

Desarrollo de una plataforma software para la gestión de recaudo por entidades en convenio

Nicolás Báez Mora

Trabajo de grado presentado como requisito para optar al título de Ingeniero de Sistemas

Director

Gabriel Rodrigo Pedraza Ferreira
PhD. en Ciencias de la Computación

Universidad Industrial de Santander
Facultad de Ingenierías Físico-Mecánicas
Escuela de Ingeniería de Sistemas e Informática

Ingeniería de Sistemas
Bucaramanga
2024

Agradecimientos

Quiero agradecer a Dios por su guía y fortaleza durante este proceso. También agradezco a mi familia por su apoyo constante. A mi director de trabajo de grado, Gabriel Rodrigo Pedraza Ferreira, le agradezco sus consejos. A todos los que contribuyeron de alguna manera, gracias. A mis amigos y seres queridos, gracias por estar ahí.

Además, quiero expresar mi gratitud a Seguros Bolívar por brindarme la oportunidad de desarrollar el software en el que trabajé durante este proyecto. Su colaboración y apoyo fueron fundamentales para el éxito de esta iniciativa. Estoy agradecido por la confianza que depositaron en mí y por permitirme aplicar mis habilidades en un entorno profesional.

También quiero dedicar un agradecimiento especial a todos los profesores que me acompañaron y guiaron a lo largo de mi formación académica durante mi carrera profesional. Su dedicación y enseñanzas fueron esenciales para mi crecimiento y desarrollo profesional. Estoy agradecido por su apoyo y por compartir su conocimiento conmigo.

Este proyecto ha sido un desafío emocionante y enriquecedor. A través de él, he adquirido nuevas habilidades y conocimientos que llevaré conmigo en mi futuro profesional. Estoy agradecido por esta oportunidad y emocionado por lo que el futuro pueda traer.

Tabla de contenido

| | Pág. |
|--|-------------|
| Introducción..... | 11 |
| 1. Generalidades..... | 11 |
| 1.1. Planteamiento del problema..... | 11 |
| 1.2. Justificación..... | 12 |
| 1.3. Objetivos..... | 12 |
| 1.3.1. Objetivo General..... | 12 |
| 1.3.2. Objetivos específicos..... | 12 |
| 1.4. Metodología..... | 13 |
| 1.4.1. Fase de Introducción Tecnológica..... | 14 |
| 1.4.2. Fase de Identificación de Requerimientos Clave..... | 14 |
| 1.4.3. Fase de Desarrollo de Procesos de Trabajo..... | 15 |
| 1.4.4 Fase de Implementación y Validación..... | 15 |
| 2. Marco de referencia..... | 16 |
| 2.1 Marco conceptual..... | 16 |
| 2.1.1. Términos relevantes..... | 16 |
| 2.1.2. Sistemas PGM..... | 16 |
| 2.1.3. Antecedentes de aplicaciones PGM..... | 17 |
| 2.1.3.1. PayJoy..... | 17 |
| 2.1.3.2. eCredable..... | 17 |
| 2.1.3.3. BillTrust..... | 17 |

| | |
|--|----|
| 2.2 Marco teórico..... | 18 |
| 2.2.1. Arquitectura software..... | 18 |
| 2.2.2 Aplicaciones web SPA..... | 18 |
| 2.2.3. Tecnologías de base de datos..... | 19 |
| 2.2.4. Tecnologías web | 20 |
| 3. Requerimientos del sistema..... | 20 |
| 3.1. Requerimientos no funcionales..... | 20 |
| 3.1.1. Usabilidad..... | 20 |
| 3.1.2. Seguridad..... | 21 |
| 3.2. Requerimientos funcionales..... | 21 |
| 3.2.1. Módulo de registro de pagos..... | 21 |
| 3.2.2. Módulo de consultas..... | 22 |
| 3.2.3. Módulo de administración..... | 22 |
| 4. Diseño del sistema..... | 23 |
| 4.1. Arquitectura software..... | 23 |
| 4.2. Diseño de la base de datos..... | 24 |
| 4.3. Diagramas de casos de uso..... | 25 |
| 4.4. Diseño UX..... | 28 |
| 4.5. Planteamiento de tareas..... | 30 |
| 5. Implementación..... | 31 |
| 5.1. Módulo de registro de pagos..... | 33 |
| 5.2. Módulo de consultas..... | 34 |

| | |
|--|----|
| 5.3. Módulo de administración..... | 37 |
| 6. Validación del sistema..... | 40 |
| 6.1. Plan de pruebas..... | 40 |
| 6.2. Resultado de pruebas unitarias..... | 41 |
| 6.3. Resultado de pruebas de interfaz..... | 43 |
| 7. Conclusiones..... | 54 |
| 7.1. Recomendaciones y trabajo futuro..... | 54 |
| Referencias bibliográficas..... | 56 |

Lista de Figuras

| | Pág. |
|--|-------------|
| Figura 1 Pizarra de Kanban..... | 14 |
| Figura 2 Arquitectura de software del proyecto..... | 24 |
| Figura 3 Diagrama entidad-relación de la base de datos..... | 25 |
| Figura 4 Diagrama de casos de uso para el procesamiento de archivos..... | 26 |
| Figura 5 Diagrama de casos de uso para la consulta de pagos procesados..... | 26 |
| Figura 6 Diagrama de casos de uso para la consulta de cartera..... | 27 |
| Figura 7 Diagrama de casos de uso para la consulta de comisiones..... | 27 |
| Figura 8 Diseño de la vista del módulo de consultas..... | 28 |
| Figura 9 Diseño de la vista del módulo de registro de pagos..... | 29 |
| Figura 10 Diseño de la vista del módulo del administrador..... | 29 |
| Figura 11 Diseño de la vista de la sección de gestión de comisiones..... | 30 |
| Figura 12 Tablero Jira..... | 31 |
| Figura 13 Vista general de la sección de registro de pagos..... | 33 |
| Figura 14 Vista del modal del recuento de pagos procesados..... | 34 |
| Figura 15 Vista del modal de consulta de pagos..... | 34 |
| Figura 16 Vista de los resultados de la consulta de pagos procesados..... | 35 |
| Figura 17 Vista del modal de consulta de balance de billetera del cliente..... | 35 |
| Figura 18 Vista de los resultados de la consulta de balance de billetera..... | 36 |
| Figura 19 Vista del modal de consulta de comisiones..... | 36 |
| Figura 20 Vista de los resultados de la consulta de comisiones..... | 37 |

| | |
|---|----|
| Figura 21 Vista del login de acceso al módulo de consultas..... | 38 |
| Figura 22 Vista del módulo de administración..... | 38 |
| Figura 23 Vista de la sección de administración de facturas..... | 39 |
| Figura 24 Vista del modal de nueva factura..... | 39 |
| Figura 25 Vista de la sección de gestión de comisiones manualmente..... | 40 |
| Figura 26 Recuento del coverage de pruebas unitarias..... | 42 |
| Figura 27 Componente de registro de pagos procesados..... | 44 |
| Figura 28 Modal de validación del componente de registro de pagos..... | 44 |
| Figura 29 Componente de consulta de pagos..... | 45 |
| Figura 30 Modal con los datos de pagos procesados..... | 45 |
| Figura 31 Componente de consulta de cartera..... | 46 |
| Figura 32 Componente de consulta de cartera..... | 46 |
| Figura 33 componente de consulta de comisiones..... | 47 |
| Figura 34 Modal con los datos de comisiones de entidades de convenio..... | 48 |
| Figura 35 Componente de acceso al módulo de administrador..... | 49 |
| Figura 36 Tabla en el módulo de administración con los datos de las facturas..... | 50 |
| Figura 37 Modal para agregar nuevas facturas..... | 51 |
| Figura 38 Componente de generación de comisión manualmente..... | 52 |
| Figura 39 Componente de generación de comisión manualmente..... | 53 |
| Figura 40 Modal de validación de fechas..... | 53 |

Glosario

Sistema PGM: Los sistemas de gestión de pagos son plataformas tecnológicas diseñadas para facilitar la administración y procesamiento de transacciones financieras electrónicas. Estos sistemas ofrecen herramientas y funcionalidades que permiten a los comerciantes gestionar la recepción y procesamiento de pagos de manera eficiente.

Software: Se refiere a un conjunto de programas y reglas informáticas que se utilizan para realizar diversas tareas en una computadora, siguiendo un conjunto específico de instrucciones.

Aplicación web: Se trata de un conjunto de programas y recursos informáticos diseñados para ser utilizados a través de un navegador web. Estas aplicaciones son accesibles desde cualquier dispositivo con conexión a internet y permiten a los usuarios interactuar con servicios, funcionalidades y contenidos alojados en servidores remotos, sin necesidad de instalar software adicional en sus dispositivos locales.

Aplicación SPA (Single Page Application): Se refiere a un conjunto de programas y recursos informáticos que constituyen una aplicación web que se carga en su totalidad en una única página web. Estas aplicaciones ofrecen una experiencia de usuario fluida al cargar dinámicamente contenido nuevo según las interacciones del usuario, sin necesidad de recargar la página completa.

Procesamiento: La manipulación y transformación de datos en diferentes formatos o estructuras para realizar cálculos, generar informes, o realizar acciones específicas según las necesidades del sistema.

Servicios web: El servidor web procesa las solicitudes entrantes de los clientes, que pueden ser peticiones de información, acciones específicas, o cualquier otro tipo de interacción.

Facturación: El proceso de emitir documentos legales que detallan transacciones comerciales, incluyendo bienes o servicios.

Comisión: Una tarifa o porcentaje acordado que se cobra por un servicio o transacción específica, generalmente en el contexto de ventas, servicios financieros o intermediación comercial.

Resumen

Título: Desarrollo de una plataforma software para la gestión de recaudo por entidades en convenio*

Autor: Nicolás Báez Mora**

Descripción: Hoy día en la actualidad las empresas van creciendo a un ritmo bastante rápido, esto debido principalmente al proceso de globalización que ha tomado el mundo al entrar en la expansión digital, de acuerdo a este crecimiento algunas empresas entran en la necesidad de diversificar los medios en los que operan, tanto en temas de procesos internos como en sus métodos de cobranza por los servicios prestados a sus usuarios. El software que se desarrolla se enfoca en la gestión integral de los recaudos por parte de las entidades de convenio de pago de facturas asociadas a la empresa principal.

Este sistema automatizado procesa los pagos recibidos, gestiona el pago de facturas pendientes y calcula las comisiones correspondientes a cada entidad de convenio. Al centralizar y agilizar estos procesos, el software permitirá una administración eficiente de los recursos financieros, garantizando un seguimiento preciso de cada transacción y facilitando la contabilidad de la empresa. Con un enfoque intuitivo y accesible, el software proporcionará a los administradores y empleados las herramientas necesarias para gestionar de manera efectiva el flujo de pagos y el cálculo de comisiones, optimizando así la operatividad del negocio.

El desarrollo de este software fue parte de una hackatón realizada por la empresa Seguros Bolívar, lo que implicó seguir un proceso estructurado que abarcó desde la planificación hasta la validación final del proyecto. Durante la fase de planificación, se identificaron las necesidades clave del cliente y se definieron los objetivos del proyecto, estableciendo un marco claro para el desarrollo. En la etapa de desarrollo, se trabajó de manera colaborativa en la construcción del sistema, asegurando que cada módulo cumpliera con los requisitos especificados. Posteriormente, se llevó a cabo una fase de validación rigurosa, donde se probaron todas las funcionalidades para garantizar su correcto funcionamiento y alineación con los objetivos planteados. Este enfoque metódico permitió la entrega de un software robusto y adaptado a las necesidades de las entidades de convenio.

* Trabajo de grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director Gabriel Rodrigo Pedraza Ferreira. Phd en Ciencias de la Computación

Abstract

Title: Development of a Software Platform for Collection Management by Partner Entities*

Author: Nicolás Báez Mora**

Description: Nowadays, companies are growing at a very fast pace, mainly due to the globalization process that has taken the world into the digital expansion era. With this growth, some companies find the need to diversify the ways they operate, both in terms of internal processes and their collection methods for services provided to their users. The software being developed focuses on the comprehensive management of collections by payment partner entities associated with the main company.

This automated system processes received payments, manages outstanding invoices, and calculates the corresponding commissions for each partner entity. By centralizing and streamlining these processes, the software will allow efficient management of financial resources, ensuring accurate tracking of each transaction and facilitating the company's accounting. With an intuitive and accessible approach, the software will provide administrators and employees with the necessary tools to effectively manage payment flows and commission calculations, thus optimizing business operations.

The development of this software was part of a hackathon held by Seguros Bolívar, which involved following a structured process from planning to final project validation. During the planning phase, the client's key needs were identified, and project objectives were defined, establishing a clear framework for development. In the development stage, collaborative work was carried out in building the system, ensuring that each module met the specified requirements. Subsequently, a rigorous validation phase was conducted, where all functionalities were tested to ensure their proper operation and alignment with the outlined goals. This methodical approach allowed the delivery of robust software tailored to the needs of the partner entities.

* Degree Work

** Faculty of Physical-Mechanical Engineering. School of Systems Engineering and Informatics. Director Gabriel Rodrigo Pedraza Ferreira. Phd in Computer Science

Introducción

En un mundo empresarial cada vez más dinámico y competitivo, la eficiencia en los procesos operativos es esencial para mantener una ventaja. Este proyecto de software se centra en abordar un desafío crítico que muchas empresas enfrentan diariamente: la recepción y procesamiento de pagos y facturas. Identificamos la necesidad de optimizar y automatizar estas tareas para agilizar los flujos de trabajo y reducir los errores manuales.

El objetivo principal de este proyecto es desarrollar un sistema que automatice y optimice estos procesos. Así, las empresas podrán eliminar tareas repetitivas, disminuir la probabilidad de errores y mejorar la gestión de los datos financieros.

Este software simplificará la recepción y procesamiento de pagos y facturas mediante la automatización de tareas redundantes. A través de la carga de archivos con datos de pago, el sistema llevará a cabo procesos y validaciones para detectar errores comunes como fechas incorrectas, facturas duplicadas y cálculos erróneos. Una vez procesados, el software registrará la información en el sistema, proporcionando una gestión integral y mayor visibilidad de los datos.

El proyecto se desarrollará en varias fases distintas, cada una de ellas detallará el proceso, incluyendo el análisis, diseño, implementación, seguridad básica y pruebas. Cada etapa estará claramente definida para asegurar una correcta comprensión y ejecución.

1. Generalidades

1.1. Planteamiento del problema

La gestión de pagos por servicios prestados representa un desafío recurrente para las empresas que ofrecen una amplia gama de servicios a una diversa base de clientes. Esta situación se intensifica con el aumento de la clientela y la complejidad de los procesos de facturación. Para solventar esta problemática, muchas empresas recurren a establecer acuerdos con entidades colaboradoras, delegando en ellas la tarea de recaudar los pagos de las facturas de los clientes. Estos acuerdos permiten una mayor flexibilidad y eficiencia en el proceso de recaudación, al tiempo que liberan recursos internos para otras actividades críticas del negocio.

La creciente complejidad del entorno empresarial y las demandas cambiantes de los clientes han generado la necesidad de optimizar los procesos de recaudación de pagos. Además, la digitalización y la globalización han impulsado la adopción de soluciones tecnológicas que permitan una gestión más eficiente y transparente de las transacciones financieras. En este contexto, el desarrollo de un sistema de software especializado se presenta como una solución clave para mejorar la eficiencia operativa y garantizar la satisfacción del cliente.

1.2. Justificación

El proyecto se centra en el desarrollo de un sistema de software especializado en la gestión integral de los pagos realizados a través de entidades colaboradoras. Este sistema tiene como objetivo principal automatizar y optimizar el proceso de recaudación de pagos, manteniendo un registro actualizado del estado de las facturas de los clientes y calculando de forma precisa las comisiones correspondientes para cada entidad colaboradora en función de su contribución. La necesidad de esta herramienta surge de la creciente complejidad y volumen de transacciones financieras, así como de la demanda de una mayor eficiencia en la gestión de recursos empresariales.

Además de abordar las necesidades específicas del negocio, el proyecto ofrece una valiosa oportunidad para adquirir y aplicar conocimientos en el campo del desarrollo de software. La implementación de este sistema implica el diseño y la integración de diversas tecnologías, incluyendo bases de datos, desarrollo de microservicios e interfaces gráficas de usuario. Esto me permitirá obtener experiencia práctica en áreas clave de la ingeniería de software, fortaleciendo mi competencia profesional y preparándome para futuros desafíos en el campo tecnológico.

Asimismo, la implementación de este sistema proporcionará a la empresa una ventaja competitiva al mejorar la eficiencia y la precisión en la gestión de pagos, lo que se traducirá en una mejor experiencia para los clientes y en un aumento de la confianza y la fidelidad hacia la empresa. Por último, la implementación de este sistema contribuirá a la estandarización y optimización de los procesos internos de la empresa, lo que facilitará la toma de decisiones informadas y la identificación de oportunidades de mejora continua. Esto permitirá a la empresa adaptarse de manera más ágil a los cambios del mercado y mantener su competitividad a largo plazo.

1.3. Objetivos

1.3.1. Objetivo general

Diseñar un sistema software de gestión, para dar soporte a las diferentes operaciones involucradas en el proceso de recaudación de pagos por parte de distintas entidades de convenio asociadas a una empresa.

1.3.2. Objetivos específicos

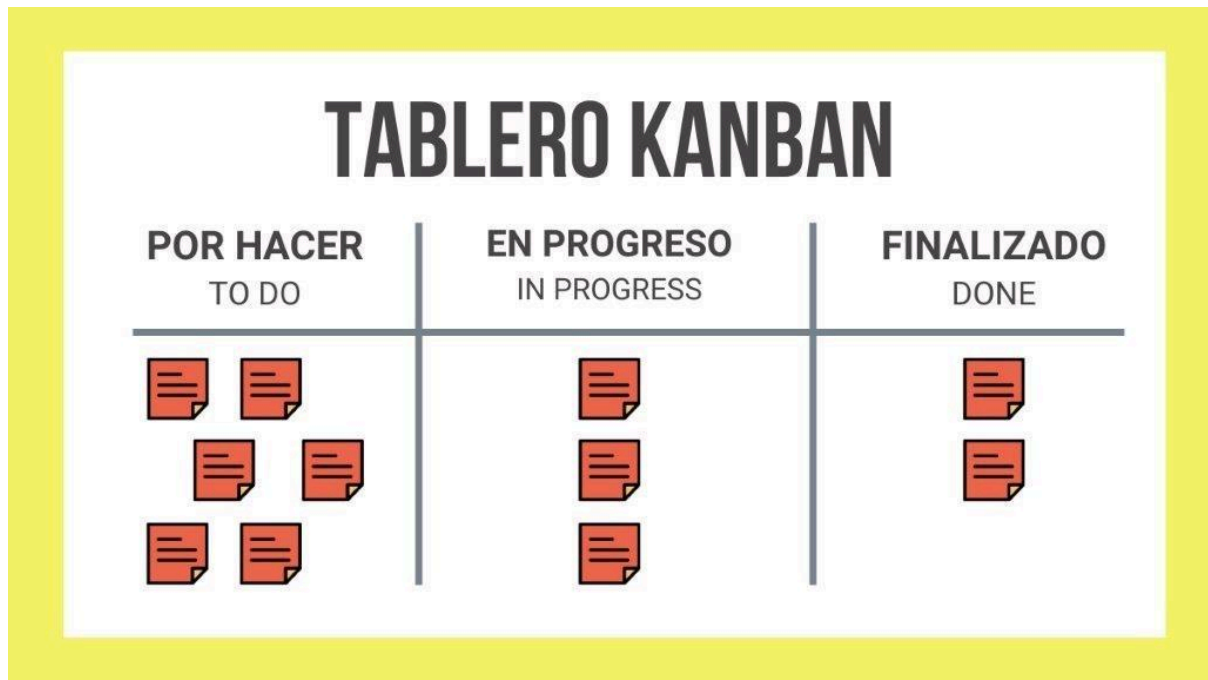
- Analizar los requerimientos del sistema, definidos en el alcance del proyecto, para establecer los criterios de aceptación de cada uno de ellos y garantizar su entendimiento completo.
- Diseñar la arquitectura y los componentes para la implementación del sistema, utilizando una abstracción adecuada y considerando las mejores prácticas y estándares actuales.

- Implementar las funcionalidades que satisfagan los requerimientos definidos, siguiendo el diseño propuesto y adaptándose a las necesidades específicas del proyecto.
- Realizar pruebas en distintos niveles del sistema, incluyendo pruebas unitarias y de la interfaz de usuario, para validar la solución desarrollada y garantizar su correcto funcionamiento.
- Evaluar el rendimiento del sistema una vez implementado, utilizando herramientas y técnicas adecuadas, para identificar posibles cuellos de botella y optimizar su funcionamiento.

1.4. Metodología

Para el desarrollo de este proyecto se plantea utilizar una metodología ya existente, Kanban, bastante utilizada en el entorno del desarrollo de software para la organización y definición de tareas.

Kanban es un enfoque de gestión visual que se utiliza para optimizar el flujo de trabajo en proyectos reales. Se basa en la idea de visualizar el trabajo en tarjetas (o "kanbans") que representan unidades de trabajo, como tareas, historias de usuario o características del producto. Estas tarjetas se mueven a través de un tablero Kanban que tiene columnas que representan diferentes etapas del proceso de trabajo, como "Por hacer", "En progreso" y "Terminado". Kanban es un enfoque flexible y adaptable para la gestión de proyectos reales, que se centra en la visualización del trabajo, la limitación del trabajo en curso, la gestión del flujo y la mejora continua. Esto permite a los equipos trabajar de manera más eficiente, colaborativa y centrada en el valor del cliente.

Figura 1*Pizarra de Kanban*

De acuerdo a esto la implementación del proyecto se realizará en 4 fases, mediante las cuales se seguirá la rúbrica planteada por el modelo Kanban, ya que el equipo de desarrollo consta de solamente una persona no se realizarán procesos cíclicos, sino más bien se llevará un enfoque más lineal en donde el avance del proyecto se irá definiendo por la culminación de las tareas planteadas en cada una de las fases.

1.4.1. Fase de Introducción Tecnológica

Esta primera fase consistirá en el análisis e investigación de las diferentes tecnologías disponibles para el desarrollo del proyecto, con el objetivo de definir las tecnologías más adecuadas para usar en el proyecto y a su vez adquirir los conocimientos necesarios para poder realizar el desarrollo del mismo, de acuerdo a esto se plantean las siguientes: Spring Boot con el lenguaje de programación Java para definir la lógica del servidor y los microservicios que expondrán la interoperabilidad con este, Angular para crear la interfaz de usuario y toda la parte visual del aplicativo y como opción para el almacenamiento de datos e información, Oracle.

1.4.2. Fase de Identificación de Requerimientos Clave

En esta segunda fase se realizará una identificación de los requerimientos clave de acuerdo a los elementos señalados en los objetivos principales del proyecto con el fin de

poder abarcar cada uno de ellos y suplir sus especificaciones, por medio de una visualización de la operatividad necesaria para tomarlos en cuenta y del mismo modo un análisis de sistemas similares que nos sirvan como eje para tomar en cuenta otros escenarios y posibles mejoras a incluir en dichos requerimientos.

1.4.3. Fase de Desarrollo de Procesos de Trabajo

En la tercera fase se empezará con el desarrollo del proyecto de software como tal, para ello se realizará un recorrido por cada uno de los requerimientos de software levantados en la fase anterior, de acuerdo a estos se ordenará la implementación por medio de módulos que irán cubriendo cada uno de estos. Cada uno de estos módulos a implementar se llevará a cabo haciendo uso de la metodología de desarrollo planteada anteriormente, donde un conjunto de tareas compondrán cada uno de estos módulos de tal manera que el avance del proyecto sea regido por dicha metodología.

- Definición y diseño de la base de datos

Esta será la primera parte de las fases de desarrollo, en esta se realizará el diseño de la base de datos a través de todas sus fases empezando desde el planteamiento de los requerimientos a nivel de datos por medio de un diagrama entidad relación, hasta la implementación en código.

- Módulo de registro de pagos

En este módulo se implementará toda la parte del módulo para el registro de los reportes con los datos de los pagos por los servicios, recolectados por las entidades de convenio, dicho módulo tendrá la tarea de la recepción de dichos reportes su procesamiento y un pequeño recuento relacionado a la consistencia de los datos ingresados en estos.

- Módulo de consultas

El módulo de consultas constara de varias secciones, que permitirán la consulta de información relacionada a los clientes directos de la empresa y también de las entidades de convenio, entre estos datos estará disponible información de procesamiento de pagos, balance de cartera de los clientes, relacionada a sus pagos y facturas, como también consulta de las comisiones generadas a las entidades de convenio por los servicios de recolecta prestados.

- Módulo de administración

El Módulo de administración constara de dos herramientas principales para la gestión de dos partes fundamentales, de las cuales dependen las características principales del aplicativo, gestión de facturas y de comisiones. El acceso a este módulo del aplicativo constara de una validación de ingreso para acceder a la manipulación de estas herramientas.

1.4.4. Fase de Implementación y Validación

Para finalizar esta última fase de implementación y validación se llevará a cabo la codificación de cada uno de los módulos descritos anteriormente con las tecnologías

planteadas y de acuerdo a esto se llevará a cabo la validación del sistema por medio de pruebas funcionales y unitarias que nos permitirán validar su correcto funcionamiento.

2. Marco de referencia

2.1. Marco conceptual

A partir del contexto del aplicativo podemos definir aspectos como el alcance de este en términos de operatividad, el cual estará comprendido o sujeto a una serie de políticas definidas por la compañía que le vaya a dar uso de este, las cuales definirán aspectos como los métodos para el cálculo de comisiones, como también se encargará de brindar los datos, relacionados a los clientes y facturas, necesarios para la correcta operatividad del sistema.

2.1.1. Términos relevantes

2.1.2. Sistemas PGM

Los sistemas PGM son herramientas informáticas diseñadas para facilitar y optimizar la administración de pagos en diferentes contextos, como empresas, instituciones financieras o entidades gubernamentales.

La historia de los sistemas PGM se remonta al desarrollo de los primeros sistemas informáticos para la gestión de pagos en la década de 1950. En sus inicios, estos sistemas se basaban en procesos manuales y en el uso de tecnologías rudimentarias, como tarjetas perforadas y máquinas de contabilidad. Con el avance de la informática, especialmente a partir de la década de 1980, los sistemas PGM evolucionaron hacia soluciones más sofisticadas y automatizadas, que aprovechaban las capacidades de los ordenadores para gestionar grandes volúmenes de transacciones de manera rápida y eficiente.

Hoy en día, los sistemas PGM son herramientas esenciales en cualquier organización que maneje una gran cantidad de transacciones financieras. Estos sistemas ofrecen una amplia gama de funcionalidades, que van desde la gestión de facturas y pagos a proveedores, hasta la administración de nóminas y la conciliación bancaria. Además, los sistemas PGM suelen integrarse con otras herramientas empresariales, como sistemas de contabilidad, CRM (Customer Relationship Management) y ERP (Enterprise Resource Planning), para ofrecer una visión integral de las finanzas y facilitar la toma de decisiones estratégicas.

Entre los aspectos relevantes de los sistemas PGM se encuentran su capacidad para mejorar la eficiencia operativa, reducir los errores y fraudes, agilizar los procesos de pago y cobro, y proporcionar información financiera precisa y oportuna para la toma de decisiones. En resumen, los sistemas PGM son herramientas fundamentales para la gestión financiera de cualquier organización en la era digital.

2.1.3. Antecedentes de aplicaciones PGM

2.1.3.1. PayJoy. PayJoy, como software, es una plataforma integral diseñada para facilitar el acceso al financiamiento a consumidores de bajos ingresos, particularmente para la adquisición de dispositivos móviles. Funciona como un sistema completo que permite a los proveedores de crédito, como fabricantes de dispositivos móviles, operadores de telefonía móvil y otras instituciones financieras, ofrecer planes de pago flexibles y accesibles directamente desde el punto de venta o a través de canales digitales.

Una de las características centrales de PayJoy es su capacidad para evaluar la solvencia crediticia de los consumidores de manera rápida y precisa. Utiliza algoritmos avanzados y análisis de datos alternativos para tomar decisiones informadas sobre la aprobación de créditos, lo que permite un proceso eficiente y ágil para los consumidores.

Además de facilitar la aprobación de créditos, PayJoy también ofrece herramientas completas para la gestión de préstamos. Esto incluye la administración de carteras, la programación de pagos, la generación de informes financieros y el seguimiento de cobranzas. Estas funciones permiten a los proveedores de crédito administrar sus operaciones de manera eficiente y optimizar el rendimiento de sus carteras de préstamos.

2.1.3.2. eCredable. eCredable es una plataforma de tecnología financiera que proporciona soluciones innovadoras para ayudar a las personas a construir y demostrar su historial crediticio utilizando información no tradicional. Como software, eCredable ofrece una amplia gama de herramientas y servicios diseñados para abordar las limitaciones del sistema crediticio convencional y brindar oportunidades de crédito a aquellos que pueden no tener acceso a él de otra manera.

Una característica destacada de eCredable es su enfoque en la utilización de datos alternativos para evaluar la solvencia crediticia de los consumidores. Esto incluye información como el historial de pagos de facturas de servicios públicos, alquileres, seguros y otros gastos recurrentes, que tradicionalmente no se tienen en cuenta en los informes de crédito convencionales. Al aprovechar esta información, eCredable permite a los consumidores construir un perfil crediticio más completo y preciso, lo que puede mejorar sus posibilidades de obtener crédito en el futuro.

2.1.3.3. BillTrust. Billtrust es una plataforma de tecnología financiera que ofrece soluciones innovadoras para la gestión y automatización de procesos de facturación y pagos empresariales. Como software, Billtrust proporciona una amplia gama de herramientas y servicios diseñados para ayudar a las empresas a mejorar la eficiencia, la precisión y la seguridad de sus operaciones financieras.

Una característica destacada de Billtrust es su capacidad para automatizar y simplificar los procesos de facturación y pagos, tanto para las empresas como para sus clientes. La plataforma permite a las empresas crear, enviar y gestionar facturas de manera eficiente, así como recibir pagos de manera rápida y segura a través de una variedad de métodos de pago.

Además de la automatización de procesos, Billtrust ofrece herramientas para ayudar a las empresas a mejorar la gestión de su ciclo de efectivo. Esto incluye funciones como la conciliación de cuentas, la gestión de cobranzas y la generación de informes financieros, que permiten a las empresas optimizar su flujo de efectivo y mejorar su salud financiera en general.

2.2. Marco teórico

2.2.1. Arquitectura software

La arquitectura de software es el diseño estructurado y organizacional de un sistema de software que abarca los componentes, sus relaciones, y los principios y directrices que gobiernan su desarrollo y evolución a lo largo del tiempo. Es como el plano o la estructura fundamental sobre la cual se construye y opera un sistema de software. Este diseño define cómo los diferentes componentes del sistema interactúan entre sí, cómo se comunican, y cómo se distribuyen en hardware o en la nube, entre otras consideraciones importantes.

La importancia de una buena arquitectura de software radica en su capacidad para asegurar que el sistema cumpla con los requisitos funcionales y no funcionales de manera eficiente, robusta, mantenible y escalable. Además, una arquitectura bien diseñada facilita la comprensión del sistema, lo que ayuda a los desarrolladores a trabajar de manera colaborativa y a realizar cambios y mejoras de manera más efectiva a lo largo del ciclo de vida del software. OpenWebinars. (s.f.). Arquitectura de software.

2.2.2. Aplicaciones web SPA

La arquitectura de aplicaciones web de página única (SPA, por sus siglas en inglés) es un enfoque de diseño de software para desarrollar aplicaciones web interactivas y dinámicas que funcionan dentro de un único documento HTML. En una SPA, la mayoría de la lógica de la aplicación se traslada al lado del cliente, lo que significa que gran parte del procesamiento y la interacción con el usuario ocurre en el navegador web, en lugar de en el servidor.

La arquitectura de una SPA típicamente incluye tres componentes principales:

- Cliente (Frontend): Este componente consiste en el código JavaScript, HTML y CSS que se descarga en el navegador del usuario. Aquí es donde reside la lógica de presentación y la interacción del usuario. Las SPAs suelen utilizar frameworks o

bibliotecas de JavaScript como React, Angular, o Vue para gestionar la interfaz de usuario de manera eficiente.

- **Servidor (Backend):** Aunque una SPA traslada gran parte de la lógica de la aplicación al cliente, aún puede necesitar interactuar con un servidor para realizar operaciones como autenticación de usuarios, acceso a bases de datos, o recuperación de datos. El servidor backend puede proporcionar una API RESTful u otro mecanismo de comunicación para permitir que la SPA interactúe con él de manera asincrónica.
- **API (Interfaz de Programación de Aplicaciones):** La API es la interfaz que permite la comunicación entre el cliente y el servidor. Proporciona puntos de conexión bien definidos a través de los cuales la SPA puede enviar solicitudes al servidor y recibir respuestas. La API puede estar basada en tecnologías como REST (Representational State Transfer), GraphQL, o WebSockets.

La arquitectura de una SPA está optimizada para proporcionar una experiencia de usuario fluida y receptiva, minimizando las recargas de página y maximizando la interactividad. Digital55. (s.f.). Qué son Single Page Application (SPA).

2.2.3. Tecnologías de base de datos

Las tecnologías de base de datos son herramientas, sistemas y software utilizados para almacenar, gestionar, recuperar y manipular datos de manera eficiente y segura. Estas tecnologías son fundamentales en la gestión de la información en organizaciones y aplicaciones de software, ya que permiten almacenar grandes volúmenes de datos de manera estructurada y acceder a ellos de manera rápida y precisa cuando sea necesario.

- **Bases de datos relacionales:** Las bases de datos relacionales son las más utilizadas como tecnología para la industria. Están diseñadas para almacenar datos estructurados en tablas relacionadas entre sí. Las tablas están organizadas en filas y columnas y utilizan claves para relacionar los datos entre ellas. Las bases de datos relacionales son muy escalables y ofrecen una gran flexibilidad en cuanto a la forma en que se pueden consultar los datos.
- **Bases de datos NoSQL:** Las bases de datos NoSQL son una alternativa a las bases de datos relacionales. No utilizan tablas y claves como las bases de datos relacionales, sino que utilizan una variedad de estructuras de datos, como documentos, gráficos y pares clave-valor. Las bases de datos NoSQL son muy escalables y ofrecen un rendimiento excelente para grandes conjuntos de datos no estructurados.

Las tecnologías de base de datos incluyen sistemas de gestión de bases de datos (DBMS, por sus siglas en inglés), lenguajes de consulta (como SQL), modelos de datos

(como relacional, NoSQL, etc.) y herramientas de administración y análisis de datos. Hiberus. (s.f.). Tecnologías de base de datos.

2.2.4. Tecnologías web

Las tecnologías web abarcan un conjunto diverso de herramientas, lenguajes, protocolos y estándares que se utilizan para desarrollar y operar aplicaciones y servicios en línea. Estas tecnologías permiten la creación, distribución y presentación de contenido en la web, así como la interacción entre usuarios y sistemas a través de navegadores web. Incluyen elementos fundamentales como HTML (HyperText Markup Language), que define la estructura y el contenido de las páginas web, CSS (Cascading Style Sheets), que establece el aspecto visual y el diseño de las páginas, y JavaScript, un lenguaje de programación que agrega interactividad y dinamismo a las aplicaciones web. Además, las tecnologías web también comprenden frameworks y bibliotecas, protocolos de comunicación como HTTP (Hypertext Transfer Protocol), bases de datos diseñadas para el entorno web y servidores web que proporcionan servicios de alojamiento y distribución de contenido. Estas herramientas son esenciales para el desarrollo y funcionamiento de aplicaciones y servicios en la era digital. MDN Web Docs. (s.f.). Web technology for developers.

3. Requerimientos del sistema

3.1. Requerimientos no funcionales

3.1.1. Usabilidad

- El sistema debe proporcionar herramientas para la gestión de facturas, comisiones y clientes para la empresa prestadora de servicios, permitiendo la consulta de información relacionada a cada uno de estos ítems e interoperabilidad entre ellos. Los actores principales son el administrador del sistema y usuarios autorizados, como el personal contable. Para utilizar estas herramientas, los usuarios deben estar autenticados y autorizados, y el sistema debe tener acceso a la base de datos relevante. Tras la autenticación, el usuario puede acceder al módulo de gestión, consultar información utilizando filtros.
- El sistema debe gestionar el ajuste automático en la cartera de cada cliente a medida que se agreguen facturas por los servicios prestados por la empresa o se registren pagos de dichas facturas por parte de las entidades de convenio. Los actores involucrados son el administrador del sistema y los usuarios autorizados. Para que el sistema realice estos ajustes, los usuarios deben estar autenticados y tener los permisos necesarios. Cada vez que se agregue una factura o se registre un pago, el sistema debe automáticamente actualizar la cartera del cliente correspondiente, asegurando que los saldos reflejan las transacciones más recientes. Este proceso debe incluir la validación de los datos ingresados.

- El sistema debe contar con una interfaz amigable e intuitiva que facilite la comprensión y el uso eficiente de sus funcionalidades. Los actores principales son los usuarios finales, que deben poder navegar y operar el sistema sin dificultad. La interfaz debe estar diseñada de manera que las opciones y herramientas sean fácilmente accesibles, permitiendo a los usuarios realizar sus tareas rápidamente y sin confusiones. Esto incluye menús claros, botones bien etiquetados y una disposición lógica de los elementos en la pantalla.
- El sistema debe poder recibir un archivo de texto con datos de pagos de facturas y procesarlo para actualizar y almacenar la información necesaria. Los actores son los usuarios encargados de cargar estos archivos. Al cargar el archivo, el sistema debe validar y extraer los datos, actualizando los registros de pago correspondientes en la base de datos. Este proceso debe garantizar que la información se persista correctamente, reflejando los pagos realizados tanto consistentes como inconsistentes.

3.1.2. Seguridad

- El sistema debe implementar medidas de seguridad para las consultas, utilizando claves de hash para proteger los datos sensibles. Cada elemento clave de consulta debe estar asociado con una clave de hash, asegurando que solo los usuarios autorizados puedan acceder a la información. Este enfoque protegerá los datos de accesos no autorizados y garantizará la privacidad de la información almacenada.
- El módulo de administración debe contar con un sistema de login que permita validar el ingreso del administrador a las funcionalidades más sensibles del sistema. Para acceder, el administrador debe tener una cuenta activa con credenciales válidas y el sistema debe estar operativo. Una vez que el administrador ingresa su nombre de usuario y contraseña, el sistema valida las credenciales. Si son correctas, el administrador obtiene acceso a funciones críticas como la gestión de facturación, la consulta y creación de facturas, y la generación manual de comisiones. Si las credenciales son incorrectas, se muestra un mensaje de error.

3.2. Requerimientos funcionales

3.2.1. Módulo de registro de pagos

1. Las entidades de convenio podrán acceder al módulo de carga de archivos de registro de pagos mediante su hash de entidad. El sistema debe permitirles cargar el archivo correspondiente, y tras procesar los datos, validar los pagos recolectados. Una vez procesado el archivo, el sistema proporcionará un resumen que mostrará las consistencias e inconsistencias encontradas en los datos.

3.2.2. Módulo de consultas

2. Las entidades de convenio podrán consultar sus pagos procesados utilizando su identificador hash como clave secreta. Para ello, deben ingresar su identificador hash en el módulo de consulta de pagos del sistema. El sistema, al recibir el identificador hash, mostrará una tabla con los pagos asociados a ese identificador. Las entidades podrán filtrar estos pagos entre procesados y no procesados. El sistema actualizará la tabla de acuerdo con los filtros seleccionados, permitiendo una revisión detallada de los pagos.
3. Los clientes directos de la empresa podrán consultar el estado del saldo de su cartera utilizando su correo electrónico y un hash asociado. Los actores principales son los clientes directos y el sistema. Para realizar esta consulta, el cliente debe ingresar su correo electrónico y el hash relacionado en el módulo de consulta de saldo. El sistema, al recibir estos datos, recuperará el saldo del cliente, que depende de factores como las facturas pendientes de pago y cualquier saldo a favor. El sistema mostrará el estado actualizado de la cartera en una interfaz de consulta. La consulta solo podrá realizarse con el correo electrónico y hash correctos, asegurando que la información sea accesible únicamente para el cliente autorizado.
4. Las entidades de convenio podrán consultar las comisiones generadas por el sistema utilizando su identificador hash. Los actores principales son las entidades de convenio y el sistema. Para realizar esta consulta, las entidades deben ingresar su identificador hash en el módulo de consulta de comisiones. El sistema, al recibir esta clave, permitirá acceder a una vista detallada de las comisiones, con la opción de aplicar filtros por fecha. La consulta mostrará información relevante sobre las comisiones, incluyendo los montos a pagar por los servicios prestados. El sistema actualizará la información presentada de acuerdo con los filtros seleccionados, facilitando una revisión detallada de las comisiones generadas. La consulta se realizará exclusivamente con el identificador hash correcto, sin requerir sesiones, asegurando que la información sea accesible sólo para las entidades autorizadas.

3.2.3. Módulo de administración

5. El administrador podrá gestionar el tema relacionado con las facturas mediante el módulo de administración. Los actores principales son el administrador y el sistema. Para gestionar las facturas, el administrador debe iniciar sesión en el módulo de administración, el cual maneja sesiones para funciones críticas. Una vez autenticado, el administrador podrá generar nuevas facturas para los clientes, así como consultar las facturas registradas en el sistema, tanto las pagadas como las pendientes. El sistema permitirá al administrador crear facturas ingresando los detalles necesarios. Además, el administrador podrá acceder a una vista completa de las facturas, filtrando por estado para revisar las facturas pendientes o pagadas.

6. El administrador tendrá acceso en el módulo de administración para gestionar las comisiones de las entidades de convenio, permitiendo la generación manual de comisiones. Los actores principales son el administrador y el sistema. Para llevar a cabo esta gestión, el administrador debe iniciar sesión en el módulo de administración. Una vez autenticado, el administrador podrá crear comisiones manualmente especificando el rango de fechas y la entidad de convenio a la que se le generará la comisión. Tras la generación, el sistema actualizará los registros de comisiones y reflejará las nuevas entradas en el historial de la entidad correspondiente. La gestión de comisiones se realizará a través del módulo de administración, asegurando que solo el administrador autenticado pueda ejecutar estas acciones.

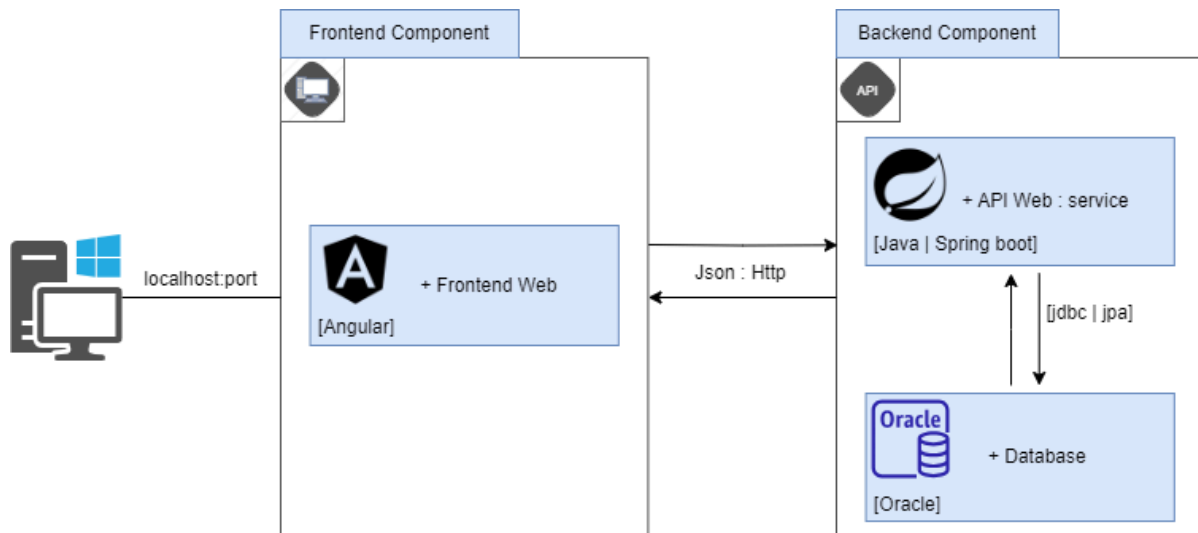
4. Diseño del sistema

Para el desarrollo y continuidad del proyecto se empieza con la definición y diseños de este, dentro de esta sección se tocaran temas como la arquitectura de software que se empleara para dar bases y solución, a través de diferentes esquemas se maquetara la estructura básica que tendrá el proyecto y con esta se podrá proyectar a futuro cómo será el sistema de gestión de recolección de registros de pagos, una vez se le haya dado finalización a este.

Para el desarrollo de esta fase se emplearán distintos gráficos de software para definir cómo funcionará el flujo de envío de información, diagramas entidad relación, para la organización de la información y diagramas de casos de uso para definir flujos del aplicativo.

4.1. Arquitectura software

La arquitectura que se optara a implementar para el desarrollo de este sistema no está muy alejada de la utilizada de manera estándar en las aplicaciones SPA, esta se fundamentara en el modelo de microservicios, cliente servidor para la transmisión y transporte de datos, junto con la integración a una base de datos para la persistencia de los datos.

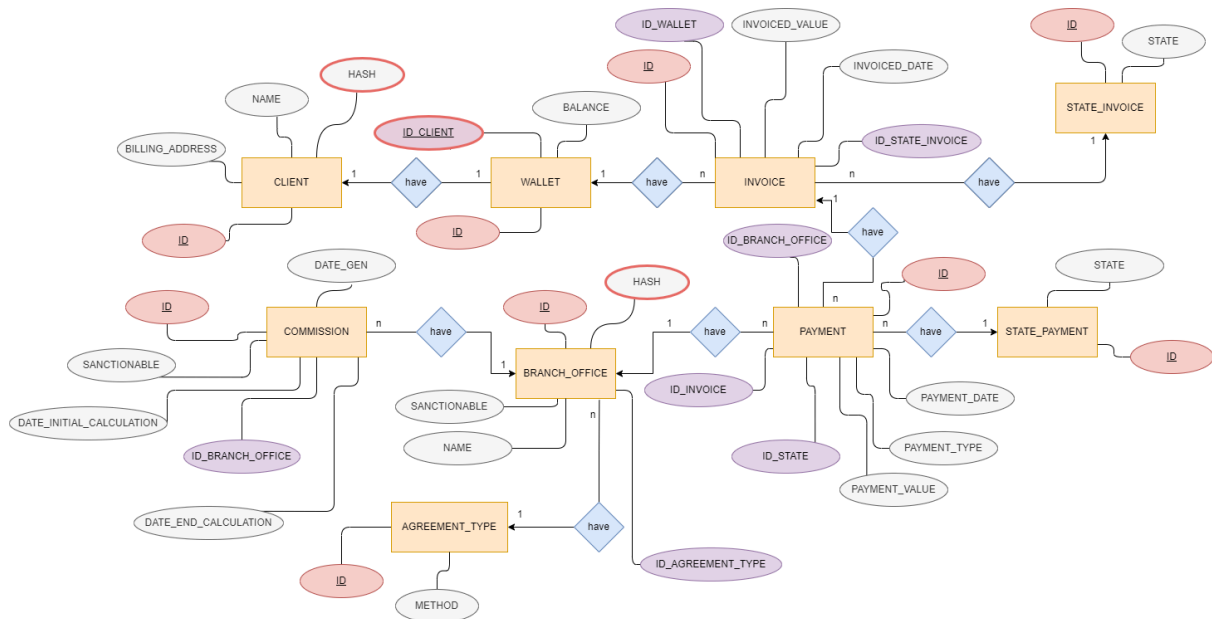
Figura 2*Arquitectura de software del proyecto*

4.2. Diseño de la base de datos

El diseño de la estructura de la base de datos es uno de las fases fundamentales en el diseño y planeación del sistema, ya que de esta dependerá la forma en que se organizara y persiste la información, por tanto la clave en saber definir correctamente este proceso es crucial para no sufrir de imprevistos en una fase más avanzada del proyecto.

El tipo de base de datos que se usará para la implementación del software será un modelo de base de datos relacional, esto es principalmente por la versatilidad que este tipo de modelo presenta y su fácil modelamiento ordenado de los datos por medio de tablas y relaciones entre estas, de acuerdo a esto se plantea el siguiente diagrama entidad relación con el cual se define de manera gráfica y visual la organización de los elementos y entidades, como también la cardinalidad de estas relaciones.

Figura 3
Diagrama entidad-relación de la base de datos



4.3. Diagramas de casos de uso

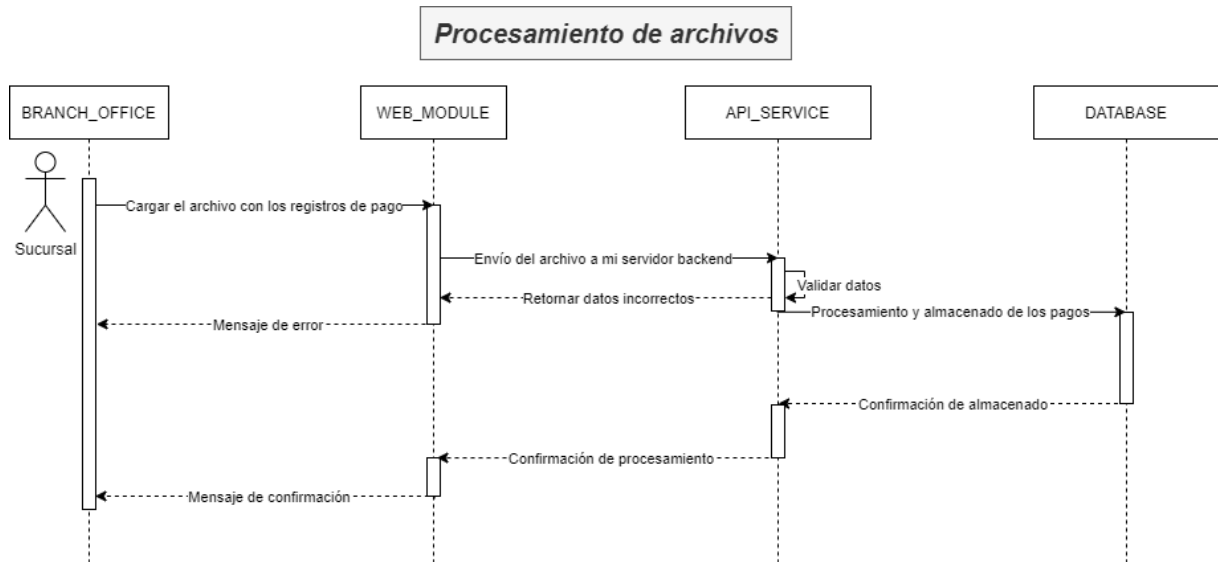
Una vez definido varios de los aspectos anteriores y con el tema de los requerimientos mucho más claro es bueno definir claramente los diagramas de casos de uso, esto con el objetivo de ver de manera más eficiente el flujo que llevará a cabo cada uno de los usuarios de nuestro sistema y poder ver qué flujo o camino lleva a cabo por dentro el sistema cuando se interactúa con alguna de sus funcionalidades.

A continuación tendremos cuatro diagramas de casos de uso con los cuales podremos ver una referencia de cada uno de los procesos ejecutados por las funcionalidades planteadas en los requerimientos.

- **Procesamiento de archivos**

Figura 4

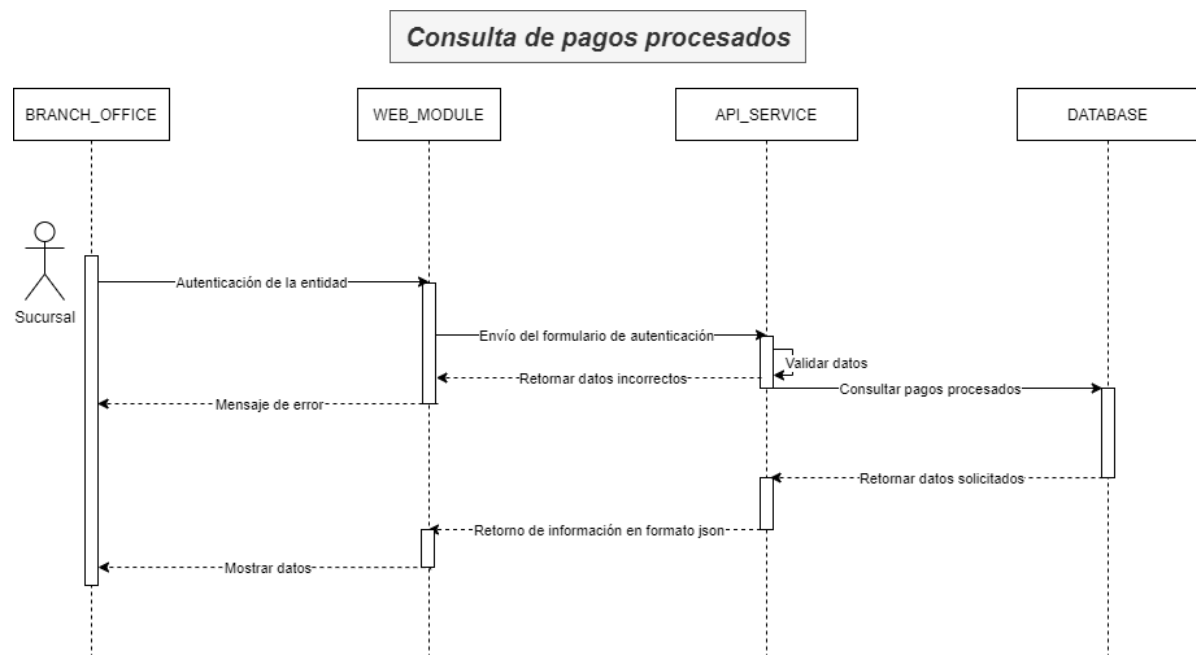
Diagrama de casos de uso para el procesamiento de archivos



- **Consulta de pagos procesados**

Figura 5

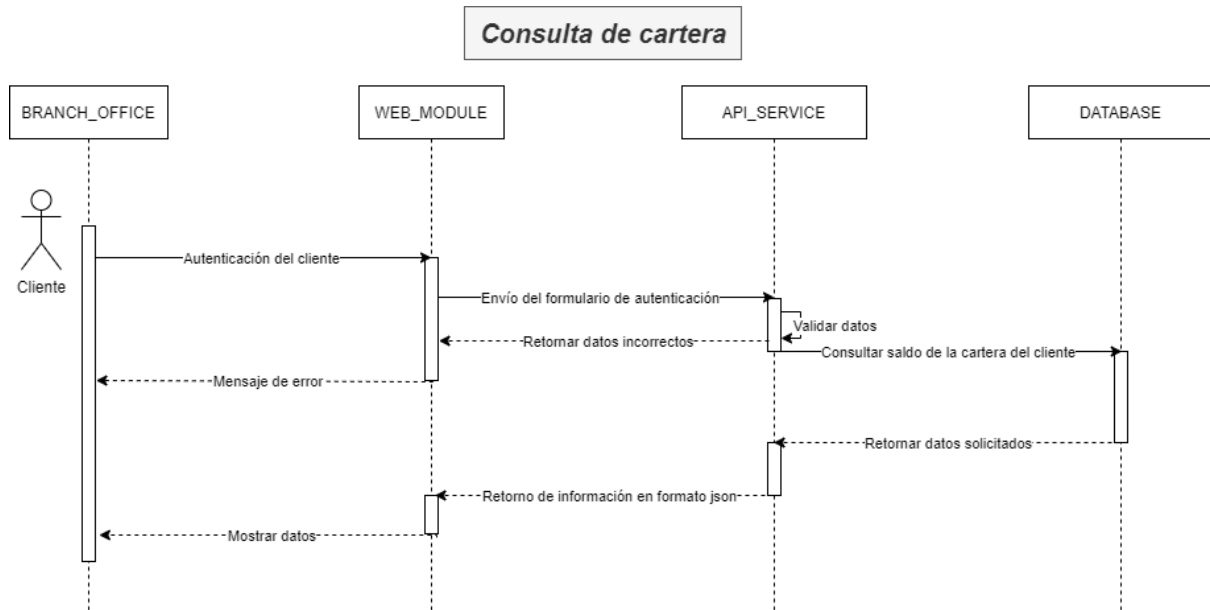
Diagrama de casos de uso para la consulta de pagos procesados



- Consulta de cartera

Figura 6

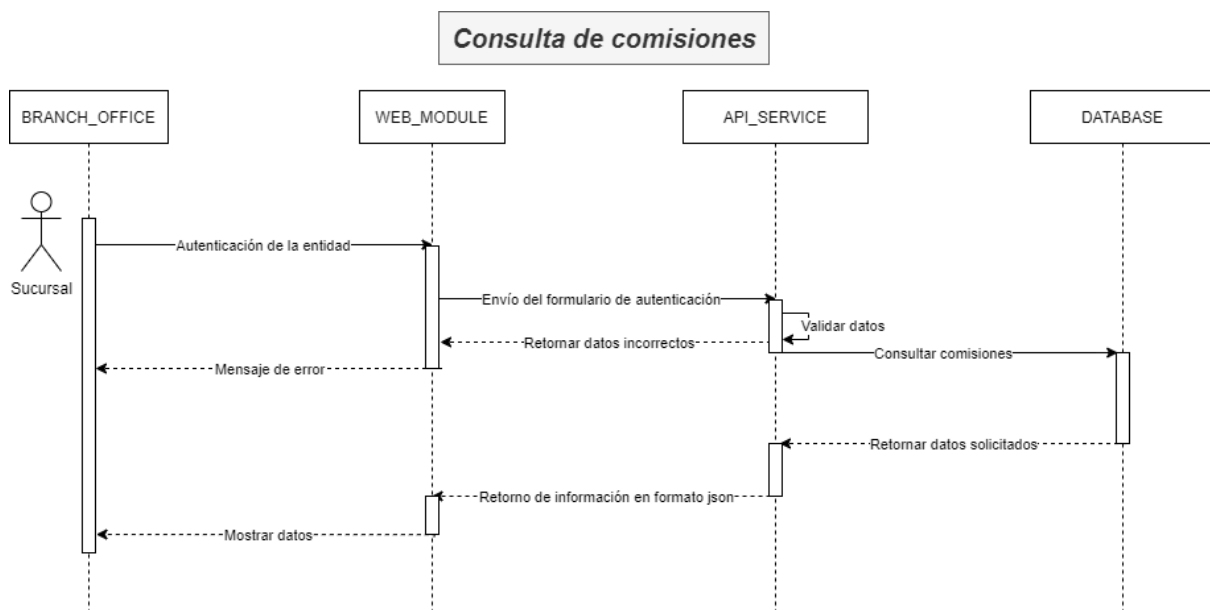
Diagrama de casos de uso para la consulta de cartera



- Consulta de comisiones

Figura 7

Diagrama de casos de uso para la consulta de comisiones



4.4. Diseño UX

Esta etapa de diseño consistirá en el maquetado y la representación visual de cómo quedará el proyecto a nivel de interfaz de usuario, de acuerdo a esto por medio de la herramienta figura se recrean alguna de las vistas de esta interfaz como se muestra a continuación.

Figura 8

Diseño de la vista del módulo de consultas



Nota. La figura muestra el diseño de la pestaña de consultas donde podrás gestionar la consulta de cada una de las secciones pertenecientes al aplicativo.

Figura 9

Diseño de la vista del módulo de registro de pagos

TEHCAMP Inicio Registro de pagos Consultas Admin

Registro

Hash de entidad

Hash

Archivo

Seleccionar archivo...

Continuar

Nota. La figura muestra el diseño de la vista de registro de archivos con el recuento de pagos procesados por parte de la entidad de convenio.

Figura 10

Diseño de la vista del módulo del administrador



Nota. La figura muestra el diseño de la vista de la sección del administrador, desde donde podrá gestionar las facturas manualmente o las comisiones de las entidades de convenio.

Figura 11

Diseño de la vista de la sección de gestión de comisiones

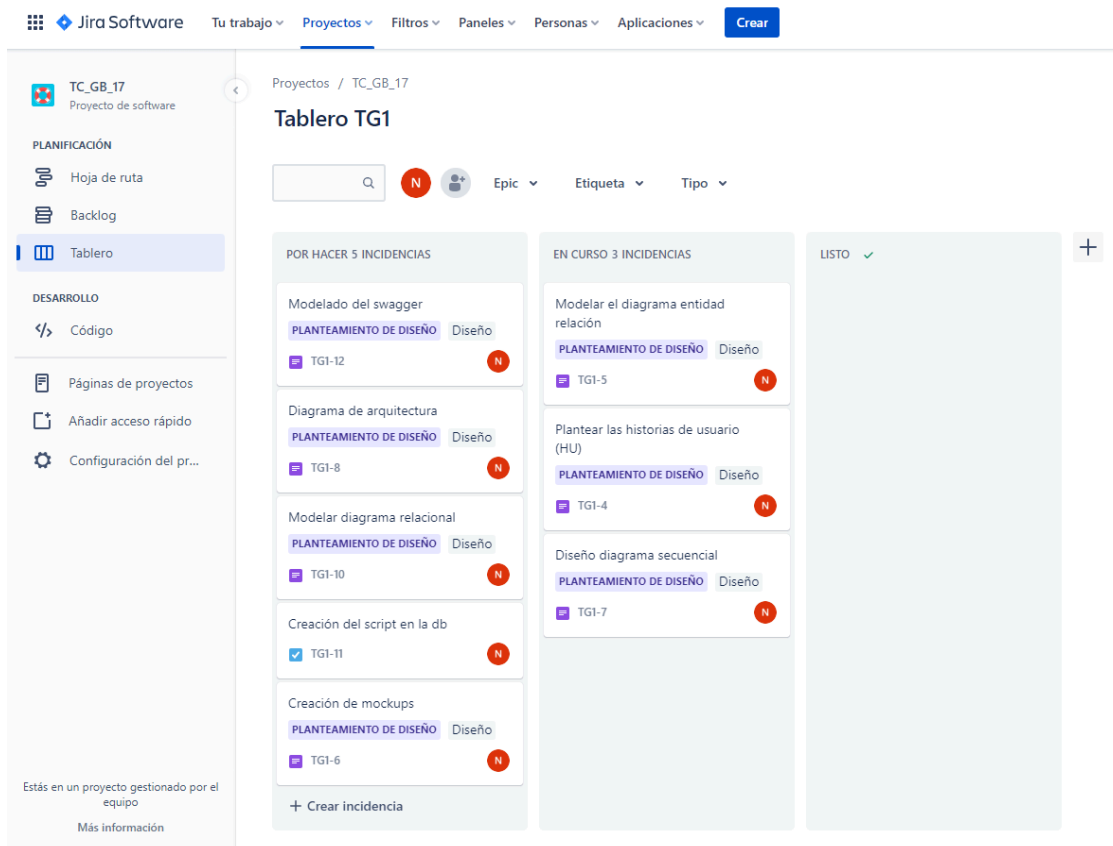
The screenshot shows a web application interface for 'TECHCAMP'. The top navigation bar is green and contains the following items: 'Inicio', 'Registro de pagos', 'Consultas', and 'Admin' (with a pencil icon). A 'salir' button with a red 'X' icon is located in the top right corner. The main content area features a white card titled 'Generar comisiones manualmente'. This card contains a form with the following fields: 'Identificador de la entidad' (with a sub-label 'Identificador'), 'Fecha inicio' (with a sub-label 'dd/mm/yyyy'), and 'Fecha fin' (with a sub-label 'dd/mm/yyyy'). A green button labeled 'Generar reporte' is positioned at the bottom of the form. The background of the card is decorated with faint illustrations of a person, a calculator, a percentage sign, and a money bag.

Nota. En la Figura se ve como será el diseño de la sección para gestionar una comisión manualmente.

4.5. Planteamiento de tareas

En esta etapa de planteamiento de tareas se llevará a cabo la definición de cada uno de los ítems que compondrán el desarrollo técnico del proyecto, a su vez se realizará la definición de las historias de usuario que nos servirán para tener la lista de tareas a nivel de desarrollo a completar, dentro de esta fase de detallamiento de tareas se tendrá principalmente a consideración la planeación clara y puntual que se llevará a cabo.

Figura 12
Tablero Jira



Nota. En la figura se ve la organización de las tareas principales en el tablero.

La mecánica de esta sección se basará en ir completando cada una de las tareas en conjunto o una por una, teniendo en cuenta que habrá tareas que dependen de otras.

5. Implementación

A partir de la metodología planteada se definirán una serie de historias de usuario, que abarcan principalmente el enfoque del desarrollo de la plataforma, para esto tendremos que basarnos en cada uno de los requerimientos clave del sistema, ya que de estos dependen la forma en que se describen estas y que criterios de aceptación serán los que se usen para validar el correcto funcionamiento una vez terminado su desarrollo. A continuación veremos una definición un poco más clara de los requerimientos asociándolos a un rol de usuario dentro de la aplicación y delimitando sus criterios de aceptación.

1. Dado que la entidad bancaria requiere cargar el archivo de pagos, cuando ingrese a la plataforma podrá seleccionar un input donde tendrá la opción de cargar los datos, entonces se abrirá un explorador donde podrá seleccionar el archivo a subir, del mismo modo en el formulario deberá poner su hash de autorización.

Criterios de aceptación:

- El campo de hash debe tener valor, de lo contrario se mostrará un mensaje de error.
 - En el input de archivo se debe adjuntar el archivo de información de lo contrario se mostrará un mensaje de error.
 - El id de la entidad registrado en el encabezado del archivo deberá coincidir con el hash correspondiente a esta, de lo contrario se mostrará un mensaje de error.
2. Como cliente quiero consultar el estado de mi cartera para tener un recuento de mis pagos.

Criterios de aceptación:

- El campo email debe tener un valor de lo contrario se mostrará un mensaje de error.
 - El campo de hash deberá tener valor, de lo contrario se mostrará un mensaje de error.
 - El email y el hash del cliente deben coincidir de lo contrario se mostrará un mensaje de error.
3. Como entidad de convenio quiero ver un recuento de los pagos consistentes e inconsistentes procesados.

Criterios de aceptación:

- El campo de hash de la entidad deberá tener valor de lo contrario se mostrará un mensaje de error.
 - El hash debe corresponder a alguna entidad existente de lo contrario se mostrará un mensaje de error.
4. Como administrador quiero tener control manual de la generación de comisiones para las entidades de convenio.

Criterios de aceptación:

- El campo de administrador deberá tener valor y ser válido, de lo contrario se mostrará un mensaje de error.
5. Como Entidad de convenio quiero poder visualizar las comisiones por los servicios prestados.

Criterios de aceptación:

- El campo de hash de la entidad deberá tener valor de lo contrario se mostrará un mensaje de error.
- El hash debe corresponder a alguna entidad existente de lo contrario se mostrará un mensaje de error.

5.1. Módulo de registro de pagos

El módulo de registro de pagos es una de las partes más importantes del proyecto, ya que en este se cargan los archivos con los datos suministrados por las entidades de convenio y con el procesamiento de dicha información se actualizan múltiples elementos del aplicativo como lo son las carteras de los clientes, comisiones de las entidades de convenio, facturas, entre otras, de acuerdo a esto se diseñó una interfaz intuitiva y sencilla de entender para el usuario con los elementos necesario para que pueda usarla eficientemente, los elementos que puede incluir son los siguientes:

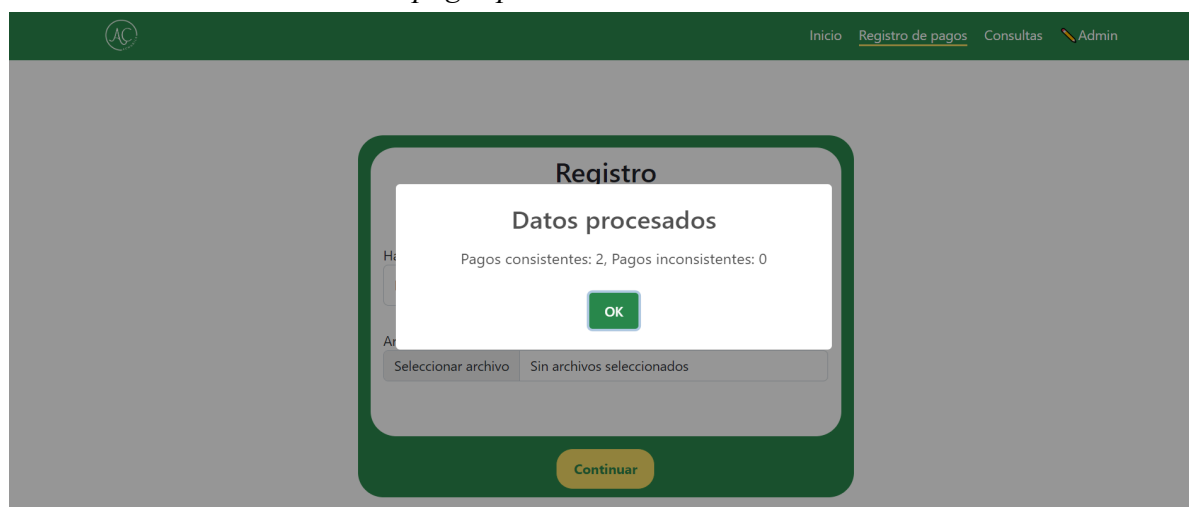
Figura 13

Vista general de la sección de registro de pagos



The image shows a screenshot of a web application interface. At the top, there is a green navigation bar with a logo on the left and menu items 'Inicio', 'Registro de pagos', 'Consultas', and 'Admin' on the right. Below the navigation bar is a white form titled 'Registro' with a green border. The form contains two input fields: 'Hash de la entidad' with a text input field containing 'Hash', and 'Archivo' with a button labeled 'Seleccionar archivo' and a text input field containing 'Sin archivos seleccionados'. At the bottom of the form is a yellow button labeled 'Continuar'.

Nota. La figura muestra la vista del componente usado para registrar los datos de pagos.

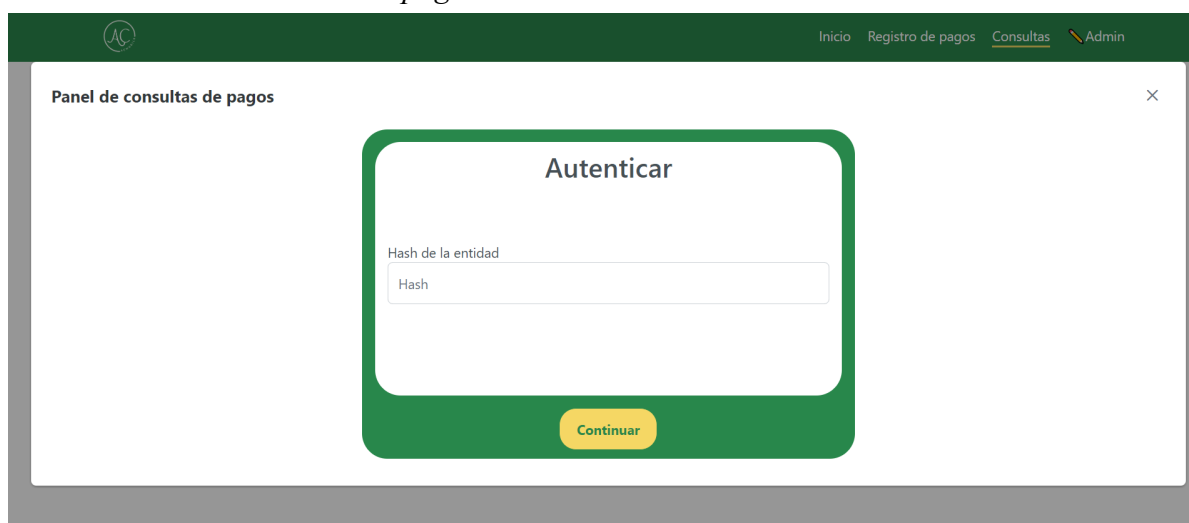
Figura 14*Vista del modal del recuento de pagos procesados*

Nota. La figura muestra el modal de recuento de los pagos procesados por el módulo.

En las figuras anteriores se puede apreciar un poco el flujo en cómo funciona la interacción de la entidad de convenio con el módulo de registro de datos de pagos, esta tiene un formulario con la recepción del archivo y un hash que funciona tipo validador, para saber quién está interactuando con el aplicativo.

5.2. Módulo de consultas

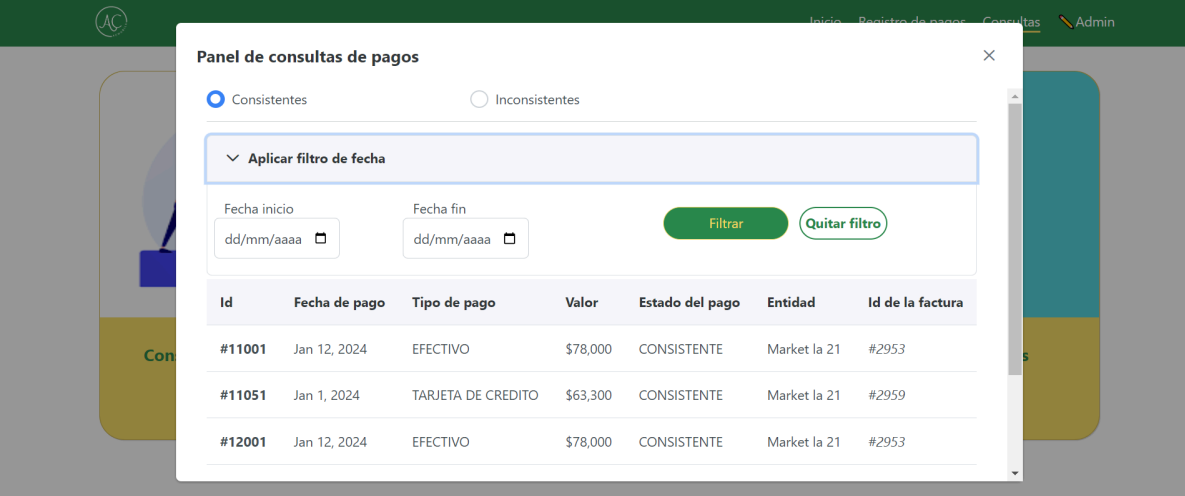
El módulo de consultas desempeña un papel importante en el tema de obtener información relacionada a cualquiera de los aspectos que cubre el proyecto, su utilidad radica principalmente en la versatilidad de los usuarios al interactuar con dicho módulo ya que con pocos pasos pueden acceder a su información deseada.

Figura 15*Vista del modal de consulta de pagos*

Nota. La figura muestra el modal para consultar los pagos procesados por una entidad.

Figura 16

Vista de los resultados de la consulta de pagos procesados



The screenshot shows a modal window titled "Panel de consultas de pagos". At the top, there are two radio buttons: "Consistentes" (selected) and "Inconsistentes". Below this is a section for date filtering with "Fecha inicio" and "Fecha fin" input fields, both containing "dd/mm/aaaa" and a calendar icon. To the right of these fields are two buttons: "Filtrar" (green) and "Quitar filtro" (white with green border). Below the filter section is a table with the following data:

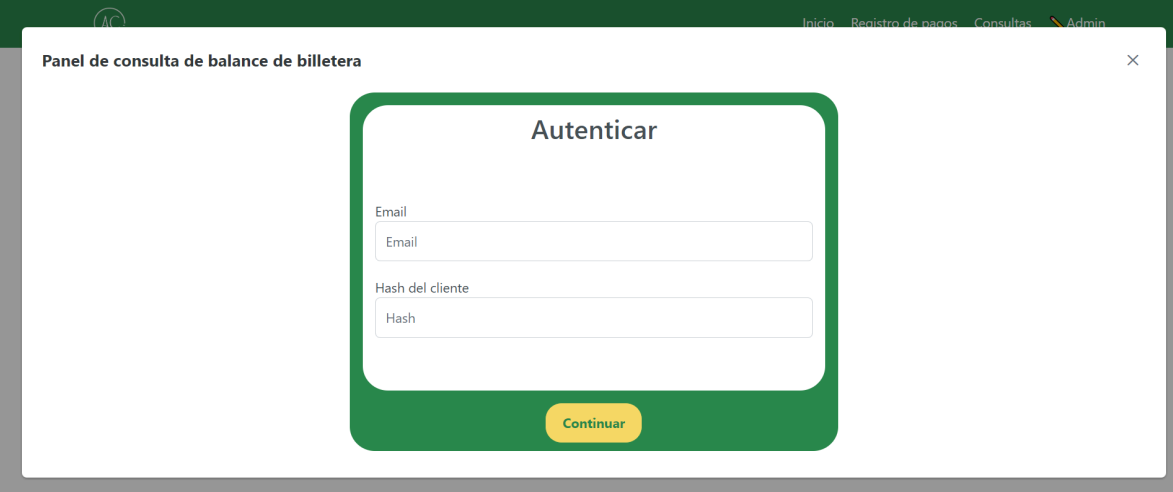
| Id | Fecha de pago | Tipo de pago | Valor | Estado del pago | Entidad | Id de la factura |
|--------|---------------|--------------------|----------|-----------------|--------------|------------------|
| #11001 | Jan 12, 2024 | EFFECTIVO | \$78,000 | CONSISTENTE | Market la 21 | #2953 |
| #11051 | Jan 1, 2024 | TARJETA DE CREDITO | \$63,300 | CONSISTENTE | Market la 21 | #2959 |
| #12001 | Jan 12, 2024 | EFFECTIVO | \$78,000 | CONSISTENTE | Market la 21 | #2953 |

Nota. La figura muestra la tabla con los datos resultantes de la consulta de pagos procesados.

De acuerdo a las figuras anteriores podemos ver la operabilidad que tendremos con la tabla resultante de la consulta de los pagos procesados, en esta se puede apreciar elementos que nos brindan información como la fecha del pago, el valor pagado, tipo de pago, identificador de la factura asociada al pago, entre otros datos importantes, del mismo modo tenemos un componente de filtrado que nos permitirá acotar la consulta a una fecha en específico, la cual se aplicará a la fecha asignada al pago, también podemos aplicar otro tipo de filtro para saber más información de los pagos que fueron inconsistentes.

Figura 17

Vista del modal de consulta de balance de billetera del cliente

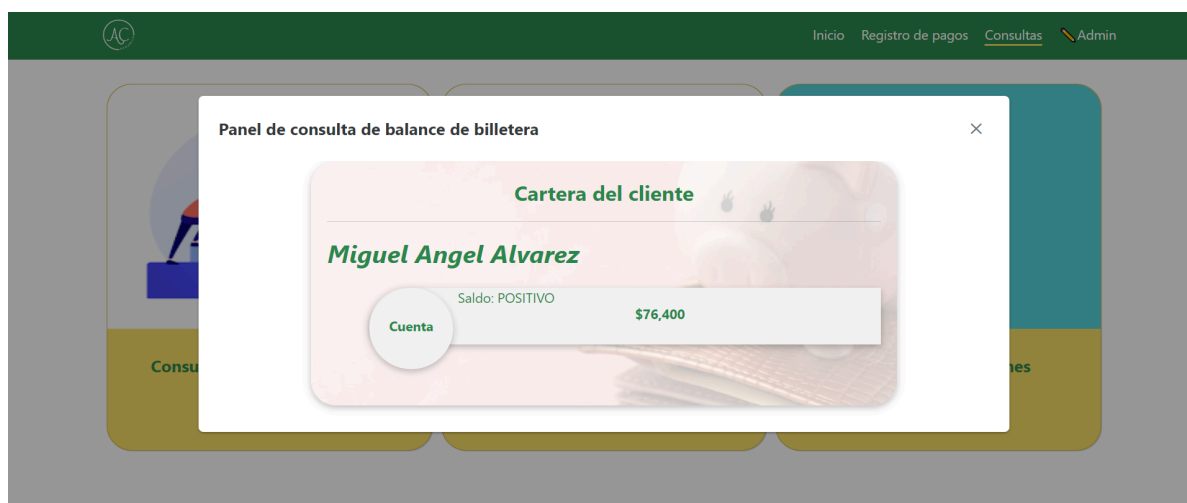


The screenshot shows a modal window titled "Panel de consulta de balance de billetera". Inside the modal is a green-bordered box with the title "Autenticar". Below the title are two input fields: "Email" and "Hash del cliente". The "Email" field contains the text "Email" and the "Hash del cliente" field contains the text "Hash". At the bottom of the green box is a yellow button labeled "Continuar".

Nota. La figura muestra el modal con el formulario para consultar el balance de billetera.

Figura 18

Vista de los resultados de la consulta de balance de billetera

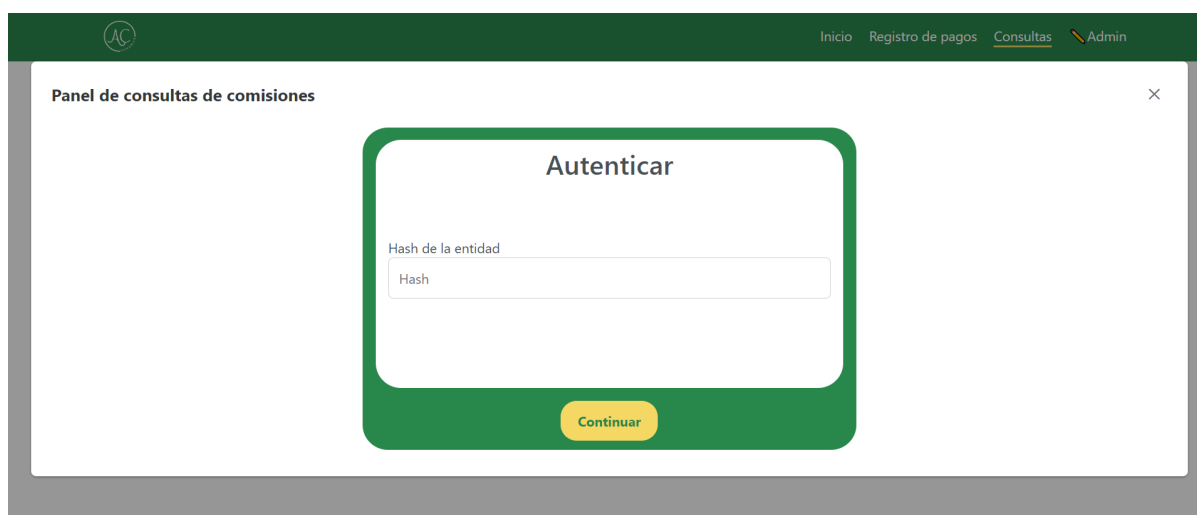


Nota. La figura muestra el modal con el recuento del balance del cliente.

En las figuras anteriores muestran el flujo esperado que tendrá un usuario que quiera interactuar en el módulo de consultas con la sección de consulta de balance de billetera, en esta, una vez diligenciado el formulario, se nos mostrará un modal con los dato de cartera relacionados al cliente, aca podremos ver si este posee un saldo positivo o negativo y a su vez el monto total de este.

Figura 19

Vista del modal de consulta de comisiones



Nota. La figura muestra el modal con el formulario de consulta de comisiones.

Figura 20*Vista de los resultados de la consulta de comisiones*

Panel de consultas de comisiones

▼ Aplicar filtro de fecha

Fecha inicio: dd/mm/aaaa

Fecha fin: dd/mm/aaaa

Filtrar Quitar filtro

| Id | Valor | Fecha de generación | Fecha inicial | Fecha final | Entidad | Sancionable | Metodo de convenio |
|--------|-------|---------------------|---------------|-------------|--------------|-------------|--------------------|
| #16001 | \$0 | Apr 1, 2024 | Mar 2, 2024 | Apr 1, 2024 | Market la 21 | No | Metodo 2 |
| #17001 | \$0 | May 1, 2024 | Apr 1, 2024 | May 1, 2024 | Market la 21 | No | Metodo 2 |

<< < 1 > >>

Numero de registros: 2

Nota. La figura muestra la tabla con los resultados de la consulta de comisiones.

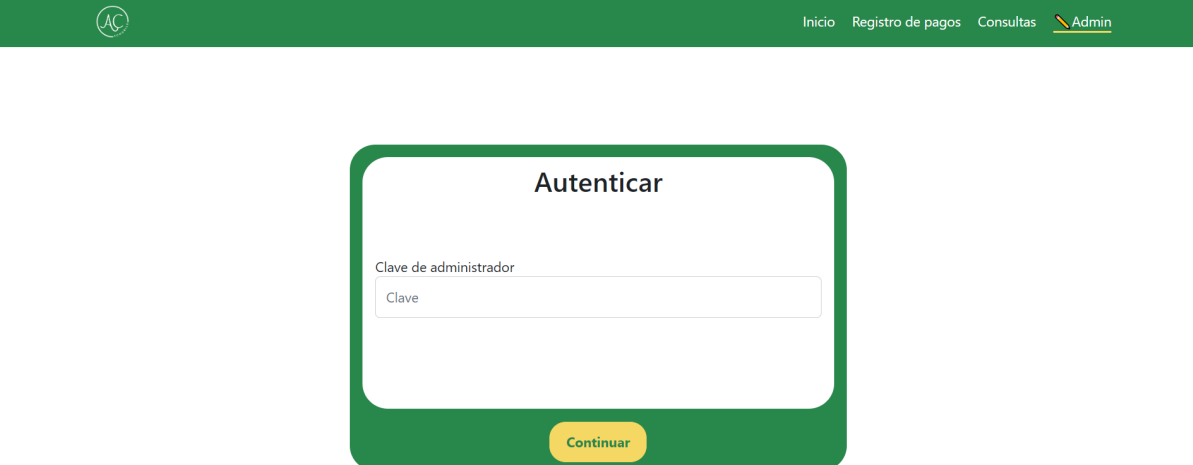
En la figuras anteriores podemos apreciar la tabla con los datos relacionados a las comisiones asignadas a cierta entidad de convenio, como también el modal con el formulario donde está accede a dicha información, entre la cual están datos de la fecha de generación de dicha comisión, el monto, el rango de fechas que se usó para realizar el cálculo y otros datos relevantes para el usuario que esté consultando dicha información, del mismo modo cuenta con su componente de filtrado que hace más útil el uso del modal de consulta.

5.3. Módulo de administración

Como parte fundamental del modelo del proyecto está el módulo de administración, el cual es crucial para poder interactuar con ciertos elementos del aplicativo que dan versatilidad a la forma en que funciona el flujo de este. El módulo de administración se implementó para organizar y administrar principalmente dos aspectos, las facturas y las comisiones de las entidades de convenio.

Figura 21

Vista del login de acceso al módulo de consultas



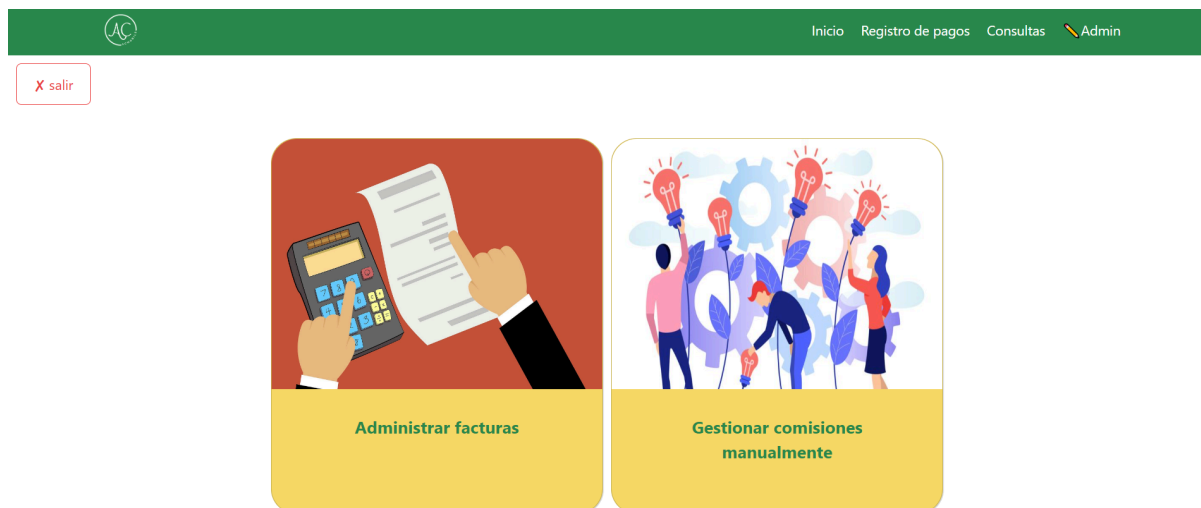
The screenshot shows a green header bar with a logo on the left and navigation links: 'Inicio', 'Registro de pagos', 'Consultas', and 'Admin' (highlighted with a red underline). Below the header is a white rounded rectangle titled 'Autenticar'. It contains a label 'Clave de administrador' above a text input field labeled 'Clave'. At the bottom of the rectangle is a yellow button labeled 'Continuar'.

Nora. La figura muestra el formulario de acceso al módulo de administración.

En la figura anterior tenemos un formulario para acceder al módulo de administración el cual requiere de una contraseña para dar acceso a este, acá se tiene una implementación de cierto nivel de seguridad para el acceso a esta sección del aplicativo que puede tener funcionalidades que no deberían ser accedidas por cualquier usuario.

Figura 22

Vista del módulo de administración



Nota. La figura muestra la primera vista del módulo de administración.

Figura 23*Vista de la sección de administración de facturas*

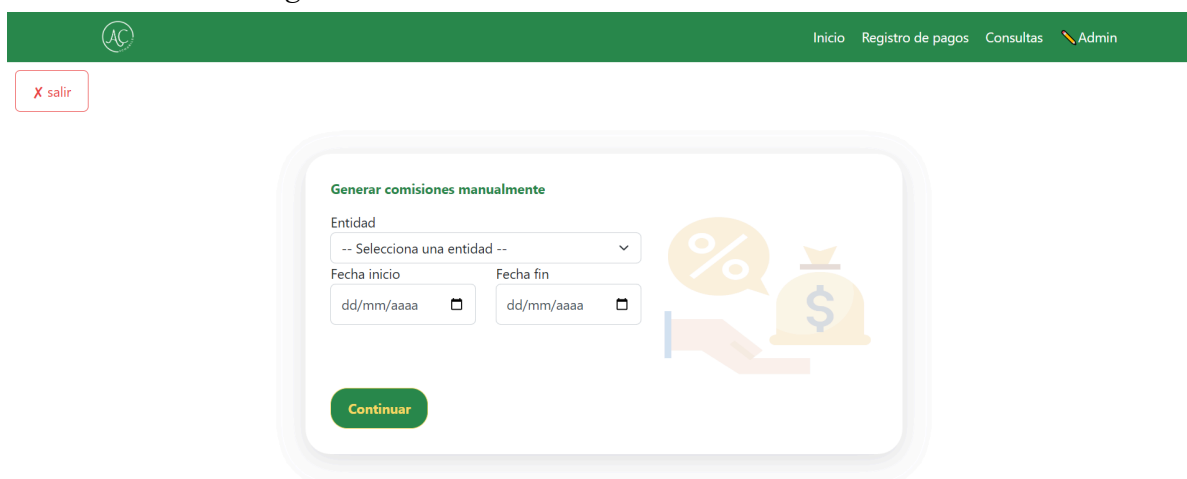
| Id | Valor | Fecha de facturación | Id del cliente | Cliente | Estado de la factura |
|-------|-----------|----------------------|----------------|-----------------------|----------------------|
| #2952 | \$80,000 | Mar 6, 2024 | #1001 | Manolo Felipe Garcia | PENDIENTE |
| #2954 | \$89,000 | Mar 13, 2024 | #1151 | Carla Yucatan Gimenez | PENDIENTE |
| #2955 | \$140,000 | Mar 5, 2024 | #1201 | Marina Soto Quispe | PENDIENTE |

Nota. La figura muestra los elementos que componen a la sección de facturas.

Figura 24*Vista del modal de nueva factura*

Nota. La figura muestra el modal con el formulario para añadir una nueva factura.

En las figuras anteriores podemos ver todas las vistas de la sección de gestión de facturas en el módulo de administración, de acuerdo a estas podemos apreciar cómo esta se compone a partir de los elementos de cada una de las figuras, en la parte de la visualización de facturas vemos información relevante relacionada a estas, como datos del cliente al cual pertenece, valor facturado, pendiente a pagar y su fecha, con esta tabla le ofrecemos datos relevantes al usuario administrador para que pueda revisarlas, a su vez tenemos varias opciones de filtrado, a nivel de fecha y de estado de la factura. Por otro lado veremos el modal disponible dentro de esta sección para la creación manual de facturas, en este tenemos varios datos necesarios para diligenciar, todas estas herramientas son útiles para poder dar una gestión más sencilla a la generación de facturas a los clientes de la empresa.

Figura 25*Vista de la sección de gestión de comisiones manualmente*

The screenshot shows a web application interface for generating manual commissions. At the top, there is a green navigation bar with the logo 'AC' and menu items: 'Inicio', 'Registro de pagos', 'Consultas', and 'Admin'. Below the navigation bar, there is a red button labeled 'X salir'. The main content area is a white rounded rectangle with a light green border. Inside, the title 'Generar comisiones manualmente' is displayed. Below the title, there is a form with three fields: 'Entidad' (a dropdown menu with the placeholder '-- Selecciona una entidad --'), 'Fecha inicio' (a date picker with the format 'dd/mm/aaaa'), and 'Fecha fin' (a date picker with the format 'dd/mm/aaaa'). A green 'Continuar' button is located at the bottom left of the form. To the right of the form, there is an illustration of a hand holding a money bag with a dollar sign and a percentage symbol.

Nota. La figura muestra el formulario donde se puede generar una comisión de manera manual.

En la figura anterior tenemos otra de las funcionalidades disponibles para el administrador, en esta encontrará un formulario para generar una comisión, a una entidad de convenio, manualmente, para ello deberá especificar tanto la entidad a la que se le quiere generar dicha comisión como el rango de fechas aplicables, donde se hará un recuento de los pagos procesados y el cálculo de la comisión depende directamente de este, el formulario es bastante sencillo e intuitivo de diligenciar y se sustenta de varias validaciones para apoyar la experiencia de usuario.

A continuación se adjuntan los links del código fuente del aplicativo, tanto del componente backend como frontend:

Backend: <https://github.com/NicolasBaezMora/InvoiceProject>

Frontend: <https://github.com/NicolasBaezMora/InvoiceProjectFrontend>

6. Validación del sistema

Una vez finalizado el plan de desarrollo del proyecto y todas las funcionalidades especificadas dentro de los requerimientos sean implementadas, se tiene que seguir con un paso fundamental dentro del ecosistema del desarrollo de software, el cual es el plan de pruebas, donde se realizarán las verificaciones necesarias y se evaluarán las funcionalidades del aplicativo para ver que funcionen correctamente.

6.1. Plan de pruebas

En esta sección del plan de pruebas se establece ciertas estrategias para abordar todo el tema de pruebas del software, para ello empezaremos con el desarrollo de las pruebas

unitarias a nivel de lógica de microservicios y funcionalidad, de esta manera validamos el correcto funcionamiento interno de la lógica de negocio, para esto haremos uso de ciertas librerías útiles para el unit testing a nivel de backend. Por otra parte para llevar a cabo las pruebas del frontend del aplicativo o interfaz de usuario realizaremos pruebas funcionales manualmente con este, para así validar que la interacción del usuario con el aplicativo será la correcta.

6.2. Resultado de pruebas unitarias

En este primer paso de validación del sistema, entramos en la sección de resultados de las pruebas unitarias. Pero antes de entrar en el análisis de dichos resultados, es importante entender el contexto y los métodos utilizados durante el proceso de prueba. A continuación, se describen los componentes y las herramientas empleadas, así como el enfoque metodológico adoptado para garantizar una cobertura de pruebas adecuada.

Contexto y Herramientas Utilizadas

- **Spring Boot:** Utilizado para construir el proyecto debido a su capacidad de simplificar el desarrollo de aplicaciones basadas en Spring, facilitando la configuración y la gestión de dependencias.
- **Java:** Elegido por su robustez, eficiencia y amplio soporte en el ecosistema de desarrollo de aplicaciones empresariales.
- **JUnit:** Utilizado como el principal framework de pruebas unitarias. Permite la creación de casos de prueba estructurados y la ejecución automatizada de los mismos.
- **Mockito:** Empleado para la creación de mocks y la simulación de comportamientos en pruebas unitarias, permitiendo el aislamiento de componentes y la validación de lógica específica sin depender de implementaciones reales.

Metodología de Pruebas

La metodología de pruebas adoptada se basó en los siguientes principios:

- Cada componente fue probado de forma aislada para garantizar que su funcionalidad específica opere correctamente, independientemente de las dependencias externas.
- Se utilizó una herramienta de cobertura de código para medir el porcentaje de código validado durante las pruebas, asegurando que se abarque una cantidad significativa de la base de código.
- Se diseñaron pruebas tanto para escenarios en los que se esperaba un comportamiento correcto (escenarios positivos) como para aquellos en los que se esperaban errores o excepciones (escenarios negativos), garantizando así la robustez del sistema.

Resultados Esperados

Antes de presentar los resultados detallados, es fundamental definir los criterios de éxito de las pruebas unitarias. Los resultados esperados incluyen:

- Un alto porcentaje de casos de prueba exitosos, lo que indicaría una correcta implementación y comportamiento esperado de los componentes.
- Alcanzar una cobertura mínima del 80%, asegurando que la mayoría del código ha sido verificado y probado.

Figura 26

Recuento del coverage de pruebas unitarias

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed Cxty | Missed Lines | Missed Methods | Missed Classes |
|---|---------------------|------|-----------------|------|-------------|--------------|----------------|----------------|
| com.proyectoTC.Taller_17_TC.dtos | | 67% | | n/a | 31 168 | 1 74 | 31 168 | 0 18 |
| com.proyectoTC.Taller_17_TC.models | | 68% | | n/a | 28 168 | 1 74 | 28 168 | 0 18 |
| com.proyectoTC.Taller_17_TC.utils | | 50% | | 15% | 12 25 | 37 79 | 3 15 | 0 4 |
| com.proyectoTC.Taller_17_TC.response_models | | 46% | | n/a | 17 30 | 13 19 | 17 30 | 0 4 |
| com.proyectoTC.Taller_17_TC.services | | 95% | | 87% | 3 42 | 2 82 | 2 38 | 0 7 |
| com.proyectoTC.Taller_17_TC | | 83% | | n/a | 1 5 | 2 8 | 1 5 | 0 2 |
| com.proyectoTC.Taller_17_TC.validators | | 95% | | 100% | 0 8 | 1 14 | 0 4 | 0 1 |
| com.proyectoTC.Taller_17_TC.converters | | 100% | | 100% | 0 48 | 0 142 | 0 30 | 0 10 |
| com.proyectoTC.Taller_17_TC.controllers | | 100% | | n/a | 0 24 | 0 90 | 0 24 | 0 7 |
| com.proyectoTC.Taller_17_TC.configuration | | 100% | | n/a | 0 20 | 0 38 | 0 20 | 0 3 |
| com.proyectoTC.Taller_17_TC.exceptions | | 100% | | n/a | 0 15 | 0 30 | 0 15 | 0 3 |
| Total | 997 of 4.408 | 77% | 18 of 72 | 75% | 92 553 | 57 650 | 82 517 | 0 77 |

Nota. En la figura se muestra una tabla con el recuento del código cubierto por pruebas.

En la figura anterior nos encontramos con una tabla de la cual se puede extraer bastante información relevante, relacionada al recuento de las pruebas unitarias. Esta tabla nos proporciona una visión detallada sobre el estado actual de la cobertura de código en nuestra aplicación. A continuación, se describen los elementos más importantes presentes en la tabla y cómo interpretarlos.

Descripción de la Tabla de Cobertura:

1. Clases:

- **Total de Clases:** Número total de clases en el proyecto que han sido consideradas durante la fase de pruebas.
- **Porcentaje de Cobertura:** Proporción de clases probadas respecto al total de clases, expresada en porcentaje. Este valor es crucial para entender la extensión de las pruebas unitarias a nivel de clase.

2. Métodos:

- **Total de Métodos:** Número total de métodos presentes en las clases del proyecto.

- **Porcentaje de Cobertura:** Proporción de métodos probados respecto al total de métodos, proporcionando una indicación de la profundidad de las pruebas unitarias a nivel de método.

3. Líneas de código:

- **Total de Líneas:** Total de líneas de código presentes en el proyecto.
- **Porcentaje de Cobertura:** Proporción de líneas de código cubiertas respecto al total de líneas, indicando la exhaustividad de las pruebas en términos de líneas de código verificadas.

De acuerdo a todos estos elementos extraídos de la tabla de información del coverage de las pruebas unitarias del proyecto podemos interpretar un poco estos resultados, el porcentaje global de cobertura de código es un indicador clave del estado de las pruebas unitarias. Un alto porcentaje sugiere que una gran parte del código ha sido sometido a pruebas, mientras que un porcentaje bajo puede indicar áreas del código que necesitan más atención y pruebas adicionales en este caso vemos que se alcanzó un coverage del 77%, el cual está bastante cerca de un coverage ideal del 80%, no obstante con este valor podemos dar certeza a que gran parte del código se encuentra probado.

6.3. Resultado de pruebas de interfaz

En esta sección, dando continuación al tema de las pruebas del sistema, nos enfocaremos en las pruebas de la interfaz web. Estas pruebas se llevarán a cabo de manera manual, comprobando cada uno de los flujos que componen el aplicativo para asegurar el correcto funcionamiento visual y operativo del mismo. El objetivo principal de las pruebas de interfaz es garantizar que todos los elementos visuales y funcionales del sistema web operen de manera correcta y sin errores. Esto incluye la verificación de botones, enlaces, formularios, y la correcta presentación de los datos en la interfaz de usuario (UI).

Las pruebas se realizan siguiendo unos pasos simples. Primero, explicaremos la interfaz web para identificar todas las áreas clave y los componentes visuales importantes que necesitan ser verificados. Luego, llevaremos a cabo pruebas de usabilidad para asegurarnos de que la interfaz sea accesible y fácil de usar para los usuarios. En cuanto a las pruebas funcionales, nos aseguraremos de que todos los botones y enlaces funcionen correctamente y que los formularios aceptan y procesan datos de manera adecuada.

A continuación iremos enumerando y describiendo cada uno de los componentes a probar.

Figura 27*Componente de registro de pagos procesados*

Registro

Hash de la entidad

Hash

El hash es necesario

Archivo

Seleccionar archivo Sin archivos seleccionados

Campo invalido

Continuar

Nota. En la figura se muestra el formulario con los campos requeridos

En esta imagen podemos observar la prueba de validaciones del formulario de registro de pagos, aca podemos observar que si queremos darle click al botón continuar para procesar la información y alguno de los campos del formulario no ha sido llenado saltaran los validadores de cada uno de estos, rectificando así el correcto funcionamiento del componente a nivel visual y funcional.

Figura 28*Modal de validación del componente de registro de pagos*

Ocurrió un error

El hash de la entidad 'Market la 21' no es valido

OK

Continuar

Nota. En la figura se muestra un modal de validación del componente

Ahora situándonos en el mismo componente realizamos una segunda prueba del correcto funcionamiento de los validadores del formulario, para este escenario cargamos un archivo con pagos procesados y especificamos un hash de otra entidad de convenio a la cual

no le corresponde dicho archivo, de acuerdo a esto tenemos como resultado que el componente devuelve un mensaje de error de validación dentro del modal que se ve en la imagen anterior, con lo cual podemos validar su correcto funcionamiento de validación.

Figura 29

Componente de consulta de pagos

The image shows a modal window titled 'Autenticar' within a 'Panel de consultas de pagos'. It features a text input field labeled 'Hash de la entidad' with the placeholder text 'Hash'. Below the input field, a red error message reads 'El hash es necesario'. At the bottom of the modal is a green button labeled 'Continuar'.

Nota. En la figura se muestra el formulario con los campos requeridos

Acá ya empezamos a validar el funcionamiento de los componentes de la sección de consulta, empezando con el apartado de consulta de pagos procesados, para esto vemos que el formulario de consulta tiene un campo requerido de manera obligatoria, el cual es el hash de la entidad y si se intenta darle click al botón de continuar no se podrá efectuar la consulta y se disparara el validador de dicho formulario.

Figura 30

Modal con los datos de pagos procesados

The image shows a modal window titled 'Panel de consultas de pagos' with a close button (X) in the top right. It has two radio buttons: 'Consistentes' (selected) and 'Inconsistentes'. Below this is a section for applying a date filter, 'Aplicar filtro de fecha', with 'Fecha inicio' (07/06/2024) and 'Fecha fin' (04/06/2024) fields. A red error message '¡Rango invalido!' is displayed below the date fields. There are 'Filtrar' and 'Quitar filtro' buttons. Below the filter section is a table with the following data:

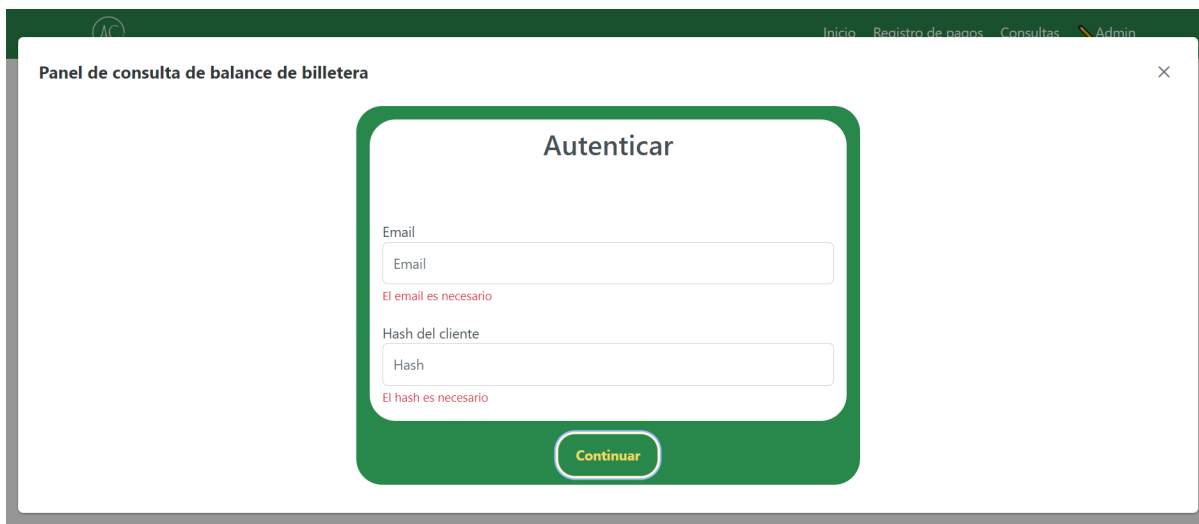
| Id | Fecha de pago | Tipo de pago | Valor | Estado del pago | Entidad | Id de la factura |
|--------|---------------|--------------------|----------|-----------------|--------------|------------------|
| #11001 | Jan 12, 2024 | EFFECTIVO | \$78,000 | CONSISTENTE | Market la 21 | #2953 |
| #11051 | Jan 1, 2024 | TARJETA DE CREDITO | \$63,300 | CONSISTENTE | Market la 21 | #2959 |
| #12001 | Jan 12, 2024 | EFFECTIVO | \$78,000 | CONSISTENTE | Market la 21 | #2953 |

Nota. En la figura se ve la tabla de pagos procesados con su filtro

En la imagen anterior vemos uno de los elementos de validación del componente de la consulta de pagos, el cual es el filtrado, acá vemos que otro de los elementos de este componente está cubierto y es el correcto diligenciamiento de las fechas de dicho filtro, donde a menos que la fecha de inicio y de fin tengan un orden correcto este no se aplicará a la tabla de datos.

Figura 31

Componente de consulta de cartera

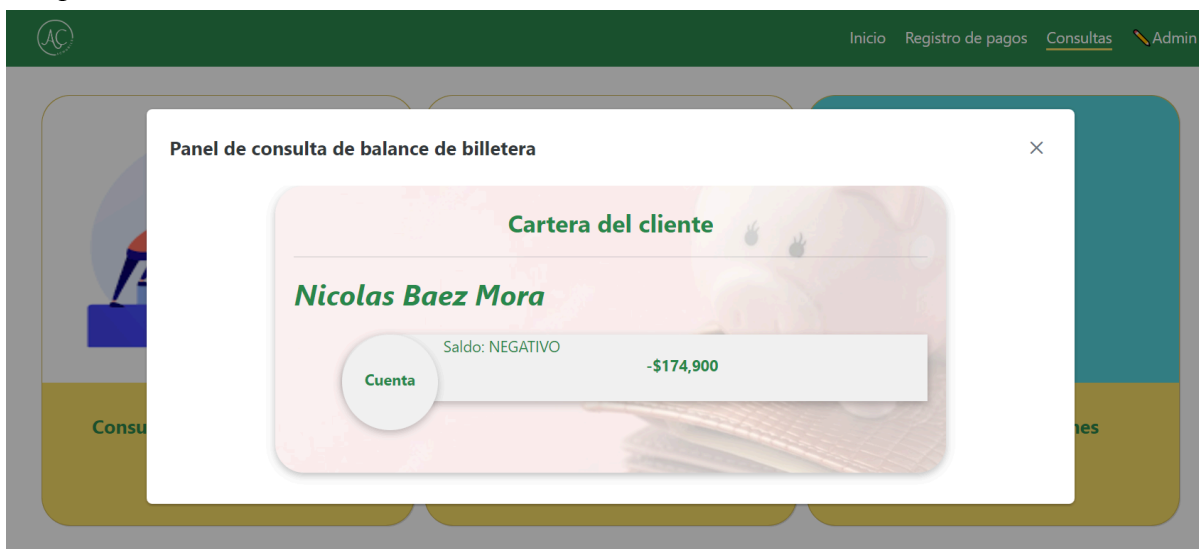


The screenshot shows a modal window titled "Panel de consulta de balance de billetera" with a close button (X) in the top right corner. Inside the modal, there is a green-bordered box titled "Autenticar". This box contains two input fields: "Email" and "Hash del cliente". The "Email" field has a red error message below it that reads "El email es necesario". The "Hash del cliente" field also has a red error message below it that reads "El hash es necesario". At the bottom of the green box is a green button labeled "Continuar". The background of the modal is white, and the overall interface has a dark green header with navigation links: "Inicio", "Registro de pagos", "Consultas", and "Admin".

Nota. En la figura se muestra el formulario con los campos requeridos

Figura 32

Componente de consulta de cartera



The screenshot shows the same modal window as in Figure 31, but now it displays the client's account balance. The modal is titled "Panel de consulta de balance de billetera" with a close button (X) in the top right corner. Inside, there is a light pink box titled "Cartera del cliente". Below the title, the client's name "Nicolas Baez Mora" is displayed in green. Underneath the name, the balance is shown as "Saldo: NEGATIVO" followed by "-\$174,900" in green. To the left of the balance, there is a circular icon labeled "Cuenta". The background of the modal is white, and the overall interface has a dark green header with navigation links: "Inicio", "Registro de pagos", "Consultas", and "Admin".

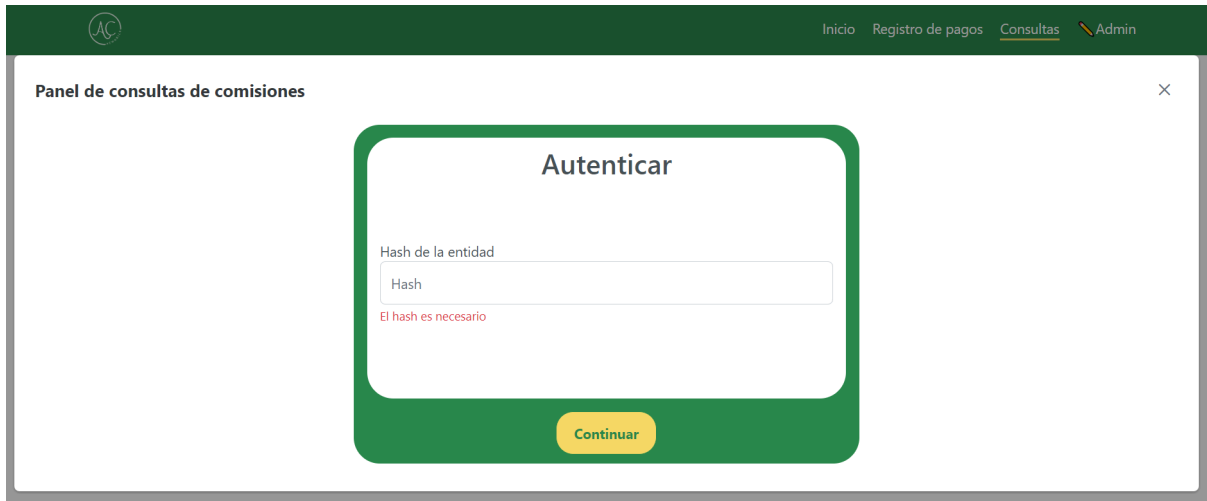
Nota. En la figura se muestra el componente de consulta de balance de cartera del cliente

En la figura anterior veremos los validadores aplicados a la sección de consulta de balance de cartera, en la cual para hacerla efectiva se deberá suministrar un email y un hash relacionado al cliente que desee consultar dicha información, de esta manera podemos ver que al ser estos dos campos requeridos no se permitirá continuar con la consulta de la cartera

del cliente a menos que estos sean suministrados, por tal motivo se puede validar el correcto funcionamiento del componente a nivel de consulta. Por otro lado en la segunda imagen podemos ver como se verá el elemento que nos mostrará el componente de consulta una vez se diligencien los datos del formulario correctamente.

Figura 33

componente de consulta de comisiones



The screenshot displays a web application interface. At the top, there is a dark green navigation bar with a logo on the left and menu items 'Inicio', 'Registro de pagos', 'Consultas', and 'Admin' on the right. Below the navigation bar, the main content area is titled 'Panel de consultas de comisiones'. In the center of this panel, there is a modal window with a green border and rounded corners. The modal is titled 'Autenticar' and contains a form with a label 'Hash de la entidad' and a text input field containing the word 'Hash'. Below the input field, there is a red error message that reads 'El hash es necesario'. At the bottom of the modal, there is a yellow button labeled 'Continuar'.

Nota. En la figura se muestra el formulario con los campos requeridos

Ahora en esta figura tenemos el componente de consulta de comisiones por entidades de convenio, donde vemos como saltan los validadores al intentar consultar los datos sin diligenciar correctamente el formulario, donde se ve que este tiene un campo requerido, el cual es el hash de la entidad de convenio que desea consultar sus comisiones generadas por los servicios prestados, de esta manera vemos de manera visual cómo se valida el correcto funcionamiento de la interfaz de usuario a nivel de este componente de consulta.

Figura 34*Modal con los datos de comisiones de entidades de convenio*

Panel de consultas de comisiones

▼ Aplicar filtro de fecha

Fecha inicio: 13/06/2024

Fecha fin: 10/06/2024

¡Rango invalido!

| Id | Valor | Fecha de generación | Fecha inicial | Fecha final | Entidad | Sancionable | Metodo de convenio |
|--------|-------|---------------------|---------------|--------------|--------------|-------------|--------------------|
| #16001 | \$0 | Apr 1, 2024 | Mar 2, 2024 | Apr 1, 2024 | Market la 21 | No | Metodo 2 |
| #17001 | \$0 | May 1, 2024 | Apr 1, 2024 | May 1, 2024 | Market la 21 | No | Metodo 2 |
| #18001 | \$0 | May 31, 2024 | May 1, 2024 | May 31, 2024 | Market la 21 | No | Metodo 2 |

Nota. En la figura se ve la tabla con la información de las comisiones con su filtro

En la imagen anterior, se observa uno de los elementos clave en la validación del componente de la consulta de comisiones, el filtrado. Aquí, podemos ver que otro aspecto crucial de este componente está cubierto, la correcta entrada de las fechas en el filtro. Es fundamental que las fechas de inicio y fin se introduzcan en el orden adecuado. Si las fechas no están en el orden correcto, el filtro no se aplicará a la tabla de datos.

El sistema se asegura de que la fecha de inicio sea anterior a la fecha de fin para evitar errores y asegurar la precisión en la consulta de comisiones. Esto garantiza que los datos mostrados en la tabla reflejan correctamente el rango de fechas solicitadas. Esta validación es esencial para mantener la integridad de los datos y para proporcionar al usuario resultados precisos y útiles. Sin esta verificación, podría haber confusión o errores en la información presentada, afectando la fiabilidad del sistema. Por lo tanto, el correcto diligenciamiento de las fechas es un paso indispensable en el proceso de filtrado de la consulta de pagos.

Figura 35*Componente de acceso al módulo de administrador*

The image shows a web interface for administrator authentication. At the top, there is a green navigation bar with a logo on the left and menu items: 'Inicio', 'Registro de pagos', 'Consultas', and 'Admin' (highlighted with a yellow underline). Below the navigation bar is a white rounded rectangle with a green border, titled 'Autenticar'. Inside this box, there is a label 'Clave de administrador' above a text input field. The input field contains the word 'Clave'. Below the input field, there is a red error message: 'La clave de administrador es necesaria'. At the bottom center of the white box is a yellow button with the text 'Continuar'.

Nota. La figura muestra el formulario de acceso al módulo administrativo

En la figura anterior se presenta el formulario de acceso al módulo de administración del sistema. Este formulario proporciona una capa sencilla de seguridad basada en una contraseña, que es necesaria para poder acceder al módulo. Este método es un mecanismo básico pero eficaz para restringir el acceso no autorizado al módulo de administración, asegurando que solo los usuarios con la contraseña correcta puedan entrar.

Un aspecto crucial de este formulario es su sistema de validación. Los validadores implementados garantizan que el usuario no pueda proceder sin antes ingresar la contraseña. Esta característica es fundamental para la seguridad del sistema, ya que asegura que todos los intentos de acceso estén autenticados adecuadamente. La validación se realiza en tiempo real, proporcionando una experiencia de usuario fluida y segura.

Si un usuario intenta acceder al módulo sin proporcionar la contraseña, el sistema responderá con un error de validación. Este error es visualizado inmediatamente en la interfaz del usuario, indicando claramente que es necesario ingresar la contraseña para continuar. Este comportamiento está ilustrado en la imagen mostrada anteriormente, donde se puede ver el mensaje de error que se genera cuando se intenta acceder sin suministrar la contraseña.

Con esto, podemos validar el correcto funcionamiento de otro de los componentes de la interfaz. La implementación correcta de la validación y la presentación clara de los mensajes de error no solo mejoran la seguridad del sistema, sino que también mejoran la experiencia del usuario al proporcionar una retroalimentación inmediata y precisa. Este ejemplo demuestra cómo un componente bien diseñado puede contribuir significativamente a la usabilidad y seguridad de una aplicación.

Figura 36*Tabla en el módulo de administración con los datos de las facturas*

| Id | Valor | Fecha de facturación | Id del cliente | Cliente | Estado de la factura |
|-------|-----------|----------------------|----------------|-----------------------|----------------------|
| #2952 | \$80,000 | Mar 6, 2024 | #1007 | Manolo Felipe Garcia | PENDIENTE |
| #2954 | \$89,000 | Mar 13, 2024 | #1151 | Carla Yucatan Gimenez | PENDIENTE |
| #2955 | \$140,000 | Mar 5, 2024 | #1207 | Marina Soto Quispe | PENDIENTE |

Nota. En la figura se ve la tabla de las facturas con su filtro

En la imagen anterior, podemos ver otro de los elementos de validación de la interfaz de usuario, específicamente dentro del módulo de administración. Este elemento se presenta como un componente adicional de filtrado, similar a los componentes previamente descritos. Su función principal es validar el orden correcto en el que se ingresan los datos para aplicar un filtro por fecha.

La validación del orden de los datos es crucial para asegurar que el filtro funcione correctamente. Este validador comprueba que las fechas sean ingresadas en el formato y secuencia adecuados antes de permitir que se aplique el filtro. Esto previene errores y asegura que los resultados mostrados sean precisos y relevantes.

Este proceso de validación es parte integral del sistema, ya que garantiza la consistencia y confiabilidad de la interfaz de usuario. El hecho de que el validador impida la aplicación del filtro en caso de datos incorrectos es una medida de seguridad que protege contra posibles malentendidos o errores de entrada por parte del usuario.

En resumen, la implementación de este validador dentro del módulo de administración no solo mejora la usabilidad del sistema, sino que también fortalece la precisión de los filtros por fecha. Esto representa otro componente de la interfaz de usuario que ha sido validado.

Figura 37*Modal para agregar nuevas facturas*

| Valor | Fecha d |
|-------|-----------------------|
| 952 | \$80,000 Mar 6, 2 |
| 954 | \$89,000 Mar 13, |
| 955 | \$140,000 Mar 5, 2024 |

Nota. En la figura se muestra el modal con el formulario para agregar facturas

En la figura anterior se presenta el componente encargado de agregar nuevas facturas a los clientes de la empresa. Este componente es esencial, ya que permite registrar cada nueva transacción de manera organizada y eficiente. La interfaz del formulario está diseñada de tal manera que todos los campos requeridos deben ser completados antes de poder proceder con la creación de una nueva factura. Esto garantiza que se recopile toda la información necesaria para procesar y registrar adecuadamente la transacción.

Además, los validadores aplicados a este formulario juegan un papel crucial en la integridad de los datos. Estos validadores se activan en el momento en que se intenta añadir una nueva factura sin haber completado todos los campos obligatorios. Si algún campo requerido está vacío o contiene información incorrecta, el validador lo detectará y alertará al usuario, impidiendo que la factura sea añadida hasta que se corrijan los errores.

Finalmente, al verificar el correcto funcionamiento de este formulario y sus validadores, podemos considerar que otro elemento de la interfaz de usuario del sistema ha sido revisado y validado. Este proceso de verificación es fundamental para garantizar que el sistema en su conjunto funcione de manera coherente y eficiente, proporcionando a los usuarios una herramienta confiable para la gestión de las facturas de los clientes.

Figura 38*Componente de generación de comisión manualmente*

Generar comisiones manualmente

Entidad
-- Selecciona una entidad --

El identificador es necesario

Fecha inicio
dd/mm/aaaa

La fecha inicial es necesaria

Fecha fin
dd/mm/aaaa

La fecha final es necesaria

Continuar

Nota. En la figura vemos los validadores del formulario de generación de comisión

En esta otra figura se presenta la segunda y última parte del módulo de administración, donde se encuentra el componente diseñado para la generación manual de comisiones. La interfaz de este formulario está estructurada para que cada uno de sus campos sea obligatorio, asegurando que se recopile toda la información necesaria antes de proceder con la generación de comisiones.

El diligenciamiento obligatorio de cada campo del formulario se debe a la necesidad de garantizar la completitud de los datos ingresados. Si un usuario intenta continuar con la generación de una comisión sin haber completado todos los campos requeridos, los validadores del formulario se activarán. Estos validadores están diseñados para verificar que no se omita ningún dato, y en caso de encontrar algún campo vacío o con información incorrecta, impedirán la ejecución de la tarea. Esto no solo previene errores potenciales, sino que también asegura que todas las comisiones generadas cumplan con los estándares establecidos por el sistema.

La activación de los validadores y la imposibilidad de avanzar en el proceso sin cumplir con todos los requisitos evidencian el correcto funcionamiento de estos mecanismos dentro del componente. Esta funcionalidad es fundamental para mantener la integridad y confiabilidad del módulo de administración, proporcionando una capa adicional de seguridad y control. Al garantizar que todas las entradas sean válidas y completas antes de permitir la generación de comisiones, se minimizan los riesgos de errores y se facilita una administración más eficiente y precisa.

Figura 39

Componente de generación de comisión manualmente



Generar comisiones manualmente

Entidad
Sucursal de tecnologia

Fecha inicio: 20/06/2024

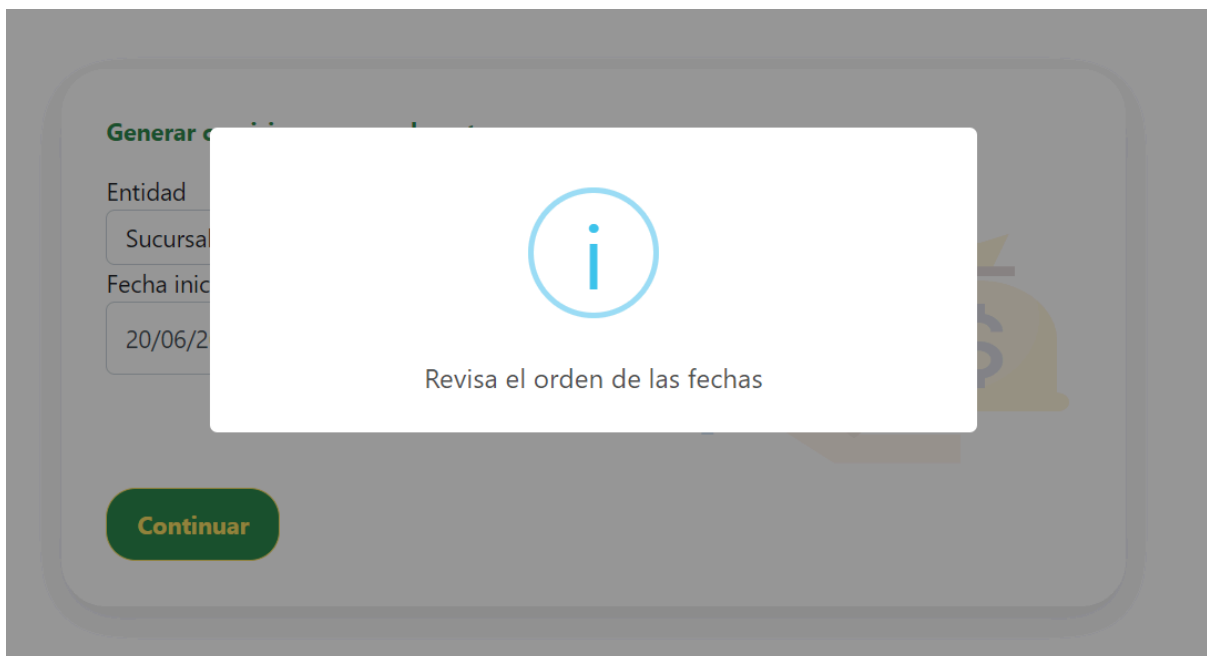
Fecha fin: 16/06/2024

Continuar

Nota. En la figura vemos el formulario con las fechas mal digitadas

Figura 40

Modal de validación de fechas



Generar comisiones manualmente

Entidad
Sucursal de tecnologia

Fecha inicio: 20/06/2024

Continuar

i

Revisa el orden de las fechas

Nota. En la figura vemos el modal de validación de fechas del componente

Ahora en estas dos últimas figuras vemos uno de los escenarios también controlados por parte de la interfaz de usuario a nivel de validadores del formulario del componente de generación de comisiones manual, aca vemos la situación en donde se diligencian todos y

cada uno de los elementos del formulario, pero al igual que en componentes anteriores el orden de las fechas se diligencia de manera errónea, al no tener un orden correcto, de esta manera vemos también cómo los validadores internos de la interfaz de usuario detectan dicho error y manejan la excepción por medio de un modal describiendo el problema presentado.

7. Conclusiones

El desarrollo de este proyecto de software ha resultado en una solución eficiente y escalable que satisface las necesidades específicas del usuario en la recepción y procesamiento de pagos y facturas. A lo largo del desarrollo, se abordaron varios desafíos técnicos que requirieron la implementación de mecanismos de validación de datos y seguridad, garantizando así la fiabilidad de las operaciones.

Uno de los resultados más destacados fue la creación de una interfaz de usuario intuitiva, que facilita la adopción y el uso eficiente del sistema. Esta interfaz no solo mejora la experiencia del usuario, sino que también contribuye a reducir el tiempo y los costos asociados con la capacitación del personal.

Por otro lado, el desarrollo de este software establece una base sólida para futuras mejoras en la gestión de pagos y facturas. El sistema automatiza tareas repetitivas y reduce la probabilidad de errores manuales, lo que permitirá a las empresas optimizar sus procesos operativos. La validación de datos integrada asegura que la información sea precisa y confiable, mejorando la integridad del sistema financiero.

Desde una perspectiva personal y profesional, este proyecto representó un desafío significativo que me permitió aplicar y expandir mis conocimientos en diversas áreas. La necesidad de aprender nuevas tecnologías y comprender profundamente los ciclos de desarrollo de software fortaleció mis habilidades de investigación y aprendizaje continuo. Además, el diseño y la implementación de la arquitectura del sistema desde cero mejoraron mi capacidad para planificar y ejecutar proyectos complejos de manera eficiente.

En resumen, este proyecto no solo cumplió con los objetivos técnicos y funcionales planteados, sino que también contribuyó significativamente a mi desarrollo profesional, proporcionándome experiencias y habilidades que serán aplicables en futuros proyectos de desarrollo de software.

7.1. Recomendaciones y trabajo futuro

Para escalar el proyecto hacia un sistema más completo y robusto, se recomienda desarrollar módulos específicos e individuales según los requerimientos de cada empresa que

utilice el software. Inicialmente, este software fue diseñado con un enfoque genérico para satisfacer las necesidades básicas de cualquier empresa que requiriera sus funcionalidades. Sin embargo, para un uso más eficiente y personalizado, es crucial adaptar y expandir los módulos para alinearse con los requisitos particulares de cada organización. Esto permitirá a las empresas obtener un valor máximo del sistema, optimizando su operatividad y eficiencia.

Además, como parte del trabajo futuro, es aconsejable desacoplar los módulos que actualmente funcionan de manera interdependiente. Este desacoplamiento permitiría que cada usuario, según su rol dentro del sistema, tenga acceso únicamente a los módulos y funcionalidades específicas que necesita. Implementar un sistema de gestión de usuarios con permisos y acciones personalizadas cerrará completamente el acceso al aplicativo, asegurando que cada usuario solo interactúe con las partes del sistema relevantes a su función. Esta estrategia no solo mejora la seguridad del sistema, sino que también facilita una experiencia de usuario más intuitiva y eficiente, optimizando el flujo de trabajo y la administración de tareas.

Además, es importante considerar la integración de capacidades de análisis y reportes avanzados dentro del sistema. Desarrollar herramientas que permitan la generación de informes personalizados y el análisis de datos en tiempo real proporcionará a las empresas información valiosa para la toma de decisiones estratégicas. Estas funcionalidades pueden incluir dashboards interactivos, gráficos y estadísticas detalladas que reflejan el rendimiento y las tendencias operativas. Al incorporar estas capacidades, el sistema no solo se convierte en una herramienta operativa, sino también en una plataforma que impulsa la eficiencia y la efectividad organizacional.

Referencias bibliográficas

Autentia. (s.f). Back: Introducción al backend y java.

https://www.autentia.com/wp-content/uploads/libros/Back_IntroduccionAlBack-Autentia.pdf

Manuel Trigas Gallego. (s.f). Gestión de proyectos informáticos. Metodología Scrum.

<https://openaccess.uoc.edu/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>

Autentia. (s.f). Frontend: Guía completa frontend.

https://www.autentia.com/wp-content/uploads/libros/Front_GuiaCompleta-Autentia.pdf

Henrik, K., & Mattias, S. (2010). Kanban y Scrum – obteniendo lo mejor de ambos.

https://eva.fing.edu.uy/pluginfile.php/246624/mod_label/intro/Kanban%20and%20Scrum.pdf

Anna Ferry Mestres. (s.f). Introducción al frontend y backend.

https://openaccess.uoc.edu/bitstream/10609/141486/1/Tecnologias%20y%20herramientas%20para%20el%20desarrollo%20web_Modulo1_Introduccion%20al%20frontend%20y%20backend.pdf

Gustabo, B. (2023). ¿Qué es MySQL? Explicación detallada para principiantes. Hostinger.

<https://www.hostinger.co/tutoriales/que-es-mysql>

Lasowski, J. (2011). Pizarra de Kanban.

<https://www.spiria.com/en/blog/method-and-bestpractices/introduction-kanban-1st-part/>

Mora-Castillo, J. A. (2016). Serialización/deserialización de objetos.

<https://www.scielo.sa.cr/pdf/tem/v29n1/0379-3982-tem-29-01-118.pdf>

Cengage Learning. (s.f). Arquitectura de software, conceptos y ciclo de desarrollo.

https://www.researchgate.net/profile/Perla-Velasco-Elizondo/publication/281137715_Arquitectura_de_Software_Conceptos_y_Ciclo_de_Desarrollo/links/57144e1408aeebe07c0641ab/Arquitectura-de-Software-Conceptos-y-Ciclo-de-Desarrollo.pdf

Oracle. (2023). Documentación de Oracle Cloud Infrastructure. Oracle.

<https://docs.oracle.com/es-ww/iaas/Content/Network/Tasks/managingpublicIPs.htm>

Google LLC 2020. Introducción a la documentación de Angular.

<https://docs.angular.lat/docs>

Autentia. (s.f). Software Design principios y patrones del desarrollo de software.

https://www.autentia.com/wp-content/uploads/libros/SoftwareDesign_PrincipiosyPatrones-Autentia.pdf

Mauvius Group Inc. 2021. La guía oficial del método Kanban.

https://resources.kanban.university/wp-content/uploads/2021/08/The-Official-Kanban-Guide_Spanish_A4.pdf

Luis Alfredo Valdes Hernandez. (s.f). Manual para la diagramación de procesos.

http://docencia.fca.unam.mx/~lvaldes/cal_pdf/cal18.pdf