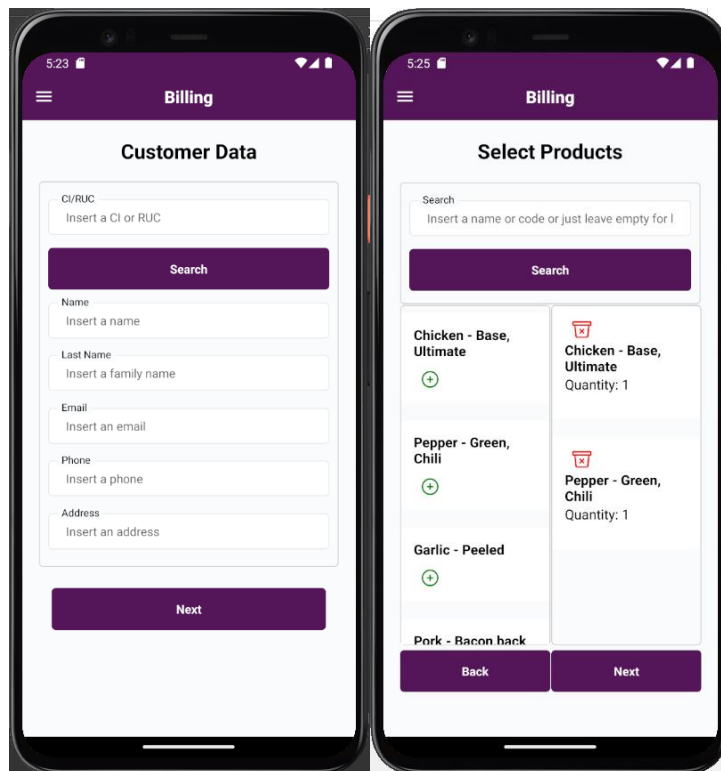
CAPÍTULO III

RESULTADOS

Este apartado muestra enteramente el resultado final del software. Posterior a su desarrollo podemos tener imágenes claras sobre su funcionamiento, así como observar el cumplimiento de los requisitos planteados al inicio y sobre los cuales se basa este proyecto. Al ser solamente un prototipo se decidió mostrar los resultados de la versión móvil ya que la versión web es similar salvo unos ligeros detalles en cuanto a su visualización.

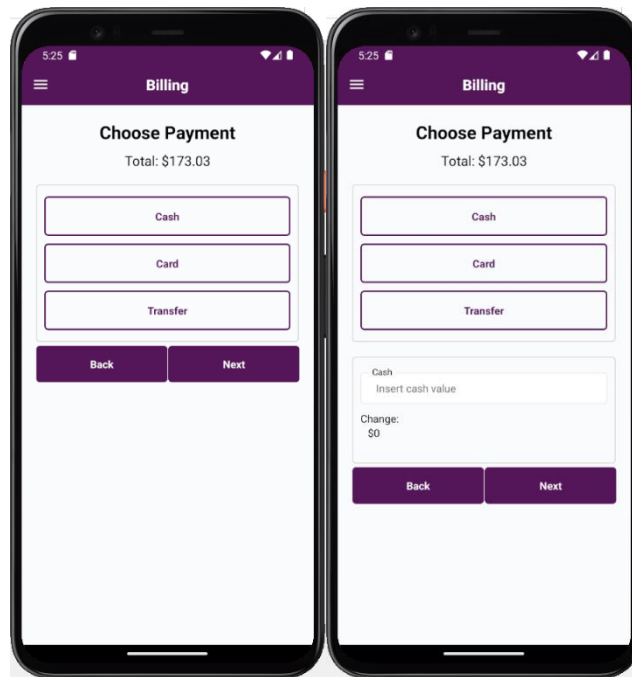
3.1. Formulario para crear factura – vista desde la aplicación móvil

Figura 23: Apartados “Datos del cliente” y “Selección de productos”



Nota. Interfaces resultantes sobre la generación de la factura, comprende el ingreso de datos del cliente y la selección de productos. Elaborado por: Los autores.

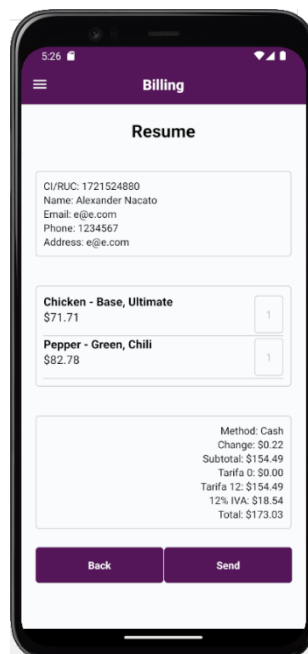
Figura 24: Apartado “Métodos de pago” y “Método en efectivo”



Nota. Interfaces resultantes de métodos de pago siendo seleccionado el pago en efectivo.

Elaborado por: Los autores.

Figura 25: Apartado “Resumen de factura”



Nota. Interfaz resultante del proceso de facturación donde se muestra un resumen final de los datos que van a ir en la factura. Elaborado por: Los autores.

Una vez creada la factura electrónica, se genera el archivo XML que debería ser enviado para su autorización, posterior eso debería ser enviado al cliente por correo electrónico. Debido a las limitantes previamente planteadas en el documento, el envío y autorización al SRI no se realizan.

3.2. Funciones desplegadas en Amazon Lambda

Al emplear una arquitectura de microservicios, es imprescindible llevar a cabo la implementación de funcionalidades en el sistema de forma individualizada, lo que implica la creación de funciones específicas para cada proceso ejecutado en el backend. Dichas funcionalidades, alojadas en entornos Lambda, se centran exclusivamente en los procedimientos destinados al análisis y gestión de las solicitudes generadas hacia las API existentes.

Figura 26: Funciones Lambda desplegadas

Nombre de la función	Descripción	Tipo de paquete	Tiempo de ejecución	Última modificación
newCategory	-	Zip	Node.js 20.x	el mes pasado
getClient	-	Zip	Node.js 20.x	el mes pasado
getTypes	-	Zip	Node.js 20.x	el mes pasado
amplify-login-custom-message-56182a2b	-	Zip	Node.js 16.x	el mes pasado
createSchemas	-	Zip	Node.js 20.x	el mes pasado
amplify-login-define-auth-challenge-56182a2b	-	Zip	Node.js 16.x	el mes pasado
GetTableEmpresa	-	Zip	Node.js 20.x	el mes pasado
amplify-login-verify-auth-challenge-56182a2b	-	Zip	Node.js 16.x	el mes pasado
newClient	-	Zip	Node.js 20.x	el mes pasado
newSupplier	-	Zip	Node.js 20.x	el mes pasado
validar_usuarios	-	Zip	Node.js 16.x	hace 2 meses
xml	-	Zip	Node.js 20.x	el mes pasado
amplify-login-creates-auth-challenge-56182a2b	-	Zip	Node.js 16.x	el mes pasado
newAccounting	-	Zip	Node.js 20.x	el mes pasado
newProduct	-	Zip	Node.js 20.x	el mes pasado
getSuppliers	-	Zip	Node.js 20.x	el mes pasado
getProducts	-	Zip	Node.js 20.x	el mes pasado
getCategories	-	Zip	Node.js 20.x	el mes pasado
getClientById	-	Zip	Node.js 20.x	el mes pasado
getQuantityByCategory	-	Zip	Node.js 20.x	el mes pasado
crear_usuarios	-	Zip	Node.js 16.x	hace 2 meses

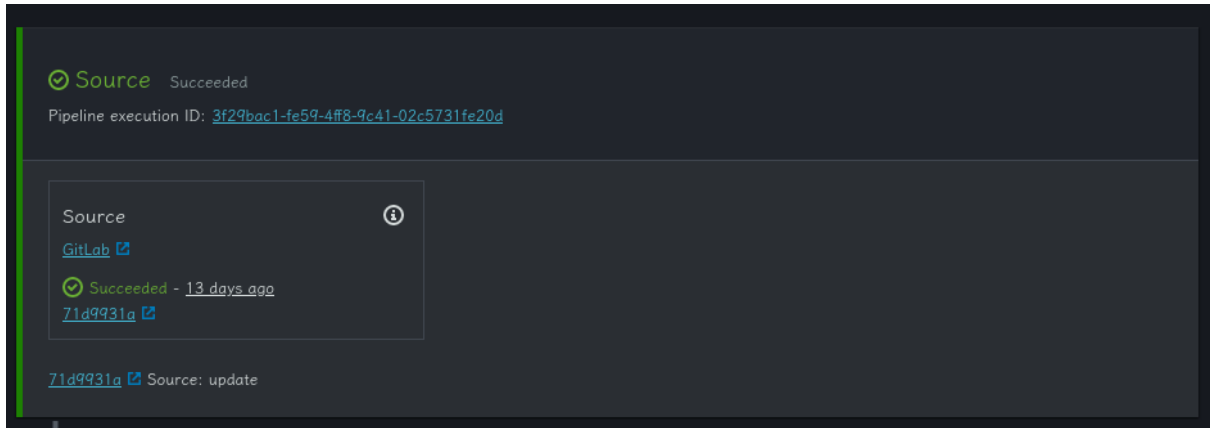
Nota. Captura de las funciones desplegadas en AWS Lambda. Elaborado por: Los autores.

3.3. Modelo de inteligencia artificial desplegado en Elastic Beanstalk

Al implementar un sistema de control de versiones, como GitLab, en conjunto con un servicio de entrega continua/despliegue continuo (CI/CD), como CodePipeline, se establece un flujo automatizado. En este escenario, el modelo, junto con sus bibliotecas funcionales, se almacena en un repositorio GitLab. Desde este repositorio, el flujo de trabajo se dirige a CodePipeline, donde se visualiza la implementación del modelo en la nube.

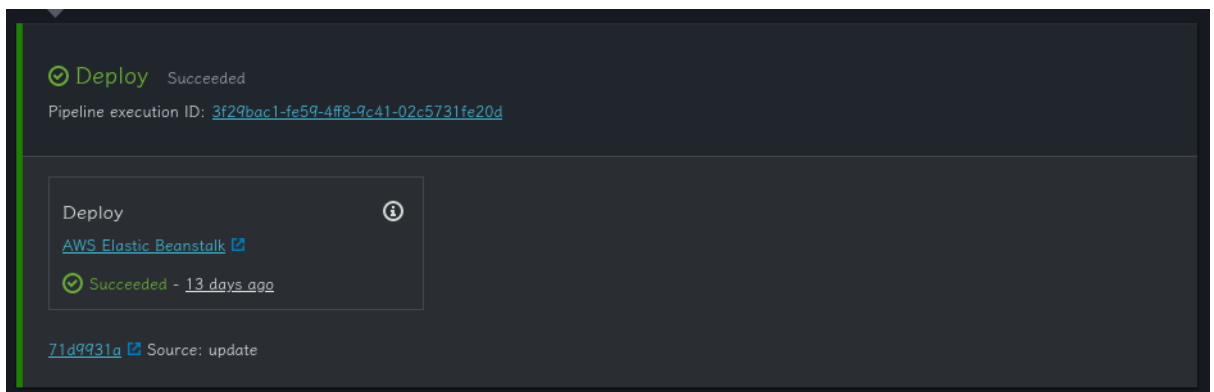
La fase final del proceso consiste en utilizar Elastic Beanstalk para desplegar y operar el modelo como API exclusiva para la inteligencia artificial. De esta manera, Elastic Beanstalk facilita la exposición y utilización del modelo de inteligencia artificial de manera eficiente.

Figura 27: Carga desde GitLab a CodePipeline



Nota. Captura de la carga del modelo alojado en GitLab a CodePipeline. Elaborado por: Los autores.

Figura 28: Despliegue del modelo en Elastic Beanstalk



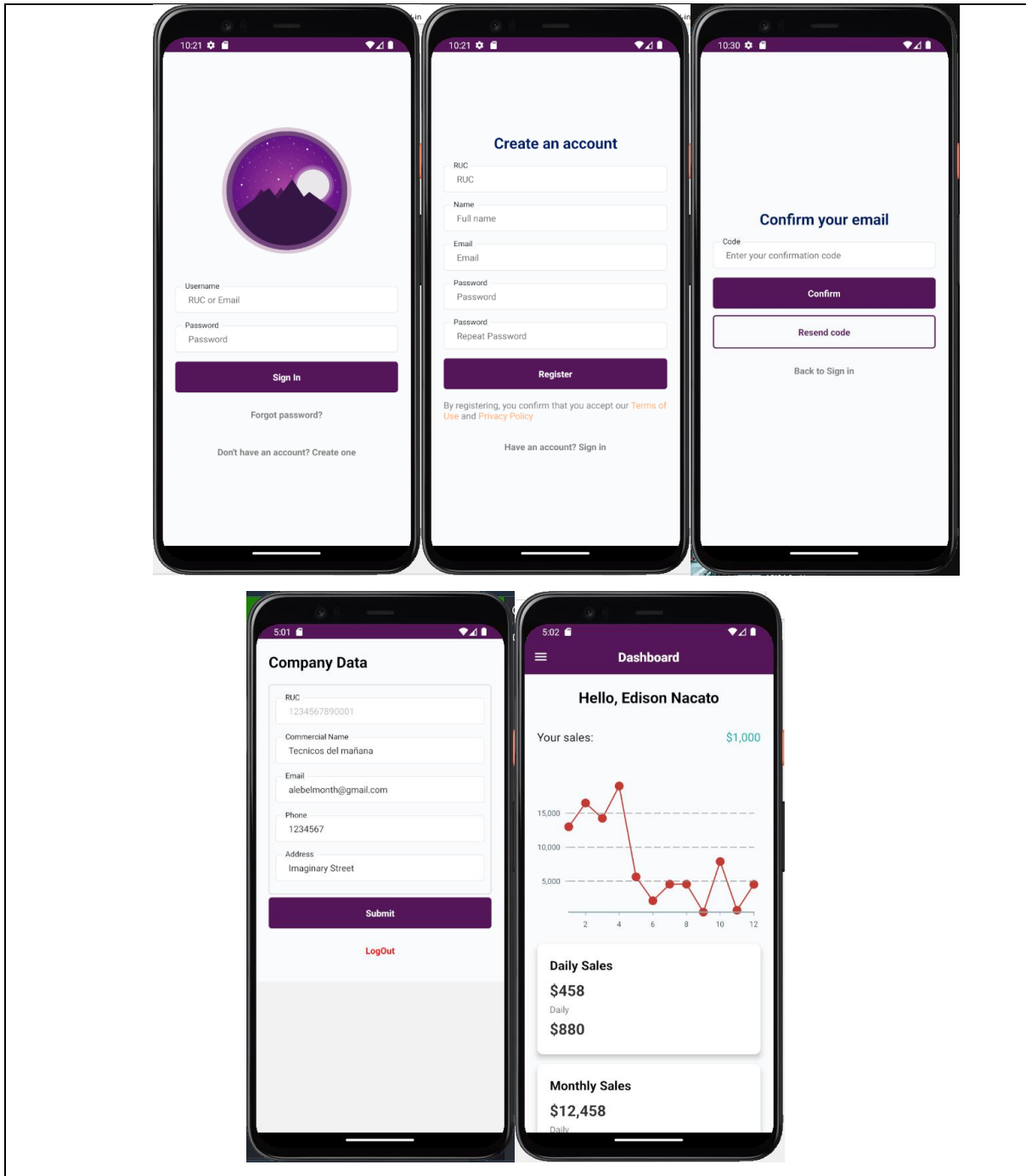
Nota. Captura del despliegue del modelo alojado en CodePipeline a su uso como API en Elastic Beanstalk. Elaborado por: Los autores.

3.4. Pruebas

Verificación de que los requerimientos han sido cumplidos y tienen un correcto funcionamiento.

PRUEBA DE REQUERIMIENTO FUNCIONAL 1	
Realizado por:	Edison Ñacato
Fecha:	02/01/2024

Descripción del requisito:	La aplicación permite el registro de nuevos usuarios. Los campos para proporcionar son: RUC, correo, nombre, apellido y contraseña.
Estado:	Aprobado
FLUJO ESPERADO	
<ol style="list-style-type: none"> 1. Ingresar al sistema desde web o aplicación móvil 2. Ingresar los datos del usuario 3. Confirmar el usuario 4. Registrar La empresa 5. Ingresar al sistema con el usuario respectivo 	
RESULTADO REAL	



PRUEBA DE REQUERIMIENTO FUNCIONAL 2	
Realizado por:	Edison Ñacato
Fecha:	02/01/2024
Descripción del requisito:	La aplicación permite el registro de nuevos clientes. Los campos para proporcionar son: CI o RUC, nombre, correo, dirección y teléfono.

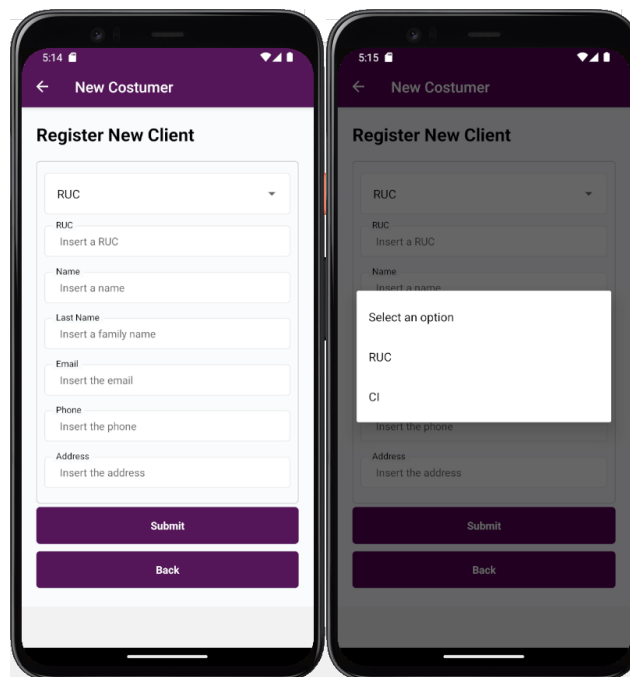
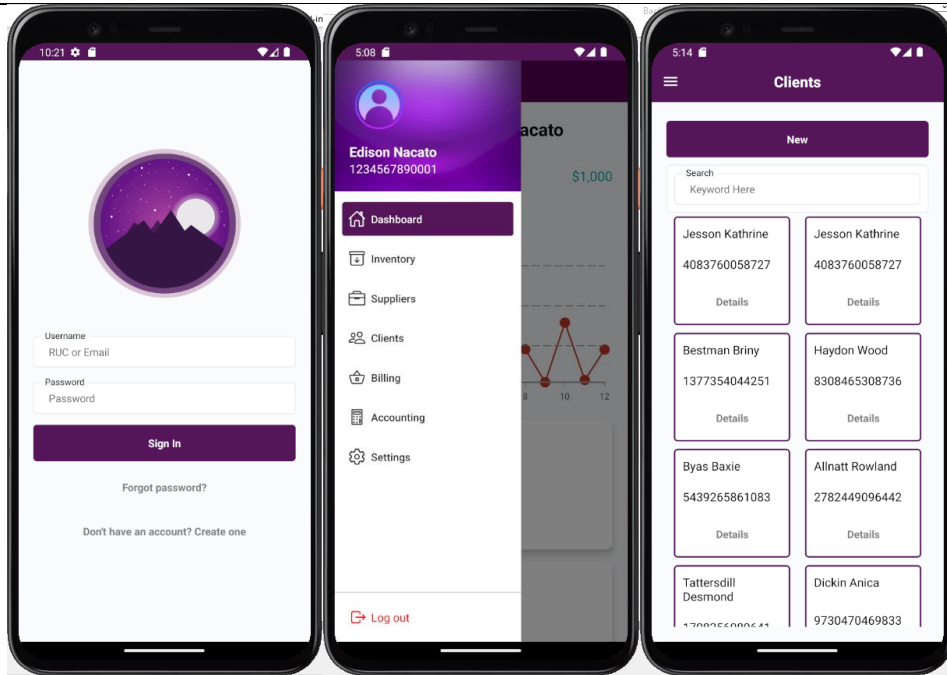
Estado:

Aprobado

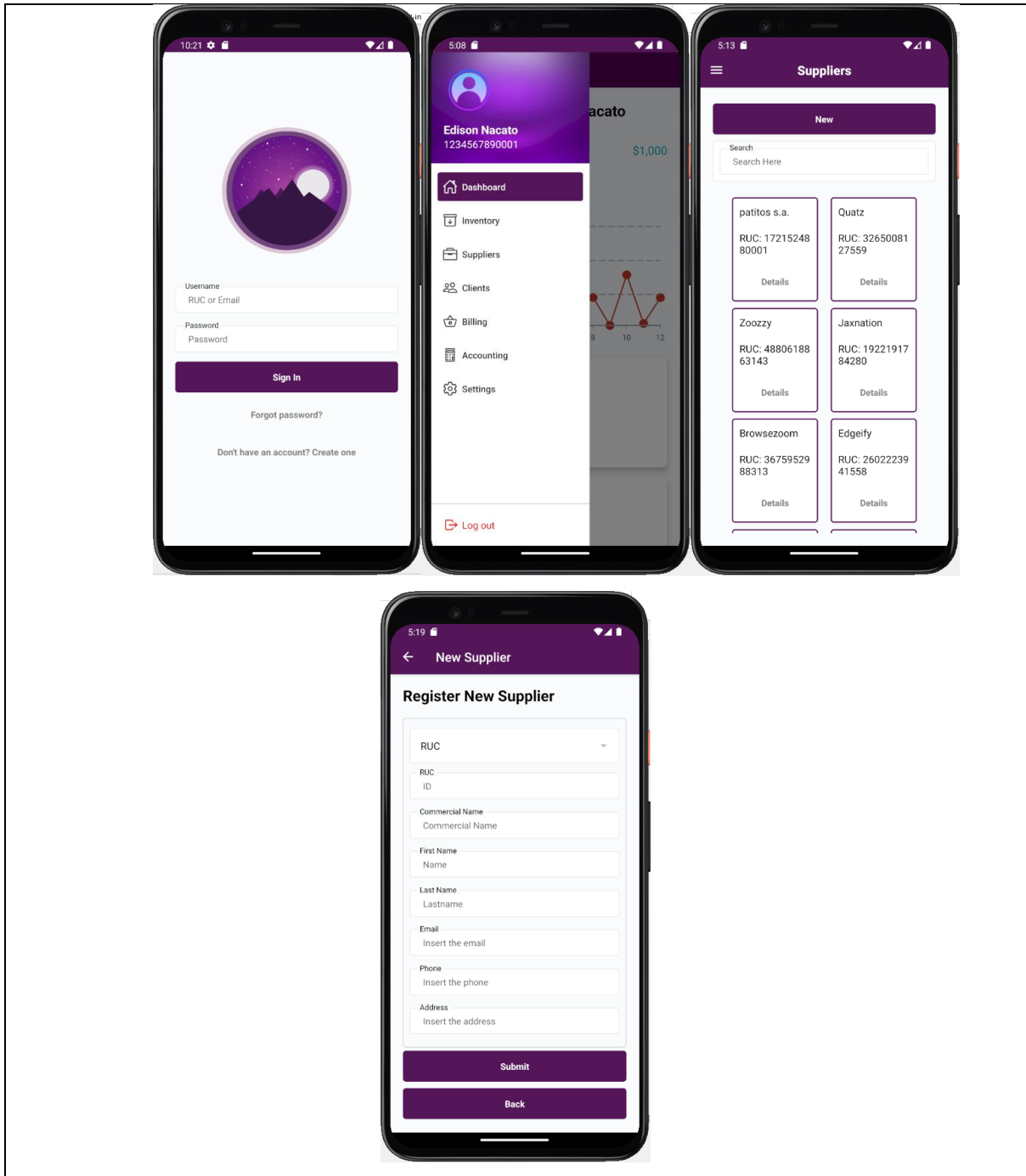
FLUJO ESPERADO

1. Ingresar al sitio de registro de clientes
2. Llenar los campos de CI/RUC, Nombre, Correo, Dirección y Teléfono
3. Registrar el cliente
4. Poder visualizar el cliente registrado

RESULTADO REAL

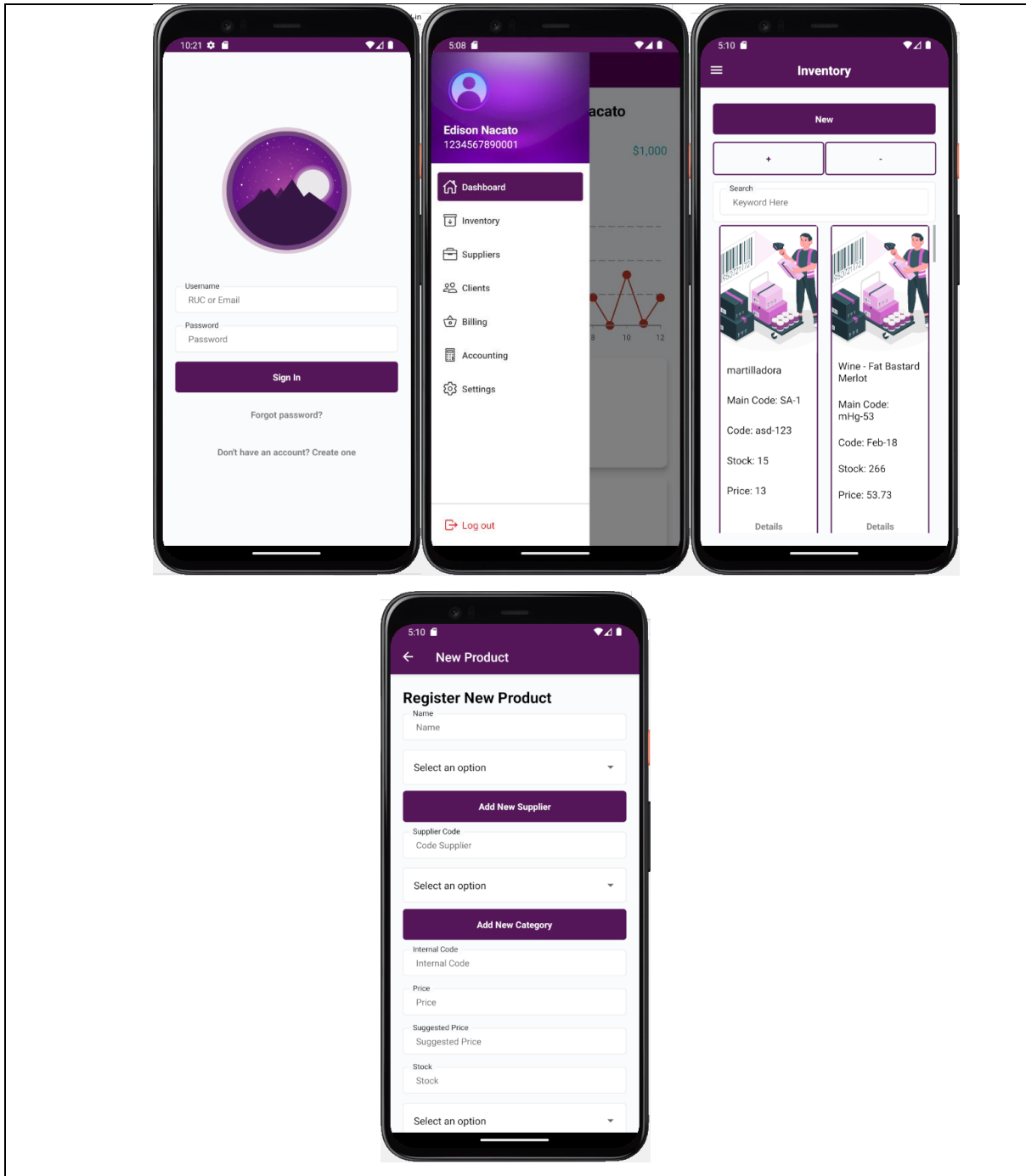


PRUEBA DE REQUERIMIENTO FUNCIONAL 3	
Realizado por:	Edison Ñacato
Fecha:	02/01/2024
Descripción del requisito:	La aplicación permite el registro de nuevos proveedores. Los campos para proporcionar son: RUC, nombre, nombre comercial, correo, dirección y teléfono.
Estado:	Aprobado
FLUJO ESPERADO	
<ol style="list-style-type: none"> 1. Ingresar al sitio de registro de proveedores 2. Llenar los campos de CI/RUC, Nombre, Correo, Dirección y Teléfono, Nombre Comercial 3. Registrar el proveedor 4. Poder visualizar el proveedor registrado 	
RESULTADO REAL	



PRUEBA DE REQUERIMIENTO FUNCIONAL 4	
Realizado por:	Edison Ñacato
Fecha:	02/01/2024
Descripción del requisito:	La aplicación permite el registro de nuevos productos. Los campos para proporcionar son: Nombre producto, proveedor, código del proveedor, categoría,

	código interno, precio, precio sugerido, stock, unidad y tipo.
Estado:	Aprobado
FLUJO ESPERADO	
<ol style="list-style-type: none"> 1. Ingresamos en la página de ingreso de productos 2. Se visualiza los productos si es que existen dentro de la base de datos 3. Ingresamos un nuevo producto 4. Llenamos los datos relaciones al producto 5. Ubicamos categorías y en el caso de no existir se crea una nueva mediante el formulario para crear categoría 6. Enviamos los datos 7. Visualizamos el producto si fue registrado 	
RESULTADO REAL	



PRUEBA DE REQUERIMIENTO FUNCIONAL 6	
Realizado por:	Edison Nacato
Fecha:	02/01/2024
Descripción del requisito:	La aplicación permite el registro de nuevas cuentas. Los campos para proporcionar son: Nombre de cuenta y padre de la cuenta.

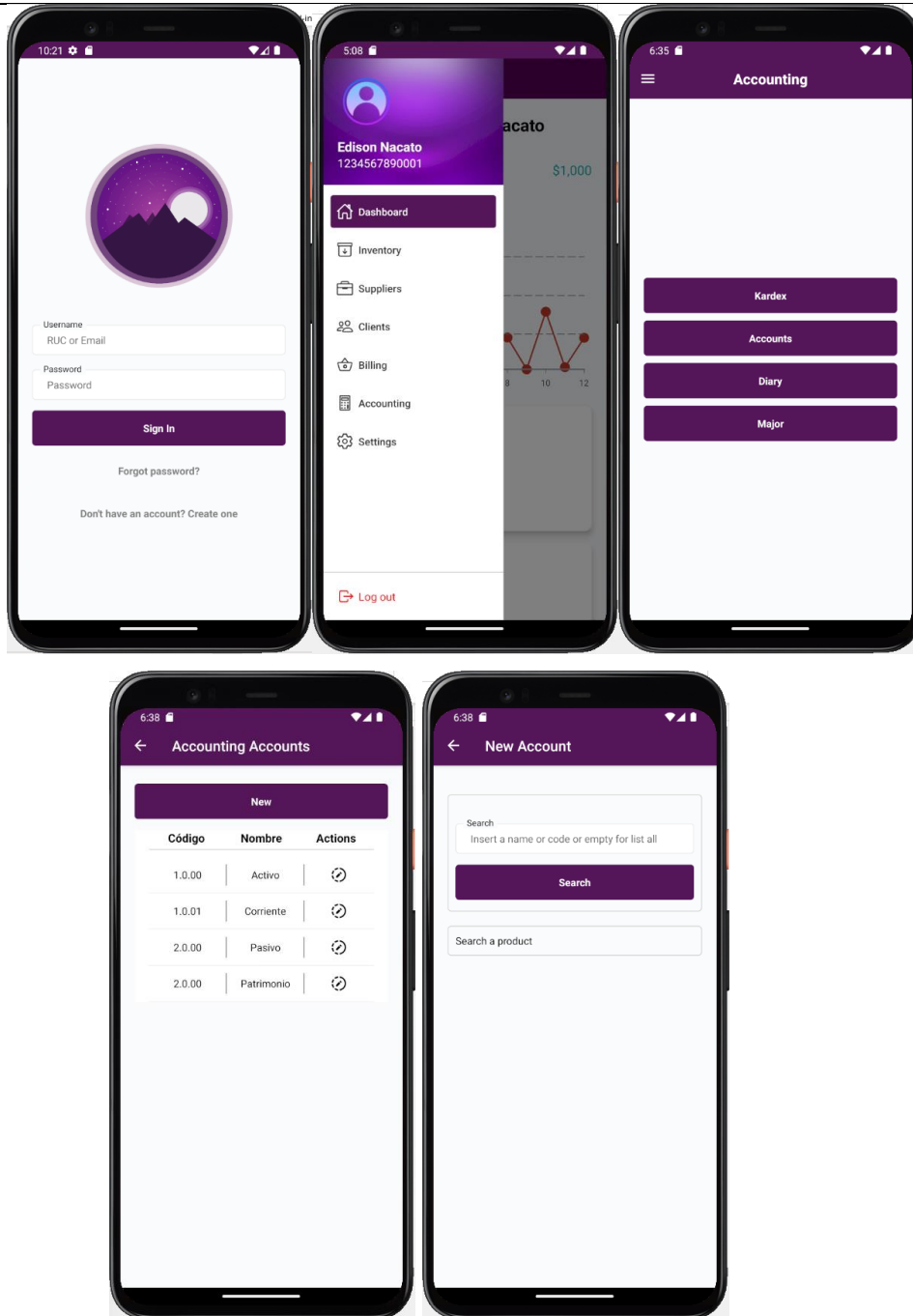
Estado:

Aprobado

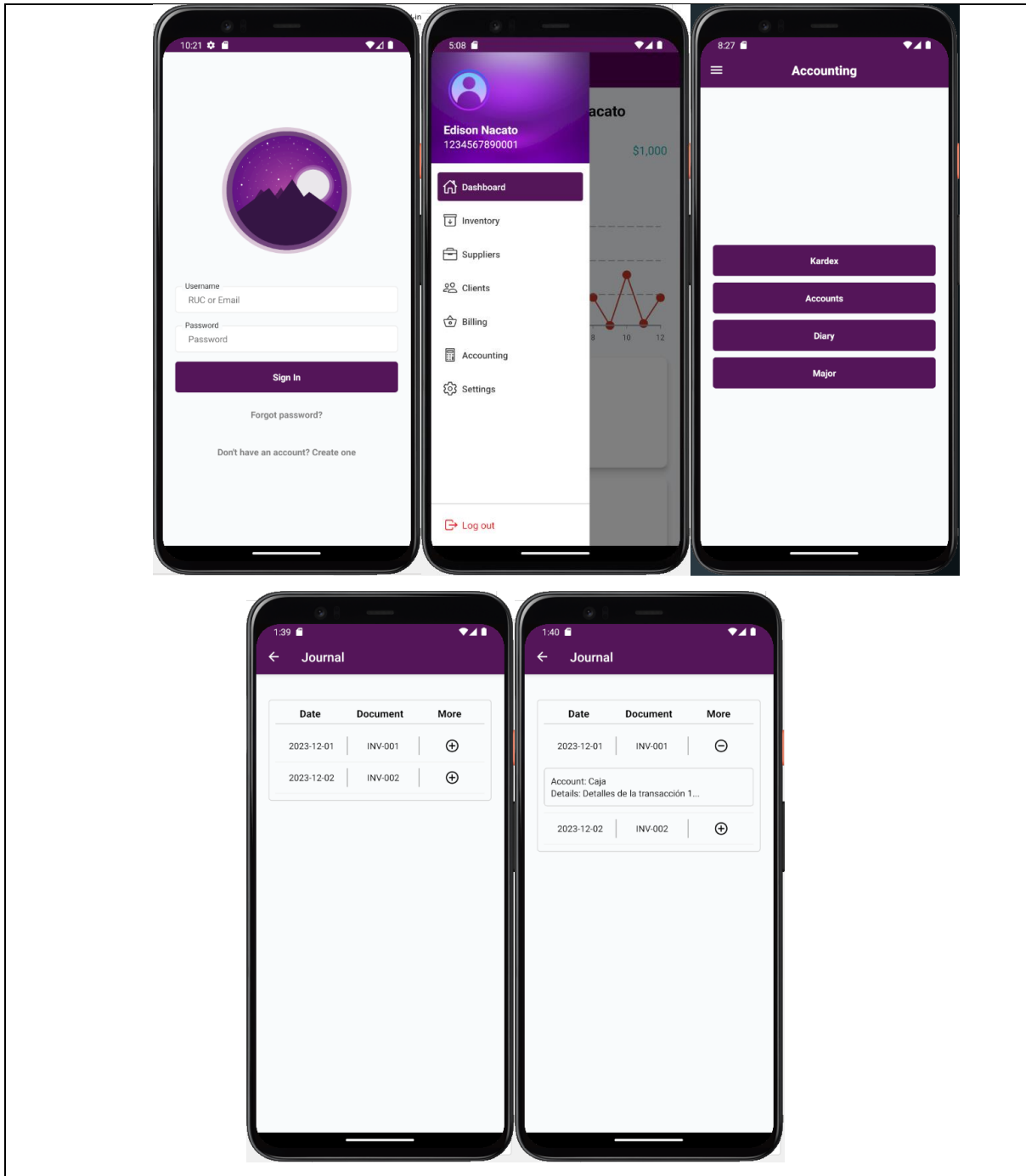
FLUJO ESPERADO

1. Ingresamos a la página de cuentas en el apartado de contabilidad
2. Ingresamos las cuentas que queremos registrar
3. Enviamos las cuentas
4. Visualizamos las cuentas en el sistema

RESULTADO REAL



PRUEBA DE REQUERIMIENTO FUNCIONAL 7	
Realizado por:	Edison Ñacato
Fecha:	02/01/2024
Descripción del requisito:	La aplicación permite el registro de nuevos asientos contables en el libro diario. Los campos para proporcionar son: fecha, numero de documento, cuenta, debe, haber y total.
Estado:	Aprobado
FLUJO ESPERADO	
<ol style="list-style-type: none"> 1. Posterior a la generación de factura, se registra el asiento contable. 2. Se registra el asiento contable en el libro diario. 3. Se ingresa al apartado de contabilidad 4. Se ingresa al apartado diario 5. Visualizar los asientos contables y si se requiere más información se da en el icono de más. 	
RESULTADO REAL	

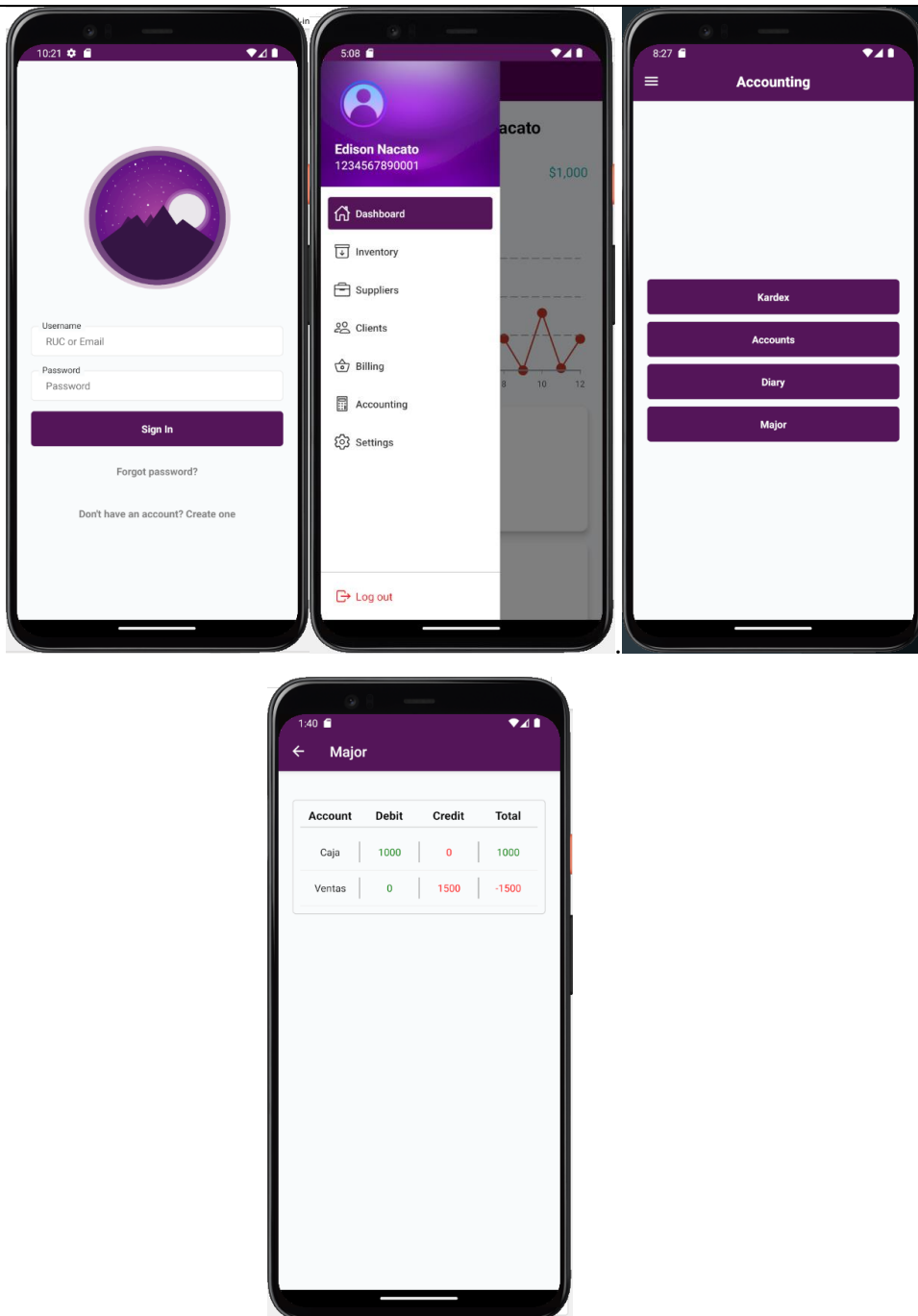


PRUEBA DE REQUERIMIENTO FUNCIONAL 8	
Realizado por:	Edison Ñacato
Fecha:	02/01/2024
Descripción del requisito:	La aplicación permite el registro de saldos en el libro mayor. Los campos para proporcionar son: Cuenta y saldos.
Estado:	Aprobado

FLUJO ESPERADO

1. Posterior a realizar la factura, automáticamente se registra el cambio en el libro diario seleccionando la cuenta contable correspondiente.
2. Posterior a registrar el asiento contable en el libro diario se registra los valores del movimiento en el balance general perteneciente al libro mayor.
3. Tener registrado en la base de datos
4. Visualizar el registro del libro mayor

RESULTADO REAL



CONCLUSIONES

A lo largo de este análisis, se ha evidenciado que las aplicaciones más sencillas poseen un potencial significativo en términos de demanda e impacto. A pesar de la amplia oferta de soluciones existentes en el mercado, la mayoría se orienta hacia competidores de mediana y gran envergadura, dejando desatendida la fuerza vital fundamental que impulsa la economía nacional.

Es crucial destacar que, aunque entidades como el SRI proporcionan soluciones tecnológicas gratuitas, estas herramientas suelen enfrentar dificultades operativas o presentar funciones de complejidad considerable. Este panorama subraya la imperante necesidad de explorar alternativas que satisfagan los requisitos mínimos y, al mismo tiempo, sean accesibles para su fácil utilización.

La IA ha experimentado un notable auge y crecimiento exponencial en los últimos meses, demostrando su utilidad en ámbitos centrados en el manejo de datos. Su aplicación ya no se limita exclusivamente a proyectos de envergadura y profesionales; actualmente, la inteligencia artificial se ha popularizado, generando una comprensión generalizada de sus capacidades y limitaciones. En mi perspectiva, su propósito fundamental radica en respaldar el progreso humano y fomentar la evolución continua.

No obstante, persisten importantes limitaciones en la tecnología actual, especialmente en ámbitos como el desarrollo de software, la ciencia de datos y el modelado de inteligencia artificial. La escasez de poder de cálculo y adaptaciones generalizadas para algunos de los lenguajes más utilizados constituye un desafío significativo. Aunque el desarrollo multiplataforma puede lograrse mediante diversos lenguajes y la adopción de determinados frameworks, es innegable que, en el caso específico de React Native, aún existen áreas de mejora para optimizar su desempeño y garantizar un funcionamiento óptimo en diversos dispositivos.

RECOMENDACIONES

El proyecto al ser un prototipo puede presentar mejoras a futuro como recomendaciones finales, de las cuales se puede destacar las siguientes:

Es totalmente necesario considerar la optimización de procesos durante el desarrollo del prototipo. En este sentido, se recomienda iniciar el desarrollo usando los servicios de AWS en vez de localmente. Esta estrategia permitirá monitorizar de manera efectiva el rendimiento y el uso de la aplicación. Al emplear servicios como AWS Lambda, se simplifica la creación

de funciones para usos específicos, lo que no solo mejora la eficiencia del backend, sino que también ahorra tiempo en el proceso de desarrollo.

En el desarrollo de una IA, es esencial priorizar la obtención de información relevante para lograr resultados más sólidos. Aplicar procedimientos avanzados de procesamiento del lenguaje contribuye significativamente a mejorar la precisión general del modelo. Resulta óptimo construir el modelo basándose en información específica del país o relacionada con las clases a clasificar. Además, se aconseja limitar la cantidad de datos y realizar un balance adecuado, ya que un exceso de datos en una clase puede sesgar el modelo. La implementación de técnicas de balanceo de clases y datos se vuelve crucial para garantizar la equidad y eficacia del modelo.

En el proceso de desarrollo, es esencial considerar la práctica de integración continua, ya que facilita la implementación regular de cambios, manteniendo un control efectivo de versiones y permitiendo la realización de pruebas de sistema con mayor flexibilidad. Al aplicar algoritmos de inteligencia artificial, esta metodología posibilita la actualización fluida del modelo o de las diversas funciones asociadas, sin afectar la versión anterior. Se recomienda el uso del servicio AWS Pipeline, el cual de manera autónoma puede llevar a cabo este proceso, permitiendo la realización de pruebas y mejoras de manera eficiente.

Es fundamental mantener entornos de desarrollo separados para las aplicaciones móviles y web. Esto se debe a que no todas las librerías, plugins y frameworks son completamente compatibles como soluciones multiplataforma. La falta de total disponibilidad puede ocasionar diversos problemas y errores. Por lo tanto, se recomienda optar por frameworks específicos y altamente eficientes para cada plataforma, garantizando así un rendimiento óptimo, al establecer un framework adecuado para el desarrollo en equipos de escritorio y otro para dispositivos móviles, se asegura que todas las funcionalidades y módulos operen de manera consistente. Esta práctica facilita la modificación y actualización de las aplicaciones, ya que cada framework cuenta con su propia documentación específica. De esta manera, se promueve un desarrollo más eficiente y libre de complicaciones, permitiendo a los equipos abordar proyectos de manera más efectiva.

REFERENCIAS

- Arellano Cepeda, O., Quispe Fernández, G., Ayaviri Nina, D., & Escobar Mamani, F. (2017). Estudio de la Aplicación del Método de Costos ABC en las Mypes del Ecuador. *Revista de Investigaciones Altoandinas - Journal of High Andean Research*, 19(1). <https://doi.org/10.18271/ria.2017.253>
- Reglamento de inversiones del código orgánico de la producción, Pub. L. No. Decreto Ejecutivo 757, Asamblea Nacional del Ecuador (2019).
- AWS. (2023a). *AWS Documentation*. AWS Documentation. <https://docs.aws.amazon.com/>
- AWS. (2023b). *Computación en la nube con AWS*. Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is-aws/>
- AWS. (2023c). *Desarrollo de pila completa - Aplicaciones web y móviles - AWS Amplify*. Amazon Web Services, Inc. <https://aws.amazon.com/es/amplify/>
- AWS. (2023d). *¿Qué es el Procesamiento de lenguaje natural (NLP)?* Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/nlp/>
- Becerra, M., & Ojeda, R. (2022). Beneficios de la facturación electrónica en las pequeñas y medianas empresas del Ecuador. *Visionario Digital*, 76–97. <https://cienciadigital.org/revistacienciadigital2/index.php/VisionarioDigital/article/view/2366>
- Cahyani, D. E., & Patasik, I. (2021). Performance comparison of tf-idf and word2vec models for emotion text classification. *Bulletin of Electrical Engineering and Informatics*, 10(5). <https://doi.org/10.11591/eei.v10i5.3157>
- Casinelli, P. (2011, November). *La Norma Internacional de Información Financiera para pequeñas y medianas entidades*. Superintendencia de Compañías, Valores y Seguros. https://www.supercias.gob.ec/bd_supercias/descargas/niif/EVENTO_INTERNACIONAL%20NIIF_%20GYE_15112011/Presentacion-SuperCias-NIIFPYME-V2.pdf
- Coello, D., Santander, K., Zambrano, W., & Cedeño, J. (2021). Innovación tecnológica y su impacto en el desarrollo de las microempresas por covid-19. *Ciencia Latina Revista Científica Multidisciplinar*, 5(4). https://doi.org/10.37811/cl_rcm.v5i4.641
- Crespo, K., Carchi, L., Zambrano, A., Orellana, A., & Gonzales, E. (2020). Mejora Continua en el proceso contable y su aporte en la competitividad de las MIPYMES en la Provincia de El Oro (Ecuador). *Espacios*, 41(2014).

- Encinas Cortés, J. (2020). *Arquitecturas serverless: qué son y a dónde nos llevan* [Grado, UAM.]. <https://repositorio.uam.es/handle/10486/692856>
- Espinosa Zúñiga, J. J. (2020). Aplicación de metodología CRISP-DM para segmentación geográfica de una base de datos pública. *Ingeniería Investigación y Tecnología*, 21(1). <https://doi.org/10.22201/fi.25940732e.2020.21n1.008>
- Hughey, D. (2017). The Traditional Waterfall Approach. In *University of Missouri*.
- Miller, C. (2018). Hands-On Data Analysis with NumPy and Pandas: Implement Python Packages from Data Manipulation to Processing. In *Hands-On Data Analysis with NumPy and Pandas*.
- Moncayo, C. (2020). La Inteligencia Artificial en la contabilidad. *Instituto Nacional de Contadores Públicos de Colombia*.
- Quisaguano Collaguazo, Mg. L. R., Pallasco Venegas, Mg. M. S., Andaluz Guerrero, A. A., Martínez Freire, M. N., & Corrales Beltrán, Mg. S. H. (2022). DESARROLLO HÍBRIDO CON FLUTTER. *Ciencia Latina Revista Científica Multidisciplinar*, 6(4). https://doi.org/10.37811/cl_rcm.v6i4.2959
- Ruan, S., Li, H., Li, C., & Song, K. (2020). Class-specific deep feature weighting for naïve bayes text classifiers. *IEEE Access*, 8. <https://doi.org/10.1109/ACCESS.2020.2968984>
- Scikit Learn. (2023). *sklearn.naive_bayes.MultinomialNB*. Scikit Learn Documentation. https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html
- Seraj, A., Mohammadi-Khanaposhtani, M., Daneshfar, R., Naseri, M., Esmaceli, M., Baghban, A., Habibzadeh, S., & Eslamian, S. (2022). Cross-validation. In *Handbook of HydroInformatics: Volume I: Classic Soft-Computing Techniques*. <https://doi.org/10.1016/B978-0-12-821285-1.00021-X>
- Solis Galindo, A. E., & Lezama León, E. (2019). Empleo de expresiones regulares como herramienta para la identificación y corrección ortográfica / gramatical en el idioma español. *Boletín Científico INVESTIGIUM de La Escuela Superior de Tizayuca*, 5(9). <https://doi.org/10.29057/est.v5i9.3953>
- SPACY Usage Documentation. (2023). *spaCy 101: Everything you need to know*. <https://Spacy.io/Usage/Spacy-101>.
- Tapia, E. (2023). Facturación electrónica es obligatoria desde noviembre de 2022 . *Primicias*. <https://www.primicias.ec/noticias/economia/facturacion-electronica-obligatoria-noviembre/>

Vera-Rivera, F. H., Gaona Cuevas, C. M., & Astudillo, H. (2019). Desarrollo de aplicaciones basadas en microservicios: tendencias y desafíos de investigación. *Revista Ibérica de Sistemas e Tecnologías de Informação*, E23.